An Enhanced Hierarchical Traitor Tracing Scheme Based on Clustering Algorithms

Faten Chaabane(✉), Maha Charfeddine, and Chokri Ben Amar

REGIM-Lab.: REsearch Groups in Intelligent Machines,
ENIS, University of Sfax, BP 1173, 3038 Sfax, Tunisia
faten.chaabane@ieee.org

Abstract. The easiness of using and manipulating digital media content has a volte face. In fact, although average users can simply be familiar with some manipulations such as a simple duplication, these manipulations can be dangerous with dishonest users whose target is illegal. Manipulating and duplicating digital media content via the Internet and Peer to Peer networks is available even to average users but can be used to unauthorized purposes with dishonest customers. Henceforth, facing the loss caused by unauthorized treatments and protecting the digital content become challenging to the media industry and research has led to different mechanisms of digital content protection. The aim of the multimedia distribution platforms, even Video on demand platforms, is to propose a suitable structure to the embedded fingerprints to ensure an efficient and fast tracing process in multimedia distribution platforms involving great number of users. The Tardos code has been the most popular tracing code due to its efficient tracing detection performance. One main challenge of the existing Tardos-based tracing approaches was to face the decoding complexity and the computational costs of the tracing process.

Hence, the tracing scheme we propose to improve in this paper was proposed previously as a group-based scheme which enables to construct groups of users according to a multi-level hierarchy. Based on clustering algorithm, we propose to construct groups of users' fingerprints, and then to apply the tracing process. The main target is to show how deep is the impact of using a clustering algorithms in the hierarchical tracing scheme.

Keywords: Tracing · Traitors · Tardos · Clustering · Hierarchical

1 Introduction

The tracing traitors was proposed to fight the copyright infringement of multimedia contents through the internet and Peer to Peer networks. It consists in hiding a fingerprint in each release of the media content before its distribution to only purchasers. More than one approach was proposed in the literature to provide efficient tracing codes, ranging from signal processing-based approaches [21]

F.J. Martínez de Pisón et al. (Eds.): HAIS 2017, LNAI 10334, pp. 379–390, 2017.
DOI: 10.1007/978-3-319-59650-1_32

to cryptology-based approaches [2,19]. The well-known probabilistic code called Tardos code has been proposed in [19] and has attracted increasing interest due to its short code length with an adjustment of the error probability [18]. Despite its fair compromise between the code length and the detection rate, the Tardos code still requires improvement with regard to its complexity decoding. This weakness is unequivocal in the case of multimedia distributing systems which involve a large size of audience and are expected to use a low complex pirates' retrieving process. The fingerprinting schemes based on Tardos code have witnessed a flurry of research efforts. This part surveys the current state of the art in traitor tracing approaches based on Tardos code. Some approaches focus on applying shifts to the Tardos code itself. Whereas, few other attempts have been made to ameliorate its tracing process by using a group-based property. According to [22], the group-based tracing scheme mainly consists in grouping users having common characteristics according to the assumption that they have more probability to cooperate together. Henceforth, this type of scheme may reduce the Tardos users' search space and consequently the complexity of its decoding step.

As a first example, in [1], a two-level hierarchical structure was proposed for the fingerprint generation process. Assigning users to a group was made randomly without considering any relationships inter or in groups. The main target was to minimize the complexity of the Tardos tracing decoding process.

In [12], the idea was to focus on the hierarchical structure proposed by [22], not for independent Gaussian signals but for Tardos-based fingerprints. Despite the good detection rates proven in this technique, the main weakness was the important length of the tracing code. In the same context, and according to the identifier multi-level hierarchical structure we proposed previously, it was necessary to find a suitable tracing strategy which enables a first group selection and then user accusation in the retrieved group. According to related work, we have focused on rising to the challenge of improving robustness results and accusation rates of Tardos code. Thus, we tried to reduce its complexity computation by proposing a multi-level hierarchical fingerprint in [5]. Henceforth, we will study in this paper the impact of clustering algorithm used to generate the groups of fingerprints on the two-level tracing scheme proposed in [6] as an accurate tracing strategy to our hierarchical fingerprints. The paper is organized as follows: in Sect. 2, we remind the scheme of the multi-level hierarchical fingerprint we proposed in [5]. In Sect. 3, we detail the structure of the improved tracing scheme. Section 4 presents the experimental assessment, and finally we conclude.

