Practical watermarking for multimedia traitor tracing

Abdul Rehman

COLLEGES BRETAGNE LOIRE | SCIENCES POUR L'INGENIEUR ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE
MINES-TELECOM ATLANTIQUE BRETAGNE PAYS DE LA LOIRE
IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 648
Sciences pour l'Ingénieur et le Numérique
Spécialité : *Télécommunications*

Par
Abdul REHMAN

Practical Watermarking for Multimedia Traitor Tracing

Thèse présentée et soutenue à IMT Atlantique, Brest, le 19 Décembre 2024
Unité de recherche : Lab-STICC
Thèse N° : 2024IMTA0450

Rapporteurs avant soutenance :

M. CANCES Jean-Pierre Professeur ENSIL/ENSCI, Limoges
M. LE MASSON Jérôme Professeur Université de Rennes, St-Cyr-Coetquidan

Composition du Jury :

President :	Mme. FONTAINE Caroline	Directrice de recherche CNRS, Paris
Examinateurs :	M. BOUTILLON Emmanuel	Professeur Université de Bretagne Sud, Lorient
	M. CANCES Jean-Pierre	Professeur ENSIL/ENSCI, Limoges
	M. LE MASSON Jérôme	Professeur Université de Rennes, St-Cyr-Coetquidan
Dir. de thèse :	M. GUILLOUD Frédéric	Professeur IMT Atlantique, Brest
Co-dir. de thèse :	M. ARZEL Matthieu	Professeur IMT Atlantique, Brest

Invité(s) :

M. DION Jean Ingénieur de recherche b-com, Rennes
M. LE GUELVOUIT Gaetan Ingénieur de recherche b-com, Rennes

ACKNOWLEDGEMENT

Without the assistance of several helpful persons, this thesis could not have been completed. I'm happy to have this chance to express my gratitude to them.

First of all, I want to start by expressing my gratitude to Gaëtan Le Guelvouit and Jean Dion, my thesis supervisors, for the three wonderful years. I had the pleasure of working and learning with them. I express my gratitude to Frédéric Guilloud and Matthieu Arzel, my thesis directors, for their confidence in me to complete my thesis work.

Second, I would like to thank the members of my thesis jury: all the members of the jury for their interest in my work.

Throughout my three years of PhD studies, I was surrounded by friendly individuals at both the IMT Atlantique and the Institute of Research and Technology b<>com. I'd like to thank the IRT b<>com Trust and Security team for creating a nice environment in the room. I appreciated the coffee conversations with Menuka Perera, Tompoariniaina Andriamilanto, and Xuan Chen, as well as the team-building events such as the escape room with Valérie Denis, Gaëtan Le Guelvouit and Tania Pouli. Additionally, I would like to thank b<>com for organizing frequent trips to IMT Atlantique - Brest for the enhanced teamwork and collaboration with my thesis directors. Moreover, I would like to express my gratitude to Isra Khaled, Camilla, and Ismael Ahmad of the CODES team at IMT Atlantique for their insightful critiques of our work and the stimulating conversations we had over coffee during lunch.

Next, I would like to thank my friends for their support throughout my thesis studies. In particular, Shakir Khan from Bertrandt Group and Moeen Ali Naqvi from Simula laboratory at the University of Oslo with whom I shared this three-year adventure both distantly, and even for the warm exchanges in person.

I want to express my gratitude to my family, both local and distant, for their support and belief in my ability to complete this thesis. Lastly, the two most significant people in my life and my role models, my parents, have my sincere appreciation. I could never have accomplished so much and reached this point in my life without their unwavering love, support, and innumerable sacrifices. This thesis is dedicated to them.

RÉSUMÉ EN FRANÇAIS

Motivations

Ces dernières années, les progrès rapides des technologies de l'information numérique ont considérablement facilité notre travail quotidien et notre mode de vie. Le développement des smartphones, des appareils photo, des TV connectées et des ordinateurs personnels nous a encouragés à produire, consommer et partager des contenus multimédias. En outre, les progrès des télécommunications et l'utilisation généralisée des réseaux à très haut débit ont rendu la distribution et le partage de contenus multimédias via l'internet incroyablement simples et attrayants.

Néanmoins, cela pose également certains problèmes sérieux, tels que la copie illégale, la modification de contenu, le téléchargement illégal, la redistribution illicite, etc. Les pirates peuvent rapidement modifier le contenu multimédia à l'aide d'attaques basiques tout en préservant une bonne qualité, puis le redistribuer sans légitimité, violant ainsi les droits légaux des propriétaires multimédias, également connu sous le nom de piratage. Le piratage peut nuire aux opérations numériques et aux modèles commerciaux. Un exemple typique est celui des nombreux utilisateurs non autorisés qui téléchargent des films célèbres à l'aide de logiciels de réseaux de streaming vidéo peer-to-peer tels que PPLive [1]et PPStream [2]. Quelle est donc la gravité du piratage ? Il est difficile d'en mesurer l'impact, surtout à l'échelle mondiale. Cependant, une enquête [3] a indiqué que les contenus piratés sont les contenus les plus regardés dans le monde. Chaque année, le piratage sur l'internet entraîne en moyenne la perte de 20,000 emplois.

Les travaux présentés dans cette thèse ont pour objectif de proposer une procédure préventive contre le piratage, en exploitant unsystème de marquage numérique (*fingerprinting*) comme technologie anti-piratage. Nous décrivons et fournissons une vue d'ensemble de chaque composant de ce système, et explorons les avantages et inconvénients potentiels. La thèse fait partie du projet parté par l'Institut de Recherche Technologique b<>com et intitulé « Video Watermarking to Fight Against Piracy for Videos on Demand », dont l'objectif est d'améliorer les solutions de diffusion de contenu, permettant au fournisseur de contenu de tracer tout utilisateur frauduleux, et enfin, de prendre les mesures technologiques et juridiques nécessaires.

Restriction du projet et objectifs de la thèse

L'objectif ultime du projet est de créer une approche globale du filigrane capable de résister aux attaques suivantes. La modification du format (4k, 1080p, 720p, 480p, 360p, 240p), du débit binaire (5Mbps - >125kpbs), du codec (mp4 (H.264, H.265, etc.), WMV) d'une vidéo par un pirate permet de supprimer ou de perturber le filigrane. Des incrustations (barres, logo, texte) peuvent aussi occulter ou altérer partiellement des filigranes. Il existe de nombreuses autres attaques dans la littérature , et le projet vise à résister à tous les types d'attaques.

Objectifs

Dans un système de marquage numérique, le fournisseur distribue le contenu aux clients en y incluant un code unique appelé filigrane. Cette technique vise à garantir que les utilisateurs ne remarquent rien de différent lorsqu'ils reçoivent le matériel, et que le fournisseur peut facilement détecter le filigrane. Lorsqu'une copie non autorisée est découverte, il est possible d'identifier les pirates responsables du partage illicite. Un système de marquage numérique fonctionne de la même manière qu'une chaîne de communication comprenant un émetteur, un canal et un récepteur. L'émetteur crée et incorpore des identifiants. Le récepteur décode et retrouve ces identifiants. Le canal représente à la fois les transformations du filigrane ainsi que la collusion entre plusieurs d'entre eux.

Contributions: Proposition d'un système de marquage numérique

Pour lutter contre le piratage, un système de marquage numérique fonctionne de manière similaire à une chaîne de communication, composée d'un émetteur, d'un canal et d'un récepteur. L'émetteur est responsable de la création et de l'intégration de l'identifiant (ID), tandis que le récepteur est chargé du décodage et du traçage de cet identifiant. La collusion et d'autres attaques de tatouage sont modélisées par des canaux de transmission, comme le montre la figure. 1.

Le système de marquage numérique que nous proposons se compose de deux étapes principales : une première étape utilisant des codes résistants à la collusion et une seconde de filigrane qui utilise une approche d'intégration à un contenu vidéo. Dans cette thèse, nous avons étudié ces deux étapes : l'étpape de tatouage du chapitre 2 est utilisée pour déterminer la combinaison optimale générateur-décodeur, et l'étape d'incorporation du chapitre 3 est utilisée pour augmenter la qualité d'extraction du filigrane. Le chapitre 4 décrit le processus de filigrane utilisé dans notre système.

Dans le système que nous proposons, un identifiant d'utilisateur est diffusé de manière aléatoire tout en subissant un codage spécifique (par exemple, des codes convolutifs). Après la

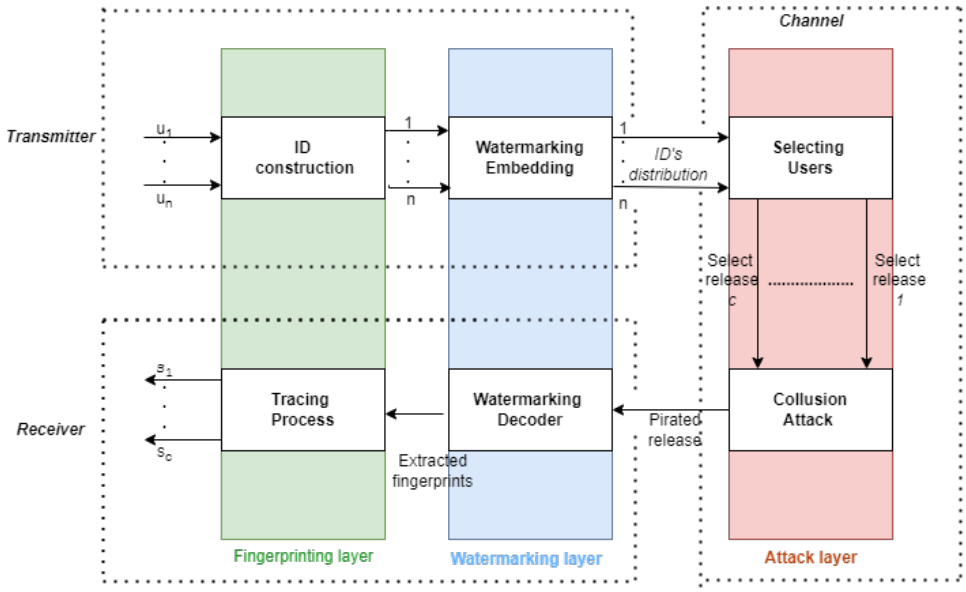


Figure 1 – Le schéma de conception complet de marquage numérique.

diffusion et le codage, une image en filigrane est créée et insérée dans une vidéo à l'aide de la transformée discrète en ondelettes, Discrete Wavelet Transform (DWT). Nous utiliserons FFMpeg pour ajouter une image en filigrane à une vidéo par le biais d'un mélange alpha.

Les approches de filigrane que nous proposons comportent trois étapes essentielles. 1) Création d'une image en filigrane : Sur la base de l'identifiant de l'utilisateur, une image en filigrane est créée après codage convolutionnel et étalement aléatoire. 2) Transformation : Une DWT est effectuée à l'aide de la fonction d'ondelettes Cohen-Daubechies-Feauveau (CDF9/7). Nous avons utilisé une décomposition à 3 niveaux de DWT. Après la DWT, les résultats sont quantifiés pour produire une image de filigrane. 3) Intégration et extraction : L'image en filigrane est intégrée dans une vidéo avec FFmpeg. Nous avons extrait les données à l'aide de Inverse Discrete Wavelet Transform (IDWT) avec désétalement et décodage.

ID générateur et décodeur

Le marquage numérique dans la protection du contenu met l'accent sur la résistance à la collusion. La couche d'empreinte digitale est divisée en trois composants : (1) la génération de code, qui implique la création de mots de code ; et (2) le décodeur de code, qui implique la mise en correspondance du mot de code mixte avec un groupe d'utilisateurs accusés comme indiqué dans la figure. 2.

Cette description est assez large et s'applique à (presque) toutes les couches d'empreinte digitale connues au sein de la protection du contenu. Le choix de différentes fonctions et seuils

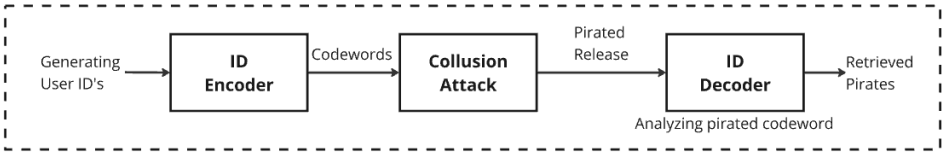


Figure 2 – Une approche binaire pour traquer les pirates : créer des mots de code utilisateur à l'aide d'un générateur de code et analyser la version piratée à l'aide d'un décodeur de code.

dans les composants de génération et de décodage distingue un schéma d'un autre. Voici un aperçu détaillé des processus de génération et de décodage qui distinguent un schéma d'un autre.