2 The Proposed Multi-level Fingerprint Generation Step

The majority of the traitor tracing field agree upon the fact that the whole fingerprinting system includes three main steps: the fingerprint generation step, their embedding into the media content and the tracing process [15]. According to related work, we have focused on rising to the challenge of improving robustness results and accusation rates of Tardos code. Thus, we try to reduce

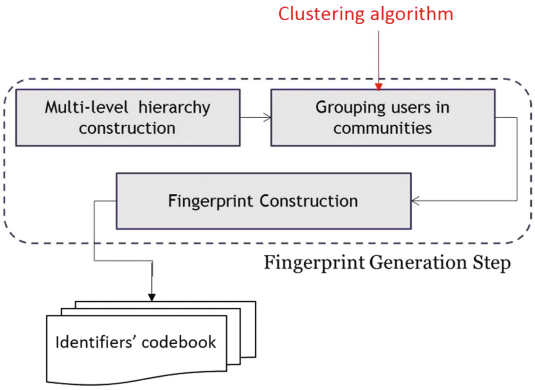


Fig. 1. Details of the fingerprint generation step.

its complexity computation by proposing a multi-level hierarchical fingerprint. Then, we embed it by an original robust watermarking technique which has proven good robustness and imperceptibility results compared to other existing approaches [9,10,17,20]. For multimedia distribution platform integrating a Tardos serialization system, one key constraint in the tracing scheme is the required computational costs to compute the scores S_j of all considered users, which is around $O(n \times m)$ operations, where n is the number of users and m is the codeword length. Through this part, we propose to involve a clustering step, as shown in Fig. 1 to construct our multi-level hierarchical fingerprint. The only requirement is the codeword length of users' fingerprints which is constrained by the applied watermarking technique used in the tracing scheme [7] and the number of groups defined by the hierarchy. The result of the generation step as shown in Fig. 1 is a codebook of codewords; a set of clustering-based identifiers.

2.1 The Hierarchy Construction Step

Reducing the search space of dishonest users by assigning a user to a specific group as depicted in Fig. 2 represents a suitable solution to face the Tardos accusation costs. The user assignment to a group can be used to counter different types of coalitions: temporal, geographic, social, etc. In the hierarchy, each chosen constraint corresponds to a level. We will detail each selected constraint separately.

Thus, according to this study made in [3], we embrace a multi-level hierarchy in the fingerprint generation step. Each criterion is represented by a level in the hierarchy. The first level is the time level where we assume that the most important period for a video life in a VOD platform is about 4 months [8,16]. Hence, we consider two groups of two-month-duration: in the first one, users' curiosity is moderately important and increases gradually to reach the maximal audience interest and in the second one it decreases to reach the minimal bound in the fourth one. A second level in the hierarchy represents the geographic criterion

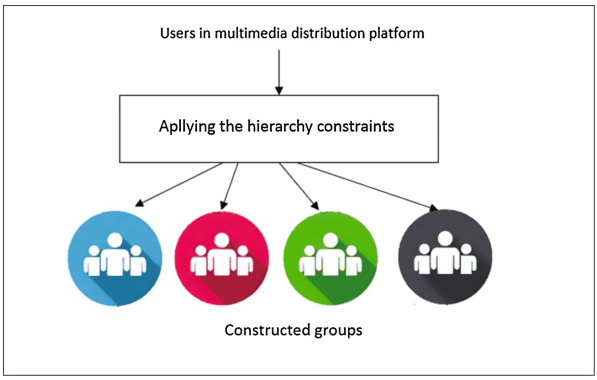


Fig. 2. Applying group-based properties to construct groups of users.

where we propose to divide platform users to two essential regions: Zone_A where the piracy phenomenon is very important, especially in Asia and Africa continents, and Zone_B where the piracy phenomenon is less important especially in Europe, America and Austria. We divide each continent to two groups of the main large countries in the third level. In the last levels, we are interested in social criteria, such as the age and the gender ones, etc. Once the hierarchy is fixed, we construct the users' fingerprints or codewords. Then, we apply a clustering step to construct groups of identifiers. Hence, The resulting fingerprint for each user is the concatenation of his community identifier concatenated to his personal identifier which is encoded with Tardos code.