Générateurs de marques numériques

L'idée principale est de créer autant de mots de code qu'il y a d'utilisateurs et d'attribuer à chaque utilisateur un mot de code unique. L'efficacité de ces mots de code est évaluée par le nombre de colludeurs qui participent à des attaques de collusion et peuvent être accusés avec suffisamment de confiance tout en ayant peu de probabilité d'accuser une personne innocente. Cela est accompli en utilisant des codes résistants à la collusion, également appelés codes de traçage de traîtres, qui offrent une robustesse contre les attaques de collusion. Le but des codes résistants à la collusion est de générer des mots de code de sorte que, quelle que soit la force des attaques de collusion, la copie finale inclue toujours suffisamment de données pour reconnaître les colludeurs.

Décodage

Soit $\mathbf{y}$ le vecteur de bits du mot de code extrait correspondant à une attaque de collusion, et donc pas à un identifiant d'utilisateur légitime. Dans ce contexte, le décodage consiste à récupérer les utilisateurs légitimes ayant participé à la génération de la copie illégitime. Pour cela, des scores sont calculés à l'aide d'une fonction de notation $g : s_{ji} = g(X_{ji}, y_i, p_i)$. L'utilisateur j est déclaré comme colluder si $\sum_{i=1}^{m} s_{ji} > Z$ pour un seuil Z donné. Une autre stratégie consiste à accuser les utilisateurs ayant les scores cumulés les plus élevés.

Construction de code hiérarchique

Importance de la construction de code hiérarchique

L'un des défis de l'utilisation du code Tardos est sa détection exhaustive. Le détecteur a besoin de suffisamment de puissance de calcul pour calculer les scores de tous les utilisateurs potentiels. La procédure de détection a une complexité de $O(m \times n)$, ce qui reflète un coût de calcul

important. Par conséquent, à mesure que le nombre d'utilisateurs augmente, le coût de détection de la collusion augmente également. Cependant, la méthode de détection peut être améliorée en réduisant les dépenses de calcul. Étant donné que les scores de corrélation doivent être calculés pour tous les mots de code possibles, le nombre d'opérations augmente linéairement avec le nombre d'utilisateurs. Pour réduire considérablement le coût, une construction N-hiérarchique peut être utilisée lorsque certains utilisateurs sont plus susceptibles de s'entendre que d'autres, en fonction de leurs contraintes hiérarchiques.

Processus de génération et de décodage

Dans une construction de code hiérarchique, les utilisateurs peuvent être divisés en groupes. L'idée est de réduire le nombre d'utilisateurs par code Tardos pour améliorer les performances de traçage des pirates. Au sein des groupes, il y aura un ensemble d'utilisateurs qui sont plus susceptibles de s'engager dans une collusion comme le montre la figure. Par conséquent, le mot de passe total des utilisateurs avant l'intégration sera un mélange de l'identification du groupe et du mot de passe de l'utilisateur comme le montre la figure. Plutôt que d'évaluer tous les mots de passe des utilisateurs, nous pouvons d'abord décoder l'identification du groupe (groupe coupable) qui ressemble le plus à l'identifiant du groupe de collusion. Ensuite, nous pouvons enquêter sur la collusion au sein du groupe détecté.

Dans une construction de code hiérarchique, les utilisateurs peuvent être divisés en groupes. L'idée est de réduire le nombre d'utilisateurs par code Tardos afin d'améliorer les performances de traçage des pirates [4]. À l'intérieur des groupes, il y aura un ensemble d'utilisateurs plus susceptibles de s'engager dans la collusion, comme le montre la figure. 3. Par conséquent, le mot codé total des utilisateurs avant l'intégration sera un mélange de l'identification du groupe et du mot codé de l'utilisateur, comme le montre la figure. 4. Plutôt que d'évaluer tous les mots codés des utilisateurs, nous pouvons d'abord décoder l'identification du groupe (groupe coupable) qui ressemble le plus à l'identifiant du groupe de connivence. Nous pouvons ensuite étudier la collusion au sein du groupe détecté.

Lorsqu'il n'y a pas de collusion entre les groupes, le problème est relativement simple ; cependant, lorsqu'il y a une telle collusion, le problème devient plus compliqué. Par conséquent, pour gérer le problème, nous devons également utiliser des codes Tardos pour les identifiants de groupe. Lorsque nous utilisons des codes Tardos pour les identifiants de groupe, il est nécessaire d'examiner et de déterminer la meilleure longueur d'identifiant.

Conclusion

Dans la couche de marquage numérique, nous avons proposé de trouver le meilleur choix possible pour les composants (générateur-décodeur), et nous avons créé et testé différentes combinaisons de composants. Ensuite, nous avons effectué quelques expériences pour trouver la

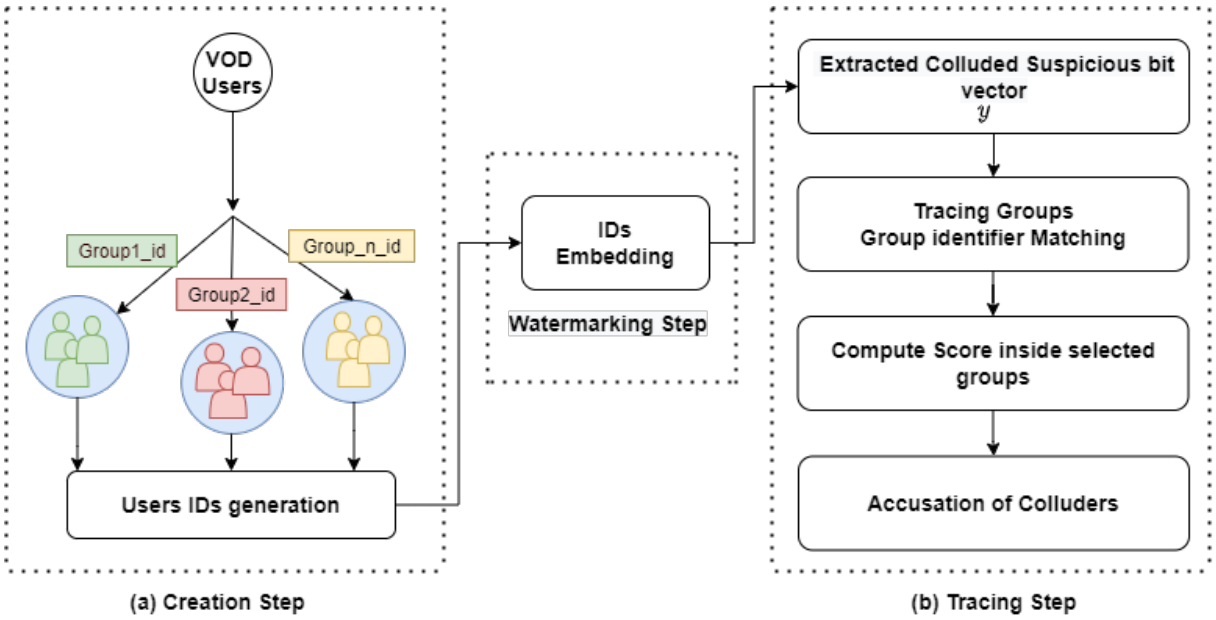


Figure 3 – (a) Étape de création: à l'aide de contraintes, un système hiérarchique divise les utilisateurs en groupes plus susceptibles de s'entendre. (b) Étape de traçage : après avoir détecté un marquage numérique en collusion, localiser le groupe puis tracer les colludeurs à l'intérieur de ce groupe.

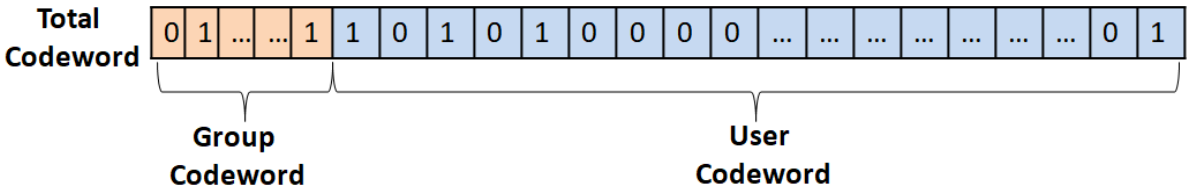


Figure 4 – Mot de code total avant l'étape d'intégration, obtenu en concaténant le mot de code du groupe et le mot de code de l'utilisateur.

meilleure combinaison de composants en ce qui concerne la détection des intrus et le coût de calcul. Tout d'abord, dans le modèle binaire, nous avons montré que la meilleure combinaison générateur-décodeur est le générateur *Laarhoven* avec le décodeur *Desoubeaux*. Cette combinaison permet d'obtenir la meilleure détection de collusion mais nécessite plus de temps de décodage.

Cependant, pour la configuration dans un système réel, nous avons examiné les mêmes générateurs/décodeurs et montré que la meilleure combinaison générateur-décodeur change alors. Nous avons montré que le générateur *Laarhoven* avec le décodeur Nearest Neighbor Search (NNS) est la meilleure combinaison sur la base du temps de traçage et de l'estimation du modèle de collusion.

Étalement aléatoire avec codage

Pour la couche de filigrane, nous avons discuté de l'importance et de l'impact du filigrane discret avec les identifiants des utilisateurs. Le filigrane discret est d'une qualité médiocre en terme de rapport signal sur bruit pic Peak Signal to Noise Ratio (PSNR). Les identifiants d'utilisateur qui ont un taux d'erreur binaire, Bit Error Rate (BER), plus élevé rendent le suivi des complices plus difficile. Les failles peuvent potentiellement entraîner la perte d'identifiants d'utilisateurs. L'utilisation de méthodes d'étalement aléatoire avec des identifiants d'utilisateur améliore le PSNR, suivi d'un taux de détection de collusion. Toutefois, l'étalement limite la longueur de l'identifiant.

Codes résistants à la propagation aléatoire et à la collusion

Une solution de protection de contenu multimédia robuste et sécurisée doit utiliser une approche de tatouage numérique forte et sécurisée pour inclure les identifiants des utilisateurs. Cependant, pour autant que nous le sachions, le développement d'une stratégie de tatouage numérique robuste et sécurisée est un défi, car la robustesse et la sécurité sont deux concepts distincts dans l'industrie du tatouage numérique.

Le tatouage numérique robuste est une technique qui devrait être robuste aux opérations de traitement de signal de routine sans stratégie particulière. Cependant, certaines attaques sont basées sur l'alternance avec la technique de tatouage numérique. Par conséquent, lors de ces attaques, l'attaquant prend des mesures précises pour effectuer une frappe afin d'effacer le tatouage numérique avec le moins de distorsion possible. Ces deux caractéristiques du tatouage numérique s'influencent et se restreignent mutuellement. Pour un tatouage numérique sécurisé, les messages sont recouverts de bits supplémentaires, appelés données redondantes, qui sont traités et transférés sur les canaux de communication.

La répartition aléatoire de x_j est basée sur α. Ici α représente la longueur de la séquence

aléatoire pour chaque symbole de code d'identification. (par exemple pour une longueur d'image $k = w \times h$, $\alpha = \frac{k}{m}$). Tout d'abord, un dictionnaire $\mathcal{D}$ doit être créé afin qu'il contienne les séquences aléatoires d pour la taille (ℓ, α) basée sur la clé secrète. Chaque nombre de symboles ℓ du code d'identification x_j est représenté par d séquences comme indiqué dans la figure. 5. La matrice résultante Z a une dimension $w \times h$ et est partitionnée en m sous-matrices indépendantes. Chaque vecteur d de longueur α séquence de $\mathcal{D}$, correspondant aux symboles de code d'identification.

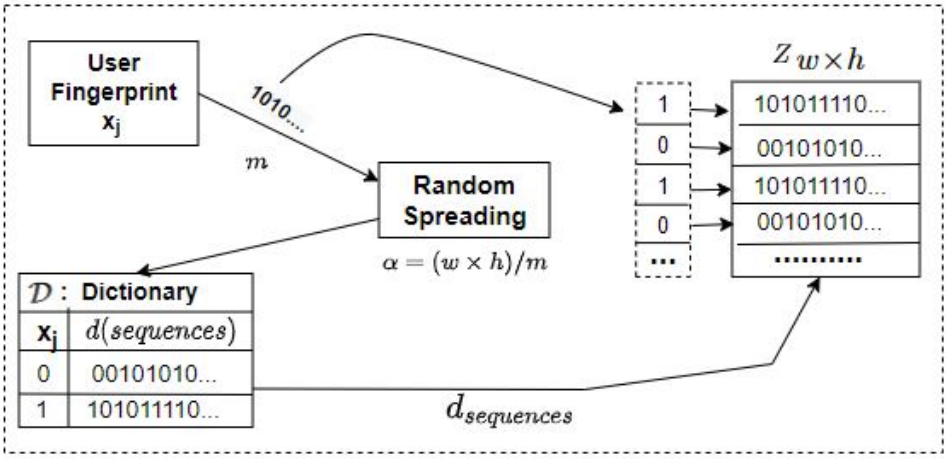


Figure 5 – Étalement aléatoire du code Tardos avec un taux α.