$$Final_{identifier} = id_{level1} + id_{level2} + .. + id_{levelk} + personal_{id} \quad (1)$$

2.2 The Group-Based Construction Involving Clustering Step

According to [4,5], the generation of group identifier using the K-means clustering algorithm has proved the most efficient group detection rates. We remind that this clustering step consists in computing iteratively the Euclidean distance between each fingerprint and the group center. For each iteration, groups are constructed according to the minimal computed distance. Once this distance is unchanged, the algorithm converges and we have the final K clustered groups of fingerprints. Thus, the resulting modified user identifier is the concatenation of his clustered group identifier and his personal identifier as follows: $ModifiedUser_{id} = group_{id} + personal_{id}$ with $group_{id} = id_{clusteredgroup}$. Once the clusters are obtained, the traitor tracing process is applied. The retrieved fingerprint is treated to check the detection success.

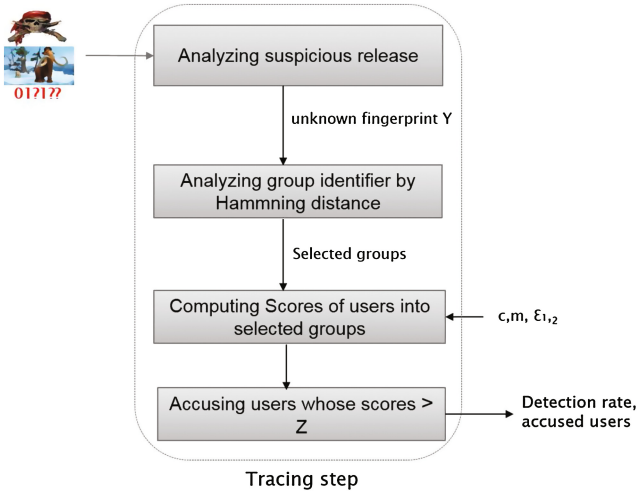


Fig. 3. Details of the tracing step.

3 The Tracing Process Using Multi-level Hierarchical Fingerprint

One key requirement in the fingerprinting system is the tracing process: when the supplier detects a copy with unknown fingerprint Y, he tries to trace back colluders by analyzing the extracted fingerprint Y, retrieving its similarity to a group identifier and hence tracing individual colluders. The tracing is performed here by the Tardos code. As shown in Fig. 3, group selection is based on computing Hamming distance between the ID_{group} of the suspicious copy and the other groups' identifiers. The selected groups are the closest to the extracted one; namely groups having the smallest Hamming distance to the ID_{group} of the suspicious copy. Then, the tracing process continues with the Tardos tracing step, the score S_j is computed per user only in selected groups. The user whose score exceeds the threshold Z [14] is thus accused. The detection rate of the proposed system is then computed to check its efficiency.

4 Experiments and Discussion

The real challenge of a traitor tracing scheme is to cover the gap between theoretical and practical results. Optimizing the fingerprinting code parameters and preserving the robustness even if the collusion size increases are the most important requirements of a traitor tracing scheme. In this contribution, we propose to construct a multi-level hierarchical fingerprint whose structure should improve the Tardos accusation process in terms of time consumption and tracing performance. To validate this multi-level structure, we propose to evaluate its impact on the tracing process when considering different collusion attacks and varying

collusion size. In all the experiments, we focus on varying the number of levels by comparing the multi-level hierarchical fingerprint to the non-hierarchical one. We propose, thus, to generate 1000 users' codewords with 5 colluders in a first example and 8 colluders in the second experiment. In a second set of experiments, we validate the impact of using a clustering algorithm on the tracing performance.

4.1 Evaluation of the Multi-hierarchical Fingerprint Versus the Non Hierarchical One in Terms of CPU Time Consumption

It is undeniable that one key constraint in the Tardos-based tracing scheme is the required computational costs to parse all users' codewords and compute their corresponding scores before the accusation.