Pour un tatouage numérique sécurisé et robuste avec un BER inférieur, nous devons ajouter une redondance supplémentaire aux codes Tardos pour obtenir des taux de traçage plus élevés. Nous étudions des approches permettant d'ajouter de la redondance à l'identifiant utilisateur avant de l'intégrer dans le multimédia. La méthode la plus connue est la diffusion aléatoire, qui génère une séquence aléatoire pour représenter les bits individuels du message d'entrée. Dans notre cas, la longueur de sortie dépend de la résolution vidéo, donc un message d'entrée plus élevé réduit le taux de diffusion. Cependant, la diffusion aléatoire a des limites avec des messages d'entrée plus élevés pour une sortie fixe. Pour cette raison, nous explorons plus en détail l'utilisation de codes correcteurs d'erreur, Error Correcting Codes (ECC), couplée à la diffusion aléatoire, pour réduire BER et ensuite pour améliorer le suivi global des colludeurs.

Proposition de filigrane résistant à la collusion basé sur DWT

Dans notre filigrane vidéo résistant à la collusion basé sur DWT proposé, un identifiant d'utilisateur est réparti de manière aléatoire tout en supportant des codes convolutionnels. Après la répartition et le codage, une image de filigrane est intégrée dans une vidéo à l'aide de DWT. En utilisant le mélange alpha, nous utilisons FFMpeg pour mélanger l'image de filigrane dans

une vidéo. Notre méthode de filigrane suggérée se compose de trois étapes de base. Les détails de chaque étape sont fournis ci-dessous.

— Générer une image de filigrane: une image de filigrane sera créée en fonction de l'identifiant de l'utilisateur. De plus, la répartition aléatoire de l'identifiant de l'utilisateur avec l'ajout de codes convolutionnels.

— Transformation: un DWT est effectué sur la base de la fonction d'ondelettes CDF9/7. Nous avons utilisé une décomposition à 3 niveaux de DWT. Après DWT, les résultats sont quantifiés (0-255) pour créer l'image de filigrane RVB.

— Incorporation et extraction: L'incorporation d'une image de filigrane dans une vidéo est basée sur FFmpeg. Pour l'extraction, nous avons utilisé IDWT avec désétalement et décodage comme indiqué dans la figure 6.

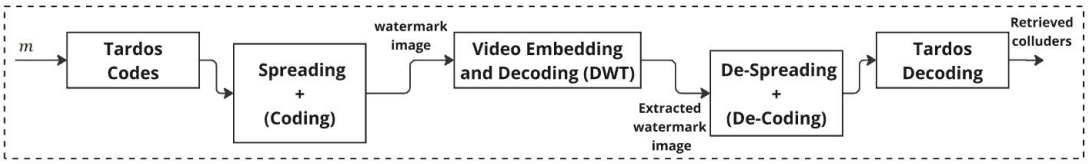


Figure 6 – Le schéma de filigranage vidéo proposé utilise des codes Tardos avec étalement aléatoire avec ECC.

Conclusion

Dans la couche de tatouage, nous utilisons l'approche de tatouage basée sur DWT combinée à un schéma de codage conjoint (étalement aléatoire avec codes convolutionnels) du chapitre 3. De plus, nous avons fourni un aperçu détaillé des techniques de tatouage et des raisons pour lesquelles nous avons choisi DWT. De plus, les attaques de collusion en temps réel sont présentées et étudiées en utilisant FFMpeg avec une approximation proche d'un modèle binaire.

Enfin, nous avons suggéré une configuration réelle et effectué des expériences avec de vraies vidéos. Nous avons examiné les performances des couches d'empreintes digitales et de tatouage dans notre schéma d'empreintes digitales basé sur DWT proposé.

Pour la couche d'empreintes digitales dans une configuration réelle, nous avons examiné les mêmes générateurs/décodeurs et montré que la meilleure combinaison générateur-décodeur diffère de celle sélectionnée pour les modèles binaires. Nous avons montré que le générateur *Laarhoven* avec le décodeur NNS est la meilleure combinaison. Les combinaisons générateur-décodeur binaire et réelle offrent un taux d'accusation similaire. La principale différence réside cependant dans le temps nécessaire au décodage. Ceci est essentiel dans les systèmes réels car nous devons découvrir rapidement le colludateur. Nous recommandons d'utiliser le décodeur NNS plutôt que *Desoubeaux*, car le décodeur *Desoubeaux* est plus complexe et difficile à mettre en œuvre.

Pour la couche de tatouage dans une configuration réelle, nous choisissons la combinaison générateur-décodeur la plus adaptée (*Laarhoven* et NNS). Ensuite, après avoir comparé les performances du schéma de codage conjoint à celles du schéma non codé, nous avons conclu que le schéma de codage conjoint surpassait.

Cette thèse est basée sur deux articles : [5] et [6].

Table of Contents

List of acronyms

DWT	Discrete Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
DFT	Discrete Fourier Transform
DCT	Discrete Cosine Transform
IDWT	Inverse Discrete Wavelet Transform
CDF9/7	Cohen-Daubechies-Feauveau
PSNR	Peak Signal to Noise Ratio
NNS	Nearest Neighbor Search
LSB	Least Significant Bit
FN	False Negative
FP	False Positive
ECC	Error Correcting Codes
VOD	Video on Demand
BER	Bit Error Rate
BSC	Binary Symetric Channel
BM	Branch Metrics
PM	Path Metrics
AWGN	Additive White Gaussian Noise
ACS	Add, Compare and Select
BCH	Bose–Chaudhuri–Hocquenghem codes
PM	Path Metric
BM	Branch Metrics
ACS	Add Compare and Select unit
LDPC	Low Density Parity Check codes
DRM	Digital Right Management
QMF	Quadrature Mirror Filter
HVS	Human Visual System

List of Figures

List of Tables

Introduction

The thesis is funded by a project within organization b<>com. b<>com explores, designs, and delivers the digital technologies of the future. b<>com develops technologies that serve key European industrial sectors, focusing on six core areas: connectivity, cybersecurity, digital twin, immersive interaction, future computing, and artificial intelligence. This thesis is part of the cybersecurity field, emphasizing multimedia content protection. The project aims to create and validate highly discreet watermarking algorithms and search for enduring watermarks in VOD video streams. This thesis contributes to the project's goal of addressing piracy (redistribution of video) in video streaming networks.

Songs, movies, mail, money, books, and TV shows have all undergone a relentless transition from analog to digital during the last few decades. The demand for cell phones, digital cameras, and personal computers has influenced multimedia content development, consumption, and sharing. Furthermore, the growth of broadband networks and technological advances in telecommunications have made media transmission and communication via the Internet relatively straightforward. Nonetheless, this carries an abundance of serious issues, including limitless replication, arbitrary change, unauthorized uploading, illegal redistribution, etc. In this age of wide digital content distribution, it is now more crucial than ever to provide reliable and powerful techniques to prevent all these issues. The formal documents are a small collection of things that we may access and control with a simple home computer. Purchasing video and other content items online is becoming more and more popular these days, as opposed to purchasing them on tangible media like CDs. Videos are perhaps the most vulnerable multimedia content, and unauthorized individuals are spreading videos for their gain and profit. Using simple techniques, pirates can swiftly change multimedia information without sacrificing quality. They can then redistribute it without authorization, violating the intellectual property rights of multimedia owners—a practice known as piracy. As a result of piracy, digital operations may suffer and the business model may be impacted.

Piracy has been more popular since the Internet has become widely used. Despite the ongoing efforts of businesses specializing in addressing this phenomenon and rights holders, digital piracy continues to increase. It is widely recognized that one of the key consequences of piracy is a loss of money for content producers.

How serious is the piracy issue then? It is challenging to measure the impact precisely, particularly globally. Even the most thorough studies sometimes present disparate numbers or employ various approaches to obtain their information. However, several statistics present a

depressing image.

- **Highest Watched**: Pirated content is the highest viewed content all over the globe. The United States contributes around 15 billion of that total, with 12.8 billion of those being TV shows and 2.2 billion being movies [8]. In other regions of the world, the statistics are similar: 215 billion views for television series and 44.7 billion for movies.
- **Job loss:** Every year, internet piracy causes over 500,000 individuals to lose their employment on average [9]. An estimated 71,000 jobs in the music industry alone are lost annually as a result of piracy. According to the Institute for Policy Innovation (IPI), between 230,000 and 560,000 jobs are lost every year in the USA due to piracy.
- **Revenue loss:** According to estimates from the International Chamber of Commerce, the problem is not going away; by 2022, the world's income loss from piracy is expected to exceed 991 billion [10]. Video piracy reduces the gross domestic product annually by 47.5 billion to 115.3 billion in the United States alone.

Piracy mostly targets content creators and distributors who can respond quickly. Nonetheless, certain content owners are more likely than others to face more serious violations.

- **Streaming services:** Illegal streaming is by far the most popular way for unauthorized people to acquire content; it can account for as much as 80% of all online piracy worldwide. In addition to passwords being freely shared, accounts can be compromised and login information sold on unofficial websites.
- **Businesses:** Hackers continuously pose a threat to sensitive corporate material. This might involve many forms of confidential internal material, private virtual events, or internal video meetings. As previously said, even information intended for public consumption is susceptible to piracy and appearance on dubious websites, which can cause long-term harm to the firm's brands.
- **Content creators:** Pirates jeopardize the lives of local and worldwide creators by illegally downloading content or utilizing screen recorders to capture content for resale or distribution.

The goal of the thesis is to design a preventive measure against piracy. A fingerprinting system is one type of anti-piracy technology. In a fingerprinting system, the supplier sends content to consumers by inserting a unique code as a watermark so that users notice nothing unusual while receiving the content. By doing so, the provider can readily recognize the watermark that has been placed on the content. When an unauthorized copy is discovered, it is feasible to determine which hackers can be held accountable for initiating the illegal sharing. A watermark is a secret alteration applied to a multimedia file by the content provider. The process of adding a watermark into the original file is known as *embedding* and the resulting file, which includes a watermark, is known as a *watermarked copy*. It is not possible to store all the watermarked files; practically, the content supplier just stores the changes that he made during distribution

(the watermark). In multimedia files, each user's watermark might be represented by a unique sequence of symbols (the fingerprint code). Then the content supplier needs to store:

— For each user, the unique sequence of symbols is presented in his multimedia copy.

— The position in the content where the different watermark symbol is embedded.

All these data are stored as a secret; if the content provider discovers an illegal duplicate due to piracy, he can examine it using a *watermarking decoder*. The decoder accepts both the stored information and the illegal copy. The decoder searches for the watermark symbol at each point, and if one is found, it is recorded as a detected symbol. The content provider compares the store symbols to the deleted symbols to identify the person who created the illegal copy. In this thesis, we will develop and investigate a fingerprinting method to combat piracy. The thesis is a part of the project named "Video Watermarking to Fight Against Piracy for Videos on Demand" at b<>com, whose objective is to improve content delivery solutions, allowing the content provider to trace any fraudulent users, and finally, to take the required technological and legal actions. The key challenge we address in this thesis is a watermarking solution with minimal latency that meets particular constraints given by b<>com project.

The thesis is arranged as follows. Chapter 1 discusses the need for content protection and the fingerprinting system as a solution to piracy. The fingerprinting system consists of three primary layers. Fingerprinting, watermarking, and channel layer. We went into detail about the functionality of these layers. The project limitation on the design of the fingerprinting system comes next. Based on these restrictions, we developed the thesis target for the fingerprinting system and made our contributions.

Chapter 2 begins with an outline of the fingerprinting layers components. We provide an overview and detailed background on collusion-resistant codes and collusion attack models. Then, we simulate and evaluate various combinations of each component of the fingerprinting layer to select the most effective possible generator-decoder combination for our studied cases. Then we explore hierarchical code construction and decoding, followed by a detailed background to minimize detection time and complexity. Then we offer dynamic settings for code construction as an alternative to the hierarchical one. Finally, we compare the performance of dynamic and hierarchical code construction via experimentation and simulation. Based on the discussion and experiments, Chapter 2 determines the most suitable generator-decoder combination for collusion-resistant codes. This chapter is based on our first publication [5].

Chapter 3 focuses on the construction of a watermarking technique with collusion-resistant codes. Then, we propose a method for embedding collusion-resistant codes through random spreading, followed by an overview of current spreading methods. In addition to random spreading, we propose using ECC in efforts to increase performance. Then, we propose a novel combination of ECC and random spreading and analyze its performance through simulations and experiments. This chapter is based on our second publication [6].

Based on the findings and recommendations of Chapters 2 and 3, Chapter 4 concentrates on the implementation of the problem in real time. In a real-time setup, we use a watermarking method to embed the watermark containing fingerprints into the videos. Furthermore, we discuss an experimental setup and a tool for performing attacks on real videos. Then, we examine in detail the performance of this real-time setup considering the outcomes from Chapters 2 and 3. We propose how a real-time setup would change the conclusions of Chapters 2 and 3.

Finally, we conclude this thesis and discuss future work and perspectives on secure and robust collusion-resistant watermarking.