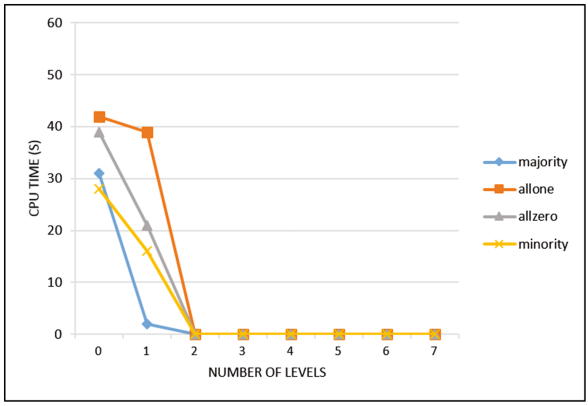


Fig. 4. CPU time consumption by hierarchical structures for different collusion attacks and for collusion size $c=5$

To show the outperformance of the proposed multi-level hierarchy when addressing the time consumption during the accusation process, we propose to compute the required CPU time to trace colluders for different collusion attacks and two collusion sizes $c=5$ and $c=8$. The time consumption in computed for varying hierarchical structures, ranging from hierarchy of one level to hierarchy of seven levels. The non-hierarchical structure is referred by a number of levels equal to zero. As illustrated in Figs. 4 and 5, the time consumption is reduced significantly when the number of hierarchical levels is increasing. For the non-hierarchical structure, it exceeds 30 s for the majority vote attack for the two collusion size, nevertheless it is close to 0.1 s since the second level for $c=5$ and below 2 s since the fourth level. Regarding the payoff in time when using the hierarchical structure, it exceeds 95% and 60% since the second level for the two curves. Now, when regarding the collusion size, it is clear that the tracing process spends more time to retrieve 8 colluders than for 5 colluders. Indeed,

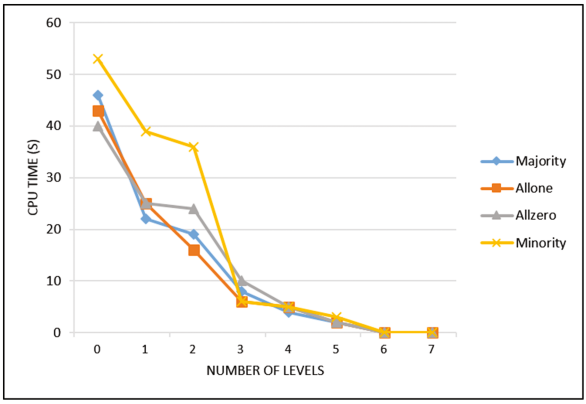


Fig. 5. CPU time consumption by hierarchical structures for different collusion attacks and for collusion size $c=8$

for $c=8$, the CPU time becomes close to 0.1 s in the sixth level. The impact of the collusion attack on the required CPU time is also an important point. In fact, from the two figures, we notice that some attacks such as the Minority vote attack has deeper effect on the time than the others. This can be proved by the fact that time values are the most important for this attack. However, with the hierarchical structure, this required time is also reduced significantly.

When studying the CPU time criterion, we notice that the hierarchical structure provides an important reduction in the time consumption of the tracing process. This can be explained by the fact that according to this structure, users are grouped together and hence the search space of the Tardos code is reduced to only the selected groups.

4.2 Evaluation of the Multi-hierarchical Fingerprint Versus the Non Hierarchical One in Terms of Positive False Alarm

Now, another important point to evaluate the tracing performance is the positive false alarm probability, the probability of accusing falsely an innocent.

As shown in Figs. 6 and 7, the probability of falsely accusing innocents is also reduced when the number of levels is increasing. For the non-hierarchical structure, the pfa is very important and exceeds 0.5 for the majority of collusion attacks. However, it is closer to 0 since the third level for $c=5$ and the sixth level for $c=8$. When studying the pfa rates, we also notice that some collusion attacks have deeper effect on the accuracy of the detection rates than others. In our experiments, the minority attack has the highest values of pfa. These values are also reduced with hierarchical structure. The behavior of the pfa curve can be explained by the fact that the hierarchical structure keeps more accuracy to the accusation process. Parsing groups susceptible to contain colluders prevents from accusing falsely innocent users. The whole experimental assessments were made with the proposed assumption that users belong to the

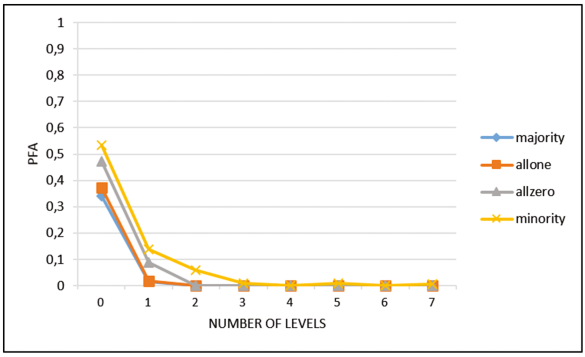


Fig. 6. Probability of false positive of hierarchical structures for different collusion attacks and for collusion size *c=5*

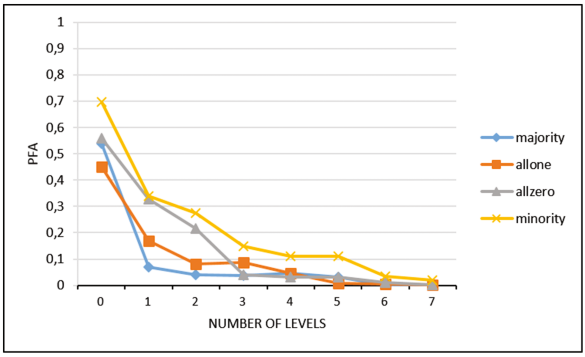


Fig. 7. Probability of false positive of hierarchical structures for different collusion attacks and for collusion size *c=8*

same group. In case of having users in different groups, the proposed tracing algorithm is executed for all the selected groups (having a minimal Hamming distance). Retrieving one colluders is fairly sufficient to stop the tracing process. Otherwise, the tracing process continues to parse all the selected groups. If users belong to all the groups of the hierarchy, the tracing process is executed for all the groups because they have the similar Hamming distance, and hence we are in the worst case (which is in contrast with our assumption). In this context, in a last part of this paper, we focus on the group-based property of our hierarchy. In fact, this property is applied randomly to groups of fingerprints; the groups are constructed with no constraints. In [24], group of users are constructed by applying a clustering algorithm to the group identifier. Data clustering process in traitor tracing context can be assimilated to an unsupervised data analysis process whose goal is to partition unlabeled users' identifiers into groups of similar characteristics, called clusters. In this part of paper, we propose to study the users' grouping step by involving a clustering step to construct our multi-level

hierarchical fingerprint. The only requirement is the codeword length of users' fingerprints which is constrained by the applied watermarking technique used in the tracing scheme [7].

4.3 Comparison of the Proposed Fingerprinting Approach to Other Hierarchical Techniques

Compared to non hierarchical fingerprints, the experimental assessments prove that the multi-level hierarchical fingerprints have good robustness to collusion attacks and are able to provide good detection rates in fewer time. Now, when regarding the existing hierarchical tracing approaches, it is obvious that they belong to different classes: code-based tracing classes and signal-based ones. Hence, we propose to report the most interesting experimental results able to allow us to compare them to the proposed approach. Looking at the different results reported in Table 1, we compare the tracing complexity of respectively [1,12,22,23]. The smallest decoding complexity is shown for both the proposed technique and [23].

Table 1. Comparative tracing complexity results to some existing hierarchical techniques

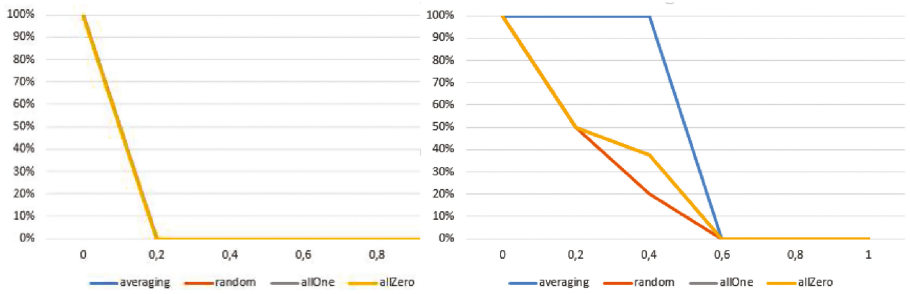
Hierarchical technique	Tracing complexity
[22]	$O(N \times l)$
[1]	$O(c_g \times \sqrt{N} \times l)$
[12]	$O(c_g \times n_g \times m)$
[23]	$O(\mathrm{m})$
The proposed technique	$O(\mathbf{m})$

Now, if we compare the experimental results related to the detection rate for [23], we notice that the majority vote attack is the only assessed collusion attack. We use the same experimental parameters proposed in [23]. We set the number of users n to 10^3, the number of groups to 8 and the number of colluders to 5. Compared to the CPU tracing time value of the [23] required to retrieve the 5 colluders whose value is about 10^{-1} s, the CPU time value required by the proposed technique is close to 10^{-2} s for a 3-hierarchical structure (the number of levels is tied to the number of groups).