Chapter 1

Problem Description and Project Constraints

In this chapter, we will address why we ought to protect online material as well as the many methods for doing so. One of these is the multimodal fingerprinting system. Then, we outline and provide an overview of each component of the multimedia fingerprinting technique, as well as explore potential advantages and downsides. This thesis is a minor part of major research aimed at developing a universal watermarking technique that is resistant to all attacks and modifications. Some project constraints will influence the thesis's aim. Given these constraints, we outline this thesis's research challenges and contributions.

1.1 Content Protection

Piracy has become more popular since the internet became widely used, thanks partly to P2P (peer-to-peer multimedia content) file-sharing tools that allow users to readily distribute and find almost any digital content while remaining virtually anonymous. For example, according to the Motion Picture Association's CEO [11], the COVID-19 pandemic's other side effect is increased movie piracy. According to The Wall Street Journal [11], movies like "The Conjuring: The Devil Made Me Do It", "The Suicide Squad" and "Black Widow" quickly became the most pirated films online after their day of release. Even after COVID-19 [12], current piracy levels remain far higher than before.

According to an industrial report by Park Associates [3], by 2027, there is a projected loss of $113 billion for streaming video providers serving just U.S. customers due to content piracy. As technology advances, these reports indicate that controlling piracy and the re-distribution of multimedia content is becoming more and more crucial.

The researchers have investigated various aspects to prevent piracy, such as the Digital Right Management (DRM) system [13], intrusion detection [14], and eavesdropping detection [15]. However, a multimedia fingerprinting system is the primary choice for content providers. It does not restrict a user from copying or utilizing the content but it is a preventive measure to piracy. In a multimedia fingerprinting system the copyright holder hides a unique watermark in each user's content copy. The purpose is to trace back the identities of the pirates when an illicit copy

is found, and then force them to be responsible for their actions. A multimedia fingerprinting system function consisting of fingerprint generation, and watermarking. Every layer's role will be described in the subsections that follow.

As an example, let us consider the VOD services. A VOD server distributes unique videos to a set of users who have subscribed to the service. However, there might be a pirate among the users who might leak or redistribute this video to multiple other persons. To fight against this illegal redistribution, a VOD provider can associate a unique user ID to the content. When an illegal duplicate is discovered, the user ID is retrieved and the user is accused of redistribution. To prevent pirates from changing the ID, it is hidden inside the video, using a technique known as watermarking.

1.1.1 Watermarking

Watermarking a user ID in the content must fulfill serval constraints:

— Independent: Each user ID should be fully independent of one another. Each user ID should be unique.

— Interference: The user ID should not modify or disrupt the content. It should not degrade the quality. Otherwise, it would be undesirable for the users. They will not purchase the multimedia file.

— Robustness: The user ID should be resistant to any changes in the content. If there is an attack or alteration to the video, the ID should be resistant to those attacks or modifications.

However, constraints $2 and 3$ are in direct contradiction with one another, because interference implies an upper limit for the visibility of the watermark, but robustness and security demand a lower limit for the capacity of the watermark on the content. In addition, robustness performance is relevant to normal operation, whereas security targets specific attacks; achieving both requires a trade-off. For these reasons, designing a watermarking scheme is challenging.

Over the past ten years, watermarking research has advanced, and as a result, many algorithms have been published in the literature. Numerous studies [16, 17, 18] discuss the qualities of various types of watermarking techniques from both a research and user viewpoint. The watermarking techniques may be classified as spatial domain or frequency (or transform) domain methods. Spatial domain watermarking is a technique for inserting a watermark into the original image in the spatial domain. The most common algorithm for spatial domain watermarking is Least Significant Bit (LSB) modification. This method changes the LSB of chosen pixels in the image. This method is comparatively simple. It can survive simple operations such as cropping and the addition of noise. However, most of the spatial domain methods cannot handle compression [19]. The main goal of frequency domain watermarking is to embed the watermarks in the spectral coefficients of the image. The most generally used transforms are the Discrete Cosine

Transform (DCT), Discrete Fourier Transform (DFT), and DWT.

On the detection side, if the content owner detects duplicate copies of some user, then he uses the watermarks decoder to decode the ID symbols. In this thesis, watermark embedding and detection should satisfy blind detection. In blind detection, the original material is not available to the watermarking decoder; whereas, in non-blind detection, the watermarking decoder may utilize the original content.

1.1.2 Digital watermarking model

The basic watermarking video model to fight against the illegal redistribution of one dishonest user is shown in Figure. 1.1. For each user, a server creates a watermark using the user's ID. This watermark is added to the video using some embedding techniques. This will eventually result in a unique video for each person based on their IDs. A pirate (dishonest user) may just re-distribute his video copy to a large number of unlawful users to generate income. The server can easily identify these suspicious video releases. The server can subsequently assess these unlawful video releases to swiftly identify the ID of the user that redistributed the video and block distribution for this traitor.

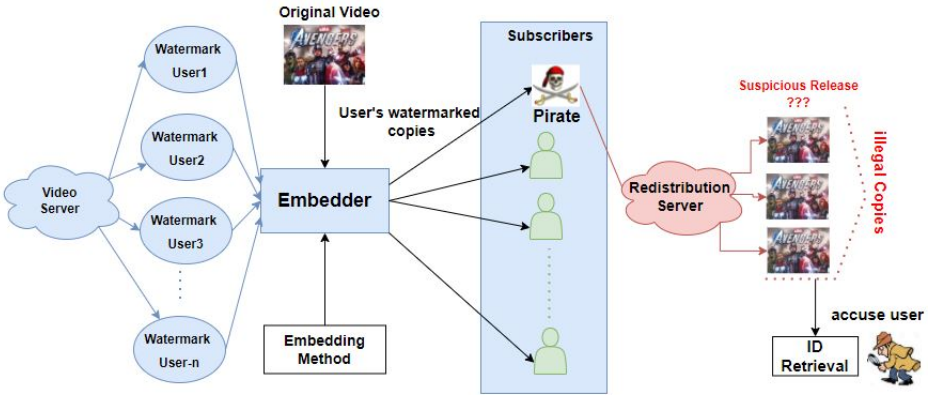


Figure 1.1 – In a VOD, after detecting the ID of a pirate in a suspicious release, the server terminated delivery to the pirate.

The user ID is encoded using a watermark image. This watermark image is subsequently added to the original image and transmitted across the communication channel. During transmission, channel signals are added to the transmitted image as noise. The receiver employs a watermark decoder to extract the original message from the noisy watermarked image.

A user ID may be lost or destroyed during the transmission of the content. A similar length of user ID and watermark image results in higher errors [20]. To address the issue, the ID must have additional redundancy. The ID length must be less than the size of the watermark image, and additional redundancy can be introduced by a technique called *spreading*. The spreading

is only feasible if the user ID is less than the size of the watermark image [21]. In spreading pseudo-random sequences are used to represent the bits of the ID. So that if there is some loss of bits while transmitting, the ID can be perfectly decoded. After the spreading of ID, the process of adding the into the original content is known as *embedding*, and the resulting file, which includes an ID(watermark image), is known as a *watermarked copy*.

The original and decoder watermark can differ by some PSNR due to channel noise, picture compression, and quantization. Lower PSNR results in a higher BER for the decoded ID. In this thesis, we utilize the Binary Symmetric Channel(BSC), to represent the errors as shown in Figure. 1.2. The next chapter provides a comprehensive comparison of BER and its impact on finding traitors using BSC.

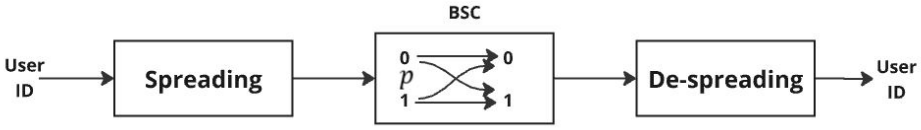


Figure 1.2 – Spreading of user ID before embedding the watermark image.

1.1.3 Watermarking attacks

The pirates are interested in the watermarks in their files to destroy the connection between their IDs and watermarked content; this is known as an attack on watermarking. The attacks can be based on either a single or multiple copies. The existing attacks on single-copy are divided into three types: removal, geometrical, and cryptographic [22].

The removal attack seeks to completely remove the watermark from the watermarked file by adding some additional information. Noise addition, cropping, compression, and other related techniques are all examples of removal attacks. Although these procedures may not be able to eliminate the watermark, they will cause severe harm. The geometric attacks are not intended to destroy the watermark, but rather to disrupt the watermark detector's synchronization with the underlying algorithm. The watermark detector can recover the encoded watermark if complete synchronization is restored. Nevertheless, the synchronization procedure may be too difficult to be practical. Geometric attacks encompass wrapping, transformation, jittering, and other related methods.

The literature [23, 24, 25] suggests the DWT is the most efficient watermarking technique against most attacks. So, in this thesis, we will also utilize DWT for the watermarking algorithm. The significance of utilizing DWT based watermarking comes from its capacity to break down signals at different sizes, which may be chosen based on the goal. Many signals rely primarily on the low-frequency component, which comprises the signal's characteristics, whereas the high-frequency component contains the signal's details or distinctions. It is a strong and vital approach

for watermarking, as it is robust against most attacks [26].

1.1.4 Collusion attacks models

Even so, most recent watermarking systems are resistant to some of the single-copy attacks [27]. But it is more difficult to handle attacks based on multiple copies also known as *collusion attacks*. To damage or blur the watermark, the hacker uses numerous copies of the same image/video, each with a distinct watermark, and then constructs a new copy that has a blurred watermark, which will be unknown to the provider. This is an issue (for example, in the film business or VOD), but it is not common practice since the attacker must have access to several copies of the same image/video, and the number required might be rather large.

In collusion attacks, the colluders could compare each other's copies, exposing various locations where they do not get the same symbol known as *detectable positions* or *marking assumptions*. The colluder can compare their IDs to determine where they received different bits, known as detectable places. Colluders can easily update the bits in detectable positions utilizing attack models. This extra information allows for a far more powerful attack on the watermark, which is challenging to handle. When colluders (a group of dishonest users) release a suspicious copy by using some collusion attack, the provider will have a significantly hard time finding all of the colluders. The provider then analyzes the suspicious content to identify and suspend the content delivery to all colluders as shown in fig. 1.3.

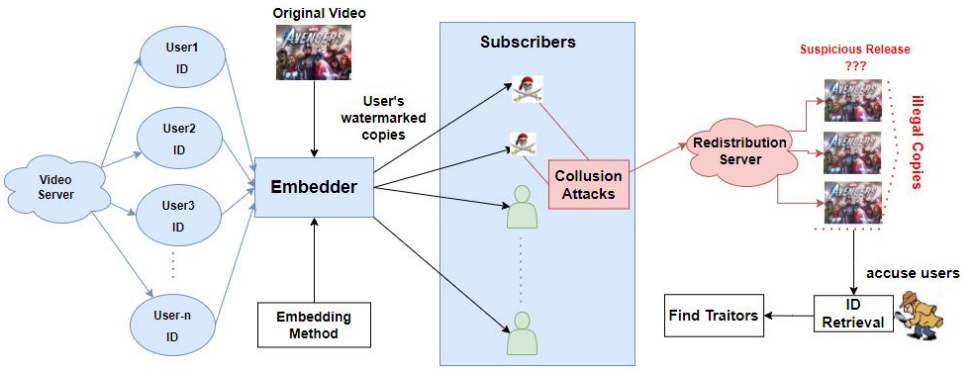


Figure 1.3 – In a VOD, tracing and terminating all the colluders from the suspicious release.

1.1.4.1 Binary collusion models:

Mathematically, a set of users $\mathcal{C}$ can participate in collusion by mixing their IDs. The collusion strategy or collusion attack defines the process that illegal users employ to create forged content and θ_b is the resulting watermark. A group of c dishonest users can merge their IDs, considering

matrix X contains all users IDs $\mathbf{X_C} := (\mathbf{x_{j_1}}, ..., \mathbf{x_{j_c}})$.

$$s_i := \sum_j x_{ji}, \quad j \in [1, ..., c], \quad i \in [1, .., m] \tag{1.1}$$

where m is the length of the ID. To create a suspicious ID y, different attacks can be performed by colluders.

— **Majority vote model**: The colluders set the majority symbol in their hands:

$$y_i := \begin{cases} 1, & \text{if } s_i > c/2 \text{ or } s = c; \\ 0, & otherwise \end{cases}$$

— **Minority vote model**: The colluders set the minority symbol in their hands:

$$y_i := \begin{cases} 1, & \text{if } 0 < s_i < c/2 \text{ or } s = c; \\ 0, & otherwise \end{cases}$$

— **All Zeros attack model**: The colluder can put 1 wherever they can: $s_i > 0$

$$y_i := (0, 1, 1, ..., 1, 1)$$

— **All one attack model**: The colluders can put a 0 wherever they can: $s_i < c$

$$y_i := (0, 0, 0, ..., 0, 1)$$

— **Coin flip model**: The colluders can toss a coin to choose whether to put 1 or 0:

$$y_i := (0, \frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}, 1)$$

Colluders can undertake a variety of other additional attacks, like average, interleaving, and so on. However, for the sake of comparing various attack models with real-time attacks on videos, we will limit ourselves to majority and minority. Because implementing these attack models for real video is easy [28]. Here is an example of a model of attack by majority and minority votes given in Table 1.1, the number of colluders $c = 3$, $\mathcal{C} = [c_1, c_2, c_3]$ with length $m = 10bits$.