4.4 Robustness to Collusion Attacks Between Different Groups

According to the experiments conducted in [5], we have proved that K-means which has provided the best recognition rate compared to the other clustering algorithms. Hence, we propose to test it for collusion including more than one group. As depicted in Fig. 8(a) and (b), the group recognition rate is enhanced

when using the K-means clustering algorithm. In fact, we vary the group participation rate in the collusion attack, and we show that the K-means clustering algorithm is able to detect at least 37.5% of colluders for a group participation which exceeds 40%. For the non hierarchical structure, the group recognition is null for all group participation size which proves that the tracing detection for this structure is not able to detect colluders belonging to different groups.



(a) non hierarchical scheme (b) clustering based scheme

Fig. 8. Robustness to different attacks: averaging, random, allOne, allZero.

Another important point is to compare the group detection capacity of the proposed technique to other group-based existing techniques. Moreover, due to the diversity of experimental assessments, we propose to report the main important and available experimental results in respectively [12,13,22,23]. Looking at the different results reported in Table 2, we notice that, for all the group-based techniques, the group detection capacity under the averaging collusion attack decreases with the increase of the group participation rate, called Group rate. Moreover, the proposed technique based on K-means algorithm shows good detection rate compared to the other existing techniques. Furthermore, we have proved previously that the group-based technique we propose provide a good compromise between the detection rate, the CPU time tracing and the decoding complexity. In this section, we have detailed the first contribution we propose in traitor tracing field. We have constructed a multi-level hierarchical fingerprint in order to apply a group-based tracing process which has proven a significant reduction of tracing and computational costs of the Tardos code even compared to other hierarchical existing tracing approaches.

Table 2. Comparative group detection capacity results to some existing group-based techniques

Technique	Group rate=0.1	Group rate=0.2	Group rate=0.25	Group rate=0.3
[22]	0.5	X	X	X
[13]	≤ 0.9	≤ 0.35	X	X
The proposed technique	**1**	**0.5**	**0.45**	**0.1**

5 Conclusion

In this paper, we presented the group-based fingerprinting system we propose for traitor tracing in multimedia distribution platform. The construction of fingerprints is based on a multi-level hierarchy where each level corresponds to a constraint inspired from the threat channel in the platform. The aim from this construction is to reduce the search space of the Tardos code and hence to reduce the complexity of the Tardos decoding step even in case of great number of users. We have also proposed to use a clustering step to enhance the group-based property which has improved the tracing results. We performed a detailed analysis of the proposed system performance according to two criteria: the robustness to collusion attacks and the tracing time criterion. The proposed fingerprinting system was evaluated for different collusion sizes. We also assigned an important consideration to the comparison of the performance of the proposed hierarchical system with non hierarchical one.

References

1. Akashi, N., Kuribayashi, M., Morii, M.: Hierarchical construction of Tardos code. In: International Symposium on Information Theory and Its Applications 2008, ISITA 2008, pp. 1–6 (2008)
2. Boneh, D., Kiayias, A., Montgomery, H.W.: Robust fingerprinting codes: a near optimal construction. In: Proceedings of the 10th ACM Workshop on Digital Rights Management, Chicago, Illinois, USA, 4 October 2010, pp. 3–12 (2010)
3. Chaabane, F., Charfeddine, M., Amar, C.B.: A multimedia tracing traitors scheme using multi-level hierarchical structure for Tardos fingerprint based audio watermarking. In: SIGMAP 2014 - Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications, Vienna, Austria, 28–30 August 2014, pp. 289–296 (2014)
4. Chaabane, F., Charfeddine, M., Amar, C.B.: Clustering impact on group-based traitor tracing schemes. In: 15th International Conference on Intelligent Systems Design and Applications (ISDA), Marrakesh, Morocco, 14–16 December 2015, pp. 440–445 (2015)
5. Chaabane, F., Charfeddine, M., Amar, C.B.: Novel two-level tracing scheme using clustering algorithm. J. Inf. Assur. Secur. **11**(4), 179–189 (2016). 11p
6. Chaabane, F., Charfeddine, M., Puech, W., Amar, C.B.: A two-stage traitor tracing strategy for hierarchical fingerprints. Multimedia Tools Appl. (2016)
7. Charfeddine, M., Elarbi, M., Koubaa, M., Amar, C.B.: DCT based blind audio watermarking scheme. In: SIGMAP, pp. 139–144 (2010)
8. Choi, J., Reaz, A.S., Mukherjee, B.: A survey of user behavior in VOD service and bandwidth-saving multicast streaming schemes. IEEE Commun. Surv. Tutorials **14**(1), 156–169 (2012)
9. Elarbi, M., Charfeddine, M., Masmoudi, S., Amar, M.: Video watermarking algorithm with BCH error correcting codes hidden in audio channel. In: IEEE Symposium on Computational Intelligence in Cyber Security, CICS 2011, Paris, France 11–15 April 2011, pp. 164–170 (2011)
10. El'arbi, M., Koubaa, M., Charfeddine, M., Amar, C.B.: A dynamic video watermarking algorithm in fast motion areas in the wavelet domain. Multimedia Tools Appl. **55**(3), 579–600 (2011)

11. Furon, T., Pérez-Freire, L.: Worst case attacks against binary probabilistic traitor tracing codes. CoRR abs/0903.3480 (2009)
12. Hamida, A.B., Koubàa, M., Nicolas, H.: Hierarchical traceability of multimedia documents. In: Computational Intelligence in Cyber Security, pp. 108–113 (2011)
13. He, S., Wu, M.: Collusion-resistant video fingerprinting for large user group. IEEE Trans. Inf. Forensics Secur. **2**(4), 697–709 (2007)
14. Laarhoven, T., de Weger, B.: Optimal symmetric Tardos traitor tracing schemes. CoRR abs/1107.3441 (2011)
15. Liu, K.: Multimedia Fingerprinting Forensics for Traitor Tracing. EURASIP Book Series on Signal Processing and Communications. Hindawi Publishing Corporation, Cairo (2005)
16. Liu, N., Cui, H., Chan, S.H.G., Chen, Z., Zhuang, Y.: Dissecting user behaviors for a simultaneous live and VOD IPTV system. TOMCCAP **10**(3), 23 (2014)
17. Mejdoub, M., Fonteles, L.H., Amar, C.B., Antonini, M.: Fast indexing method for image retrieval using tree-structured lattices. In: International Workshop on Content-Based Multimedia Indexing, CBMI 2008, London, UK, 18–20 June 2008, pp. 365–372 (2008)
18. Peikert, C., shelat, A., Smith, A.: Lower bounds for collusion-secure fingerprinting. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 472–479 (2003)
19. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC, pp. 116–125 (2003)
20. Wali, A., Ben Aoun, N., Karray, H., Ben Amar, C., Alimi, A.M.: A new system for event detection from video surveillance sequences. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010. LNCS, vol. 6475, pp. 110–120. Springer, Heidelberg (2010). doi:10.1007/978-3-642-17691-3_11
21. Wang, Z.J., Wu, M., Zhao, H., Liu, K.J.R., Trappe, W.: Resistance of orthogonal Gaussian fingerprints to collusion attacks. In: 2003 International Conference on Multimedia and Expo, 2003, ICME 2003. Proceedings, vol. 1, pp. I-617–I-620, July 2003
22. Wang, Z.J., Wu, M., Trappe, W., Liu, K.J.R.: Group-oriented fingerprinting for multimedia forensics. EURASIP J. Appl. Signal Process. **2004**(14), 2153–2173 (2004)
23. Ye, C., Ling, H., Zou, F., Lu, Z.: A new fingerprinting scheme using social network analysis for majority attack. Telecommun. Syst. **54**(3), 315–331 (2013)
24. Yong, Z., Aixin, Z., Songnian, L.: DCT fingerprint classifier based group fingerprint. In: 2014 International Conference on Audio, Language and Image Processing (ICALIP), pp. 292–295, July 2014