The goal is to identify all potential colluders who engaged in a collusion attack to generate illicit content. In the literature, Tardos codes [29] are considered as state-of-the-art solutions to combat collusion. There are numerous code generators and decoders available depending on the settings and parameters. The specifics of these generators and decoders will be explained and compared in the next chapter.

Table 1.1 – An example of majority and minority collusion for $m = 10bits, c = 3$ users in collusion.

c_1	0	1	1	1	0	1	0	0	0	1
c_2	0	1	1	1	0	0	0	0	1	0
c_3	0	1	0	1	0	0	0	1	1	1
s	0	3	2	3	0	1	0	1	2	2
$\theta_b : Maj$	0	1	1	1	0	0	0	0	1	1
$\theta_b : Min$	0	1	0	1	0	1	0	1	0	0

1.2 Project Restrictions and Their Impact on the Thesis Target

The resistance against collusion is just a small part of the project; the ultimate objective is to develop a universal watermarking technique that is resistant to the following attacks:

— Format modification (4k,1080p,720p,480p,360p,240p): If an attacker changes the format of a video from its original format to erase or disrupt the watermark, the watermarking technique must be resistant.

— Bit-rate alteration (5Mbps− >125kpbs): If an attacker attempts to reduce the total bit-rate of a watermarked video, watermarking techniques should be able to counteract such an attack.

— Codec conversion (mp4(H.264, H.265,...), WMV): The watermarking technique should be resistant to any type of codec conversion.

— Incrustation (bars, logo, text): The watermarking method should be resilient, as the attacker may be tempted to hardcoat the video with bars, text, and logos to modify or deactivate the watermark.

— Colorization (saturation, gray-scale, contrast, etc.): Watermarking should be robust if an attacker attempts to adjust pixel intensity to remove the watermark.

— Degradation (bur, horizontal flip, noise, etc.): This is a popular signal processing attack in which the attacker attempts to add noise and distortion to the signals by altering them. The watermarking technique should be able to handle and resist any form of signal-processing attack.

— Geometric tampering (rotation, cropping, scaling, etc.): This is one of the most significant attacks on watermarking systems since it disrupts the general synchronization of the watermarked image. The watermarking approach should be resistant to any geometric attacks.

— Collusion attacks: According to section 1.1.4, pirates can use a variety of collusion attacks to create and distribute illegal video copies. This thesis contributes to the initiative to combat collusion attacks.

1.2.1 Project restrictions

A watermarking technique is subject to several restrictions. According to the literature [30, 31], frequency-based watermarking techniques can prevent the vast majority of watermarking attacks. So, the project also utilizes a frequency-based method(DWT-based watermarking). To fully link the thesis solution with the project, a few more constraints exist for the thesis objective. Using these constraints, the project was able to achieve resilience against most of the attacks, as outlined in 1.2. So, in this thesis, we must adhere to these constraints to give a solution for collusion attacks.

— *Watermarking technique:* In this thesis, we have to rely on the DWT-based watermarking technique.

— *Watermark image size:* For this thesis another constraint is the size of the watermark image, which is a $360p$ image with a size of $360 \times 640 = 230400$.

— *Capacity of watermark:* To achieve discrete watermarking, the watermark's capacity must be higher(close to 1).

1.2.2 Thesis target

The primary goal of this thesis is to provide a technique that is resistant to collusion attacks for videos. The thesis suggests using Tardos codes to avoid collusion attacks, taking into account the project's restrictions on watermarking technique, watermark size, and capacity. The objective of the collusion attack study is to identify all colluders among a million users in VOD, irrespective of the number of colluders involved in creating a suspicious copy. Given the large number of users, it might require an increased allocation of computing resources.

1.2.3 Research questions and answers

What is the optimal generator/decoder combo for collusion-resistant codes?

Despite the popularity of Tardos codes as a statistical solution to collusion, there exist a lot of open problems with Tardos code when utilized in real applications with multimedia content. Numerous researchers have utilized Tardos codes and provided various variants containing different enhancements, such as shorter code length, a better accusation decoder, and less complexity. Each of these code generator and accusation decoder versions has significance regarding the goal of accusation of colluders. The goal of accusation can be; 1) capture one. As the term says, catch at least one colluder; 2) capture all: catch all of the colluders. However, it is currently uncertain which variant for code generation and accusation decoder performs best in real time with videos. Since the project also has restrictions on the size of the watermark and the watermarking technology. It is critical to investigate all of these variations of the generator and decoder.

In this thesis, to contribute to this problem, we provide an in-depth review of the literature

for all types of code generators and decoders. We evaluate the performance of every variant in a binary model as well as real-time with videos. We provide the best possible combination for the code generator and decoder considering the restrictions of the project.

How to minimize complexity with collusion-resistant codes for a higher number of users?

It is critical to analyze if the project aim of finding all possible colluders from the suspicious video can be met using Tardos codes with manageable complexity and decoding time. Tardos codes have limitations in terms of code length and the number of colluders that can be traced. The goal is to find all possible colluders from a million users with the restriction on the size of the watermark. Using Tardos codes, as the number of users increases, so does the code length, decoding time, and complexity.

We conduct an extensive literature study on hierarchical code construction, to reduce decoding time and complexity for a higher number of users. Then, we suggest employing dynamic settings from code construction to further minimize decoding time. Finally, we compare the suggested method's performance to that of hierarchical construction. This concludes the possible settings for the Tardos code parameters that were attainable under project constraints.

How to improve the PSNR relative to discrete watermarking?

Considering the necessity for discrete watermarking, a watermark's invisibility indicates a higher capacity. As the capacity of the watermark increases, so does the level of noise, which leads to a decrease in Tardos code performance. To catch all colluders, we need a higher length of Tardos codes. Higher Tardos length results in higher BER for decoded Tardos code due to lower PSNR$= 20dB$ during discrete video watermarking

How to increase PSNR during watermarking, resulting in a lower BER for Tardos codes?

The project uses random orthogonal spreading for watermarking, which increases PSNR. Random spreading is derived from the spread spectrum in digital communication, which improves security by spreading the signal across a wide frequency band. However, the gain provided by random spreading is decreased as the Tardos code length is increased for a given watermark size. In addition to random spreading, we propose to investigate ECC to increase performance. We investigate convolutional codes with random spreading and present a novel combination of the two to improve performance even more.

Chapter 2

Collusion Resistant Codes

In this chapter, we will describe in detail the creation of codewords and the decoding process. In this chapter, we propose to find the best possible choice codeword generator/decoder combo, for that, we create and test different combinations. Then, we perform some experiments to find the best combo concerning colluder detection and computational cost. In addition, to address detection rate and complexity, we study the literature on hierarchical construction and perform experimentation. Then, we propose an alternative setup for hierarchical code construction given the goal of improving the detection rate w.r.t. complexity.

2.1 User's codeword generator/deoceder

The fingerprinting in the content protection emphasizes collusion resistance. The fingerprinting layer is divided into three components: (1) code generation, which involves creating codewords; and (2) code decoder, which involves mapping the mixed codeword to a group of accused users as shown in Figure. 2.1. This description is fairly broad and applies to (nearly) every known fingerprinting layer inside content protection. The choice of different functions and thresholds in the generation and decoding components distinguishes one scheme from another. Here's a detailed rundown of the processes of both generation and decoding that distinguish one scheme from another.

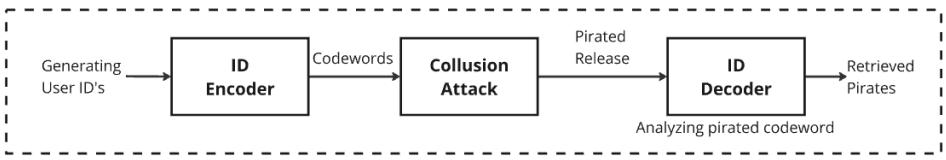


Figure 2.1 – A binary approach for tracing pirates: create user codewords using a code generator and analyze the pirated release using a code decoder.

2.1.1 Codeword generators

The main idea is to create as many codewords as there are users and assign each user a unique codeword. The efficiency of such codewords is assessed by the number of colluders who

participate in collusion attacks and can be accused with enough confidence while having little probability of accusing an innocent person. This is accomplished by using *collusion-resistant codes*, also known as *traitor tracing codes*, which give robustness against collusion assaults. The purpose of collusion-resistant codes is to generate codewords so that, regardless of how strong the collusion attacks are, the final copy still includes sufficient data to recognize the colluders.

In the literature, firstly in [32], Boneh and Shaw proposed marking assumptions and a well-known two-level binary collusion-resistant code that combined a partially randomized inner code with a deterministic outer code. Based on marking assumptions, we can divide the codes into two types: poor traceability and high traceability. With a poor traceability code, it is possible to accuse and frame innocent users with a low probability. However, it must effectively identify at least one colluder for high traceability while never framing an innocent. The code length for the Boneh and Shaw scheme is $m = {c_o}^4 \log \frac{n}{\epsilon_1} \log \frac{1}{\epsilon_1}$, where n is the number of users, ϵ_1 is the chance of a False Positive (FP) error and c_o is maximum number of colluders. They also gave a lower bound on the expected code length $m = \mathcal{O}(c_o \log \frac{1}{c_o \epsilon_1})$. Later these codes were improved by [33] for multimedia content by setting more realistic assumptions.

Later on, Tardos [29], provided a significant improvement in the field of collusion-resistant codes. He provided collusion-resistant codes based on a probabilistic approach. Tardos codes have a theoretically minimal code length, which is ideal for any collusion attack. Tardos showed a tighter bound of $m = \mathcal{O}({c_o}^2 \log \frac{1}{\varepsilon_1})$, and he gave a fully randomized binary code with $m = 100{c_o}^2 \log \frac{1}{\varepsilon_1}$, that achieves that bound.

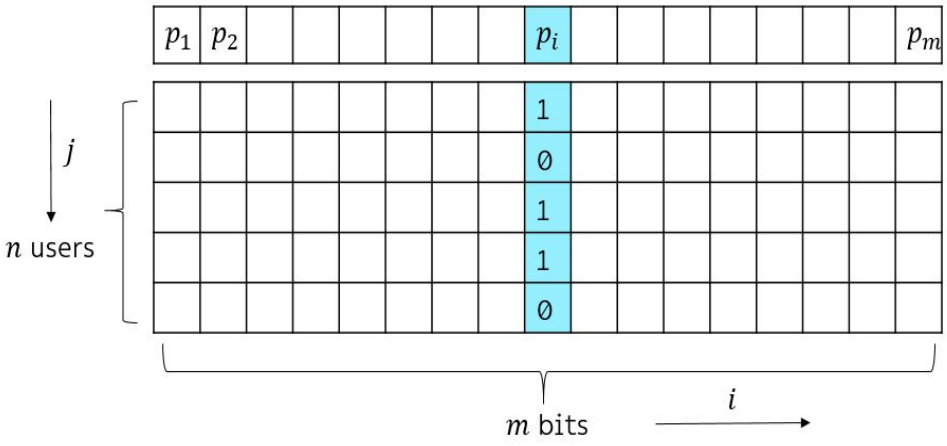


Figure 2.2 – Code matrix $\mathbf{X}$ created by content owner

The binary Tardos codes can be generated as follows. The content owner creates a $n \times m$ binary matrix $\mathbf{X}$ where each row $\mathbf{x_{j:}}$ corresponds to a codeword for user j as shown in Figure. 2.2. Each entry x_{ji} in column i (with $i = 1, ..., n$) follows a Bernoulli distribution with parameter p_i. Parameters p_i are drawn from an arcsine distribution [29] and are denoted random biases.

For Tardos codes, the random bias generation is detailed in Algorithm 1.

Algorithm 1 Random bias for Tardos codes

1: let $t = \frac{1}{300c_o}$, and let $t' = arcsin\sqrt{t}$.
2: $\forall i \in [1, m]$; draw a random r_i according to uniform distribution in $[t', \frac{\pi}{2-t'}]$.
3: $\forall i \in [1, m]$; calculate $p_i = sin^2(r_i)$.

Due to the optimal asymptotic performance of Tardos codes as a probabilistic solution to collusion, numerous researchers have utilized them and provided variants containing different enhancements, such as shorter code length, a better accusation decoder, and less complexity. We limited ourselves to smaller code lengths in this thesis because of watermark image length restrictions. For a given image size, increasing the code length leads to higher BER.

2.1.2 Reduction of Tardos code length

Blayer and Tassa explored [34] an approach to reducing code length by enhancing parameter selection for Tardos codes. They replaced Tardos code constants with parameters in their initial setting, resulting in a set of inequalities that those parameters had to satisfy. They followed by searching for a solution to the inequalities that would reduce the code length m. In this way, the code length can be divided by 4. Later on, Skoric et al. [35] reduced the code length further. Skoric et al. proposed a generalization for binary alphabets, that makes use of a Dirichlet distribution and, as stated, for a given alphabet size, it reduces to the arcsine distribution. Furthermore, they observed that, for the colluder size c sufficiently large, the accusation sums of the innocent user and of the colluder have nearly Gaussian probability distributions. They also indicated that if these distributions are completely Gaussian, then for binary alphabets with given c_o colluders they proposed a smaller code length of $m \approx \pi^2 {c_o}^2 \ln \frac{1}{\epsilon_1}$. Furthermore, they showed, by invoking the Central Limit Theorem, that an even tighter code length of $m = \frac{1}{2}\pi^2 {c_o}^2 \ln \frac{1}{\epsilon_1}$ is sufficient in most cases.

Hagiwara et al. [36] reduced the Tardos code length for a limited number of colluders and provided an initial framework for the discrete Tardos code. Based on this framework Nuida et al. [37] proposed a discrete distribution for p_i that is based on c_0. If the actual collusion size $c \leq c_0$, the modified discrete distribution enhances decoding; however, when the collusion size $c > c_0$ the modified discrete distribution has lower performance.

Following that, Amiri and Tardos presented [38] a higher rate of fingerprinting codes by integrating these two approaches [39, 29]. They showed that the rate of their code reaches the fingerprinting capacity and provided a precise estimate of the rate of the fingerprinting code as the m approaches infinity. Nevertheless, the high computational cost of their fingerprinting code's accusation process makes it difficult to apply in practice.

As opposed to these theoretical contributions, Furon et al. [40] developed an experimental method for estimating the smallest code length of binary symmetric Tardos code. They investigated the worst-case attack that the colluders might conduct and then utilized a rare event analysis approach to estimate the error probabilities ϵ_1 and ϵ_2, where ϵ_2 is the probability of not accusing a colluder. In the end, for a specific collusion size, they could predict the minimum length of code that met certain error probability conditions. Their practical results showed that the estimated code lengths were significantly less than the previously known theoretical bottom limitations. Their practical results showed that the estimated code lengths were significantly lower than the previously established theoretical lower limits. In this thesis, our implementation of the Tardos code is based on this improvement.

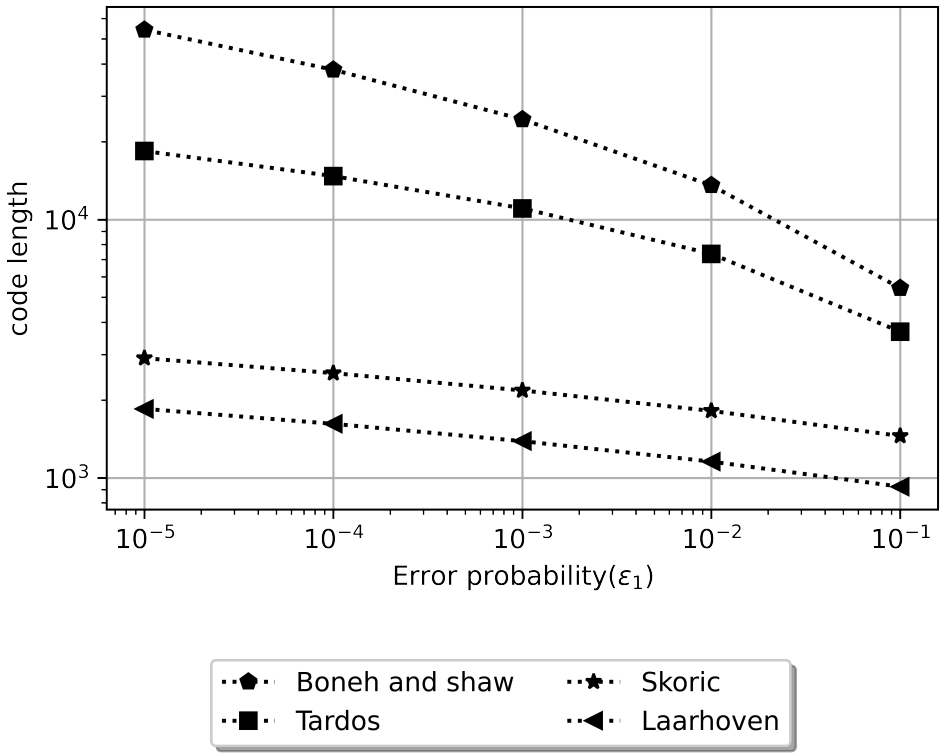


Figure 2.3 – Comparison of code lengths for different the codeword generation methods suggested in the literature, assuming $n = 1000$ and $c_o = 4$.

Finally, Laarhoven and de Weger in [41] proved that Nuida discrete distribution [37] asymptotically converges to the arcsine distribution of Tardos codes. Laarhoven construction gives codes that are up to 4 times shorter than Blayer and Tassa's, and up to 2 times shorter than the codes from Skoric et al. For Laarhoven codes, the bias distribution is detailed in Algorithm 2. The code length depends on the generating method parameters, error probability (ε_1), and

number of alleged colluders (c). However, it should be kept as low as practical. If the goal is to accuse all colluders while accommodating a higher number of users, the code length should be increased accordingly. Figure. 2.3 compares the code length for different codeword generation methods suggested in the literature.

Algorithm 2 Random bias for Laarhoven codes

1: $\forall i \in [1, m]$; draw r_i according to uniform distribution from $\left(\frac{3\pi}{(8c_o+4)}, \frac{7\pi}{(8c_o+4)} \cdots \frac{\pi}{2}, -\frac{3\pi}{(8c_o+4)}\right)$,
2: $\forall i \in [1, m]$; calculate $p_i = sin^2(r_i)$.

2.1.3 Decoding

Let $\mathbf{y}$ denote the extracted codeword bit vector corresponding to a collusion attack, and thus not to a legitimate user identifier. In this context, decoding means to retrieve the legitimate users who participated in the generation of the illegitimate copy. To this aim, scores are calculated using some scoring function g: $s_{ji} = g(X_{ji}, y_i, p_i)$. User j is declared as a colluder if the $\sum_{i=1}^{m} s_{ji} > Z$ for a given threshold Z. Another strategy consists of accusing the users with the greatest cumulative scores.

Along with Tardos codes, the optimal scoring function without embedding, regardless of the collusion attack, denoted as the Tardos scoring function, was proposed in [29], and read as:

$$g(X_{ji}, y_i, p_i) := \begin{cases} +\sqrt{\frac{1-p_i}{p_i}} & \text{if } X_{ji} = 0; \\ -\sqrt{\frac{p_i}{1-p_i}} & \text{if } X_{ji} = 1; \end{cases} \tag{2.1}$$

This scoring function is to be compared with a threshold $Z = 20c_o \ln \frac{1}{\epsilon_1}$ to accuse user j.

Then it's generalized in [35, 42] by replacing various fixed numerical parameters with variables. The Tardos-Skoric function is provided in [35] reads as:

$$g(X_{ji}, y_i, p_i) := \begin{cases} +\sqrt{1 - {p_i}^{2y_i - 1}(p_i)^{1-2y_i}} & \text{if } X_{ji} = y_i; \\ -\sqrt{{p_i}^{2y_i - 1}(1 - p_i)^{1-2y_i}} & \text{if } X_{ji} \neq y_i; \end{cases} \tag{2.2}$$

When the maximum number of colluders c_{max} can be defined and assuming apriori information about the collusion attack represented by parameter θ_c, an optimal decoder is provided in [43], and will be denoted *Desoubeaux* decoder hereafter. The Z can be chosen based on the code generator and parameter settings. The Desoubeaux scoring function is reminded in [43] reads as:

$$g_{\theta_c}(X_{ji}, y_i, p_i) := \log\left(\frac{\mathbb{P}(Y = y_i | X_{ji}, p_i, \theta_c)}{\mathbb{P}(Y = y_i | p_i, \theta_c)}\right) \tag{2.3}$$

Here θ_c is the collusion model and the probabilities can be found in [40, Eq. (8-9)]. The users with higher scores will be accused as colluders. The decoder presented in [44] is still based on

the knowledge of c_o but without any need for apriori information on the collusion attack. We will refer to this score as the *Laarhoven* score. The Z can be chosen based on the code generator and parameter settings. It is described in [44] given by:

$$g(X_{ji}, y_i, p_i) := \begin{cases} \log\left(1 + \frac{1}{c_o}\left(\frac{1-p_i}{p_i}\right)^{2y_i-1}\right) & \text{if} \quad X_{ji} = y_i; \\ \log\left(1 - \frac{1}{c_o}\right) & \text{if} \quad X_{ji} \neq y_i; \end{cases} \tag{2.4}$$

The last decoder considered will be denoted as NNS and was proposed in [45] as the nearest neighborhood search. The NNS score does not require any a priori on c_{max} nor the collusion attack and is detailed in [45] given by:

$$g(X_{ji}, y_i, p_i) := \frac{(2X_{ji}-1)(2y_i-1)}{\sqrt{p_i(1-p_i)}} \tag{2.5}$$

The Z can be chosen based on the code generator and parameter settings. The decoding time and computing resources are crucial criteria to consider when determining a decoder's efficiency. Each decoder has a different complexity. The complexity for all the above-described decoders is given in Table 2.1, where $\rho \leq 1$ is determined by the initial settings of the fingerprinting scheme [45], and $k \leq c_{max}$. From Table 2.1, we can foresee that the NNS score decoder is the less time-consuming one. Note also that, if the decoder requires θ_b, it will be impractical, even if it performs well, since the θ_b will not be known at the decoder side in a practical scenario.

Table 2.1 – Decoding complexity for computing $g(X_{ji}, y_i, p_i)$ for i, j with various decoders.

Decoders	**Decoding complexity**
Tardos-skoric	$O(mn)$
Laarhoven	$O(mnc_o)$
Desoubeaux	$O(mn^k c_o \theta_b)$
NNS	$O(mn^\rho)$

2.2 Best Generator-Decoder Combination

As outlined in the literature, we consider the two ID-generating algorithms, Tardos and Laarhoven, and the decoders, Tardos-Skoric, Laarhoven, Desoubeaux, and NNS. We need to find the best generator-decoder combination based on decoding difficulty and accusation rate. In this section, we will present a simulation model and conduct experiments to compare generators and decoders.

2.2.1 Experiments and results

We investigated majority and minority vote collusion attacks, as provided in Section 1.1.4, because both these attack models are easier to stimulate with real-time using videos. In binary mode, the idea is to perform majority and minority vote collusion attacks on the codewords of the randomly selected traitors. We randomly choose $c = 12$ traitors in each simulation and apply the collusion attack model θ_b. The decoder will use the colluded bit vector, together with p_i and $\mathbf{X_{ji}}$, as illustrated in Figure 2.4. For 100 Monte Carlo simulations with a fixed X matrix and selecting random colluders more than one colluder is traceable even with several users $n = 100$. We noticed that Laarhoven and Desoubeaux's accusation is better in comparison with other decoders as shown in Table 2.2 and 2.3. The literature [40] also suggests utilizing the Desoubeaux decoder.

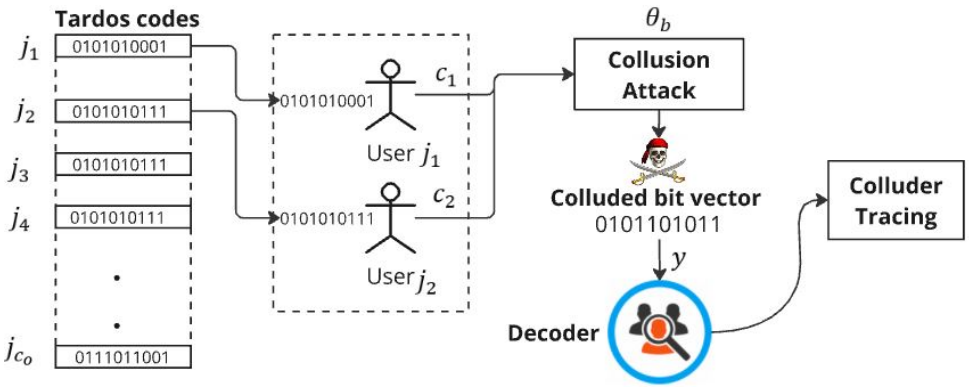


Figure 2.4 – Simulation model to emulate the collusion of c_o colluders and its tracing thanks to Tardos codes.

Table 2.2 – Average detected colluders using different decoders for Tardos codes with $c = 12$

Attack	n	**Tardos-Skoric score**	**Laarhoven score**	**Desoubeaux score**	**NNS score**
Majority	30	3.95	6.86	6.65	5.81
	50	3.85	4.87	4.96	3.95
	100	1.91	1.96	2.86	2.91
Minority	30	3.96	6.62	6.56	5.69
	50	3.83	4.53	4.48	3.64
	100	1.71	1.78	2.85	1.79

Table 2.3 – Average detected colluders using different decoders for Laarhoven codes with $c = 12$

Attack	n	**Tardos-Skoric score**	**Laarhoven score**	**Desoubeaux score**	**NNS score**
Majority	30	4.81	6.62	6.96	5.92
	50	4.62	4.79	4.81	4.98
	100	1.89	1.81	2.82	2.89
Minority	30	5.77	5.32	7.96	5.83
	50	3.96	4.96	6.51	3.62
	100	2.00	1.98	2.98	2.99

2.2.2 Discussion and conclusion

About the choice of the Bias distribution: After completing several trials, we determined that the Laarhoven discrete bias distribution function performs slightly better than the Tardos one, as shown in Table 2.2 because of the value of c_o, which is stated in the case of the Laarhoven bias distribution function. We recommend the Laarhoven discrete bias distribution version over the Tardos continuous bias distribution for a given c_o.

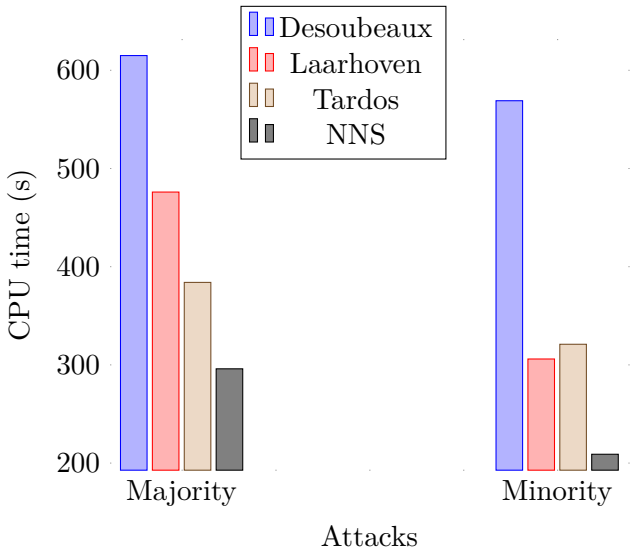


Figure 2.5 – Tracing time for all the decoders for two attacks over Tardos codes with parameters $n = 100$, $c = 12$ averaged over 100 *trials*

About the choice of the decoder score: The Desoubeaux decoder outperforms all previous decoders relying on the collusion attack model θ_b. Furthermore, as shown in Figure.2.5, usage of the Desoubeaux decoder is challenging due to the longer time required to trace the colluders. In this section, the complexity of computation is compared by evaluating the time consumption on a machine loaded with a Core-i5-CPU@1.6GHz (8 CPUs) and 16 GB RAM.

2.3 Hierarchical Code Construction

2.3.1 Importance of hierarchical code construction

One of the challenges of using Tardos code is its exhaustive detection. The detector needs enough computing power to calculate the scores for all prospective users. The detection procedure has a complexity of $O(m \times n)$, reflecting an extensive computational cost. As a result, as the number of users increases, so does the cost of detecting collusion. Yet, the detection method may be improved by lowering the computing expenses. Since correlation scores must be calculated for all possible code words, the number of operations grows linearly with the number of users. To significantly reduce the cost, an N-hierarchical construction can be utilized when certain users are more likely to collude than others, depending on their hierarchical constraints.

2.3.2 Generation and decoding process

In a hierarchical code construction, users may be divided into groups. The idea is to reduce the number of users per Tardos code to improve the pirate tracing performance [4]. Inside the groups, there will be a set of users who are more likely to engage in collusion as shown in Figure. 2.6. As a result, the total codeword of users before embedding will be a mix of the group identification and the user's codeword as shown in Figure. 2.7. Rather than assessing all of the user codewords, we can first decode the group identification (guilty group) that most closely resembles the colluding group identifier. Then, we can investigate the collusion inside the detected group.

When there is no collusion between the groups, the problem is relatively easy; however, when there is such a collusion, the problem gets more complicated. As a result, to manage the problem, we must also utilize Tardos codes for group identifiers. When employing Tardos codes for group identifiers, it is necessary to examine and determine the best identifier length.

The authors of [4] suggested a hierarchical Tardos code structure for both group identifiers and user codewords to determine the best length of a group identifier m_g. Assuming that the distribution of correlation scores for codewords is Gaussian, Equation 2.6 stands that parameters β, ϵ_g, and ϵ_u can be used to derive an acceptable m_g given the user codeword length m_u, where ϵ_g is the false-positive probability for groups and ϵ_u is for users. The parameter $\beta = \frac{c_g}{c_u}$ takes into account the number of tolerable colluders for groups (c_g) and the users (c_u). Using Equation 2.6 from [4], we can calculate the shortest possible length for group identifiers.

$$\frac{m_g}{m_u} := \beta^2 \times \frac{\ln \frac{1}{\epsilon_g}}{\ln \frac{1}{\epsilon_u}} \tag{2.6}$$

A four-level hierarchical structure has been proposed in [46] for DCT based audio watermarking. This approach reduces complexity and decoding time while maintaining a high detection

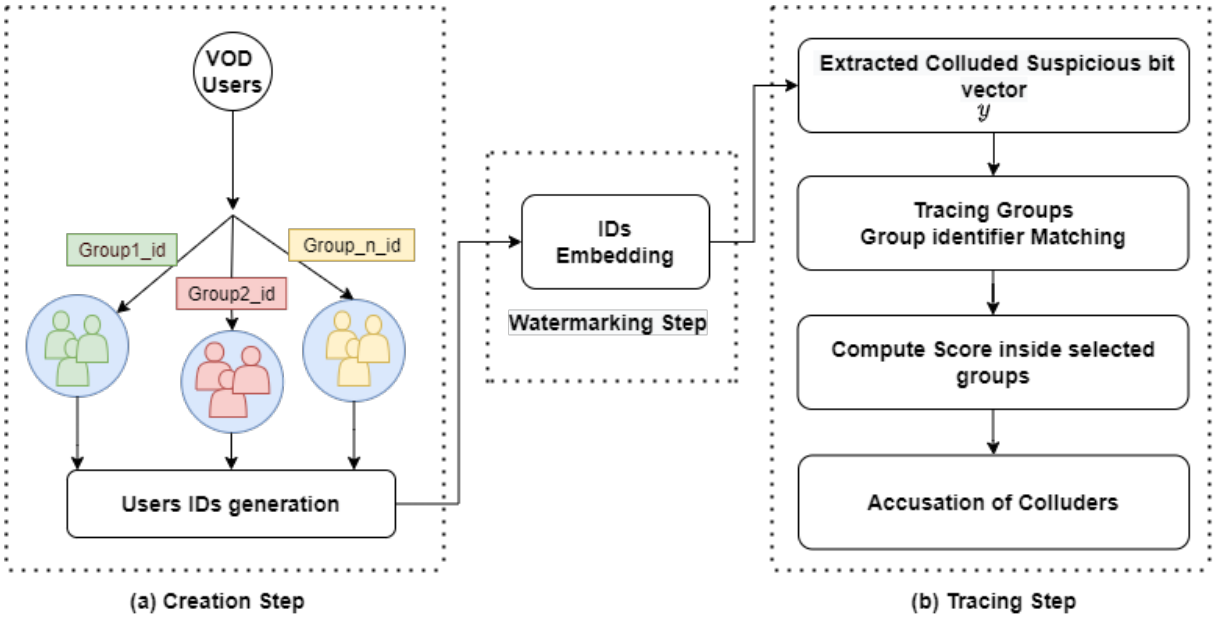


Figure 2.6 – (a) Creation step: Using constraint, a hierarchical system splits users into groups that are more likely to collude. (b) Tracing step: After finding a colluded fingerprint, locate the group and then trace the colluders inside that group.

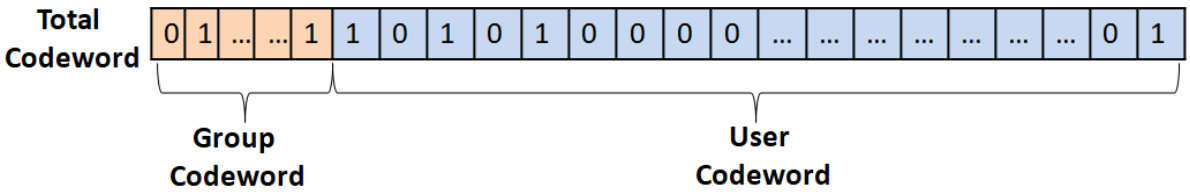


Figure 2.7 – Total codeword before embedding step, obtained by concatenating group codeword and user codeword.

rate. However, they apply the collusion model in the tracing process and fail to tackle the collusion between groups. This is not ideal because the collusion model will be unknown in a real-time setting. There is a trade-off between the total codeword length and the number of groups. The number of groups in a hierarchical architecture increases the length of the fingerprint. It is also challenging to handle and hide such large amounts of data in multimedia without errors.

According to [47], increasing the number of groups increases codeword length but lowers the number of operations, leading to a gain in cost. They overlooked to take into account the noise in the real-time transmission. The BER increases with the noise level. Additional research [48, 49] uses techniques such as clustering to better integrate users inside groups. It shows an increase in detection rates, but also a decline in the context of group collusion. Based on the findings in the literature, utilizing hierarchical architecture allows for an improved detection rate while also reducing complexity. We performed experiments based on the optimal generator-decoder combination, as stated in Section 2.2.2.

2.3.3 Experiments and results

To demonstrate the performance gain of the generation-decoder combo, we will simply consider g number of groups. For non-hierarchical construction, we built the user code using Tardos(n, p, X) codes, then performed a majority vote collusion attack and traced back the colluders. However, for hierarchical construction, we must generate Tardos(g, p_g, X_g) codes for group identifiers and additional Tardos(n, p_u, X_u) codes for user codewords, as shown in Figure. 2.8. Then, split the users within every group by $\frac{n}{g}$. We assume that users are distributed uniformly throughout each group. For the tracing process, we must first trace the groups participating in the collusion and then identify users inside those groups.

We set the parameters for user codewords creation $n = 10^3, c_u = 7, \epsilon_u = 10^{-4}$, which results in $m = 4460$. The parameters are set based on the following setup in the real world for a $360p$ watermark image. The Equation 2.6 can be used to specify the parameter for group codewords. By taking $\epsilon_g = 10^{-2}$ and $c_g = 4$ into account, $m_g = 730$ is obtained. For the comparison between hierarchical construction of codewords with non-hierarchical under the condition the length of codeword is equal, $m = 4460(m_g = 730, m_u = 3730)$ and the false-positive error probability $\epsilon_1 = 1 \times 10^{-6} = (\epsilon_g = 10^{-2}, \epsilon_u = 10^{-4})$. Given the number of users $n = 10^3, 10^4$ and 10^5, we show the average number of colluders that were detected in both hierarchical and non-hierarchical codeword constructions. The computational costs of the 4-hierarchical are contingent upon the number of detected group identifiers, whereas the non-hierarchical Tardos code construction is constant. Depending on the distribution of the colluders, the hierarchical construction increases the detection rate even for larger numbers of colluders and significantly reduces the computational costs as shown in Figure. 2.9.

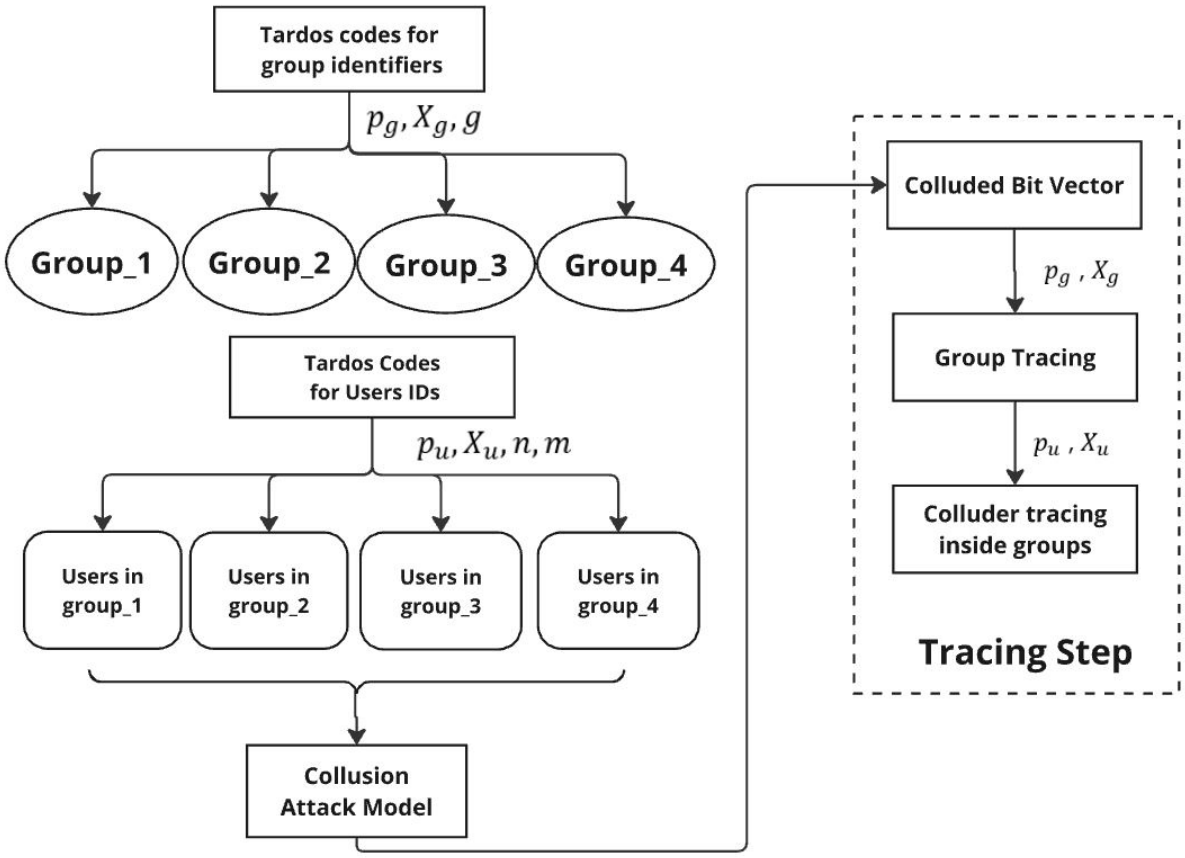


Figure 2.8 – The simulation model for the 4-hierarchical construction of codewords.

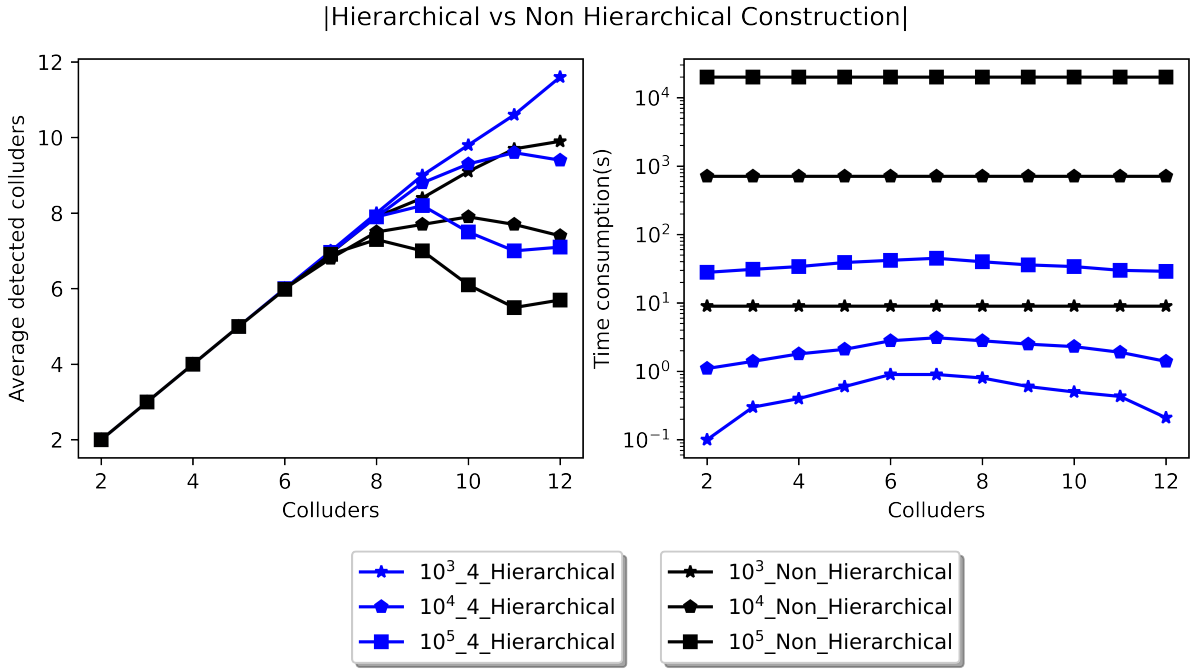


Figure 2.9 – The comparison between hierarchical codeword construction with non-hierarchical based on majority vote attack.

2.4 Proposed Hierarchical Construction

2.4.1 Generation and decoding process

It's shown in [4, Figure. 8], using hierarchical construction reduces computation cost, but the number of detected groups drops, as c_o increases. Considering our target of a big $n = 10^6$, depending on the number of groups and colluders inside the groups, it is hard to prevent collusion among all groups for hierarchical construction. We proposed an alternative hierarchical code construction in which a unique Tardos code is generated for a group of random users. Tardos code will be generated for each group using an independent p, which helps in randomizing the resulting X matrix. The proposed hierarchical construction will assist in eliminating the need for a distinct Tardos code for group identifiers. The result allows for accommodating more users and prevents collusion between groups, but the computational cost will rise as the number of groups rises.

2.4.2 Experiments and results

For the experiment, we used a similar technique to a hierarchical construction, but instead of creating Tardos codes for group identifiers, we just created separate Tardos codes for users inside each group. In the tracing step, the decoder needs all the user's codeword information and all the distribution bias as shown in Figure. 2.10.

Given the number of users $n = 10^3, 10^4$ and 10^5, we show the average number of colluders that were detected in both dynamic hierarchical and non-hierarchical codeword constructions. The computing costs of the 4-hierarchical dynamic configuration vary with the number of searches for the groups, but the non-hierarchical Tardos code construction remains constant. Figure 2.11 shows that dynamic hierarchical structure not only boosts the detection rate but also increases the computational cost. The cost rise is primarily due to collusion between groups. If we have users from a single group, we just need to search once, but the computing cost grows as the group's collusion increases. In the end, dynamic hierarchical construction outperforms simple hierarchical construction in terms of detection rates. However, the complexity increases when compared to a simple hierarchy, although it remains lower than non-hierarchical.

2.4.3 Real-time Hierarchical constraints

There are multiple constraints in implementing the hierarchical code creation in real-time. These constraints are entirely dependent on the environment and needs of the content provider. Hierarchical constraints include cultural, geographical, and social constraints. These constraints are determined by the content provider depending on their requirements. It is more likely that a dishonest user will collude with someone who shares similar social, cultural, or geographic

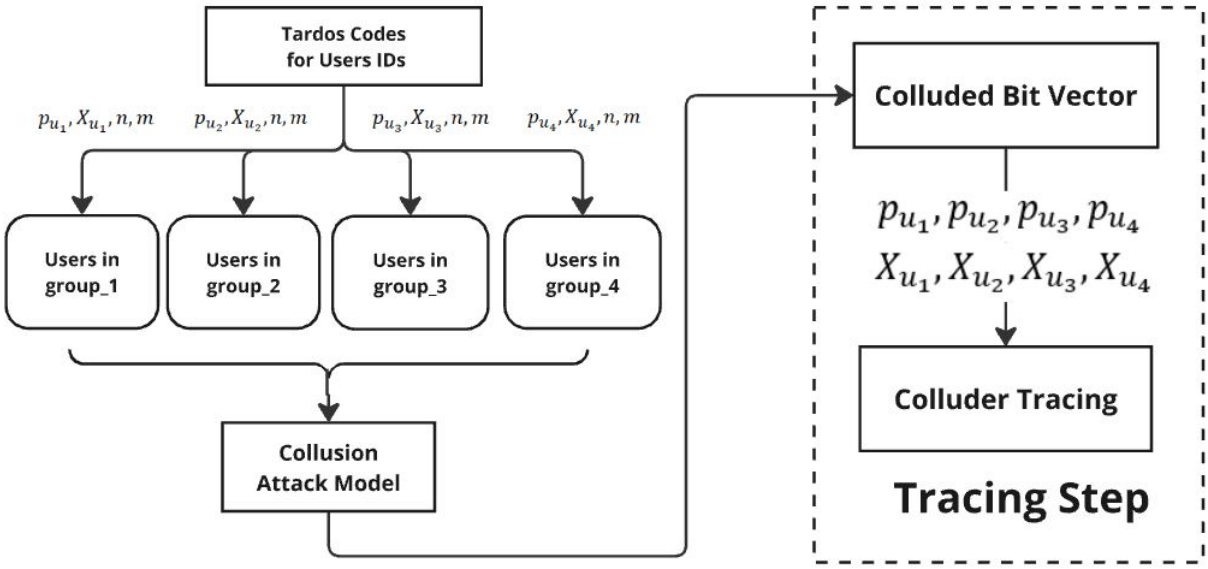


Figure 2.10 – The simulation model for the 4-hierarchical dynamic construction of codewords.

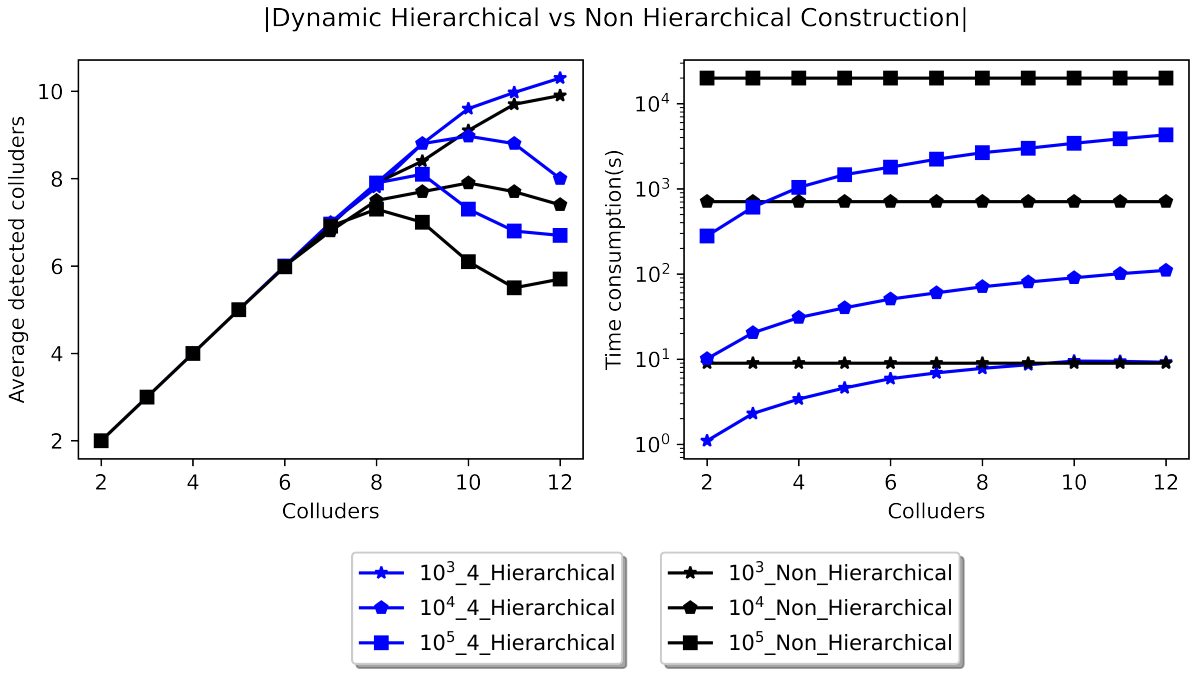


Figure 2.11 – The comparison between dynamic hierarchical codeword construction with non-hierarchical.

constraints. The cultural constraints refer to the frequency with which viewers access a video. The frequency with which a video is accessed varies according to culture and popularity. Curiosity is another aspect that contributes to cultural constraints since viewers are more interested in the top ten videos on VOD platform. Over four months, the viewer's frequency of access decreases from 100% to 10% as shown in [50], which is why the frequency of access is so vital constraint for hierarchical construction.

Furthermore, certain locations or nations are well-known for having the highest rates of multimedia piracy, therefore consumers in these places will be more likely to collude than others. This is when geographical constraints come into play, as shown in [51]. Additional constraints can be the social parameters such as the age and gender of the users. As shown in [52], the women are more interested in the VOD platform, probably, they are also more engaged in privacy than males. As a result, a specialist needs to define the hierarchical constraints: users who are more inclined to collude should be placed in an identical subset. Then groupings of subsets who are more inclined to collude are combined into a single group. This idea serves as the foundation for further categorization. This allows content providers to detect the collusion with more flexibility. Of course, pirates can examine a variety of additional factors. Pirates, for example, can hide their locations by using a VPN service. This may be investigated further when the constraints are imposed by the content provider.

2.5 Summary and Conclusion

In this chapter, we studied different collusion code generators/decoders based on Tardos work. We compared the Tardos and Laarhoven as generators with Tardo-Skoric, Laarhoven, Desoubeaux, and NNS as decoders. To choose the most suitable generator-decoder, we evaluate performance in terms of detection rate and complexity. Simulations demonstrate that utilizing Laarhoven codes decoded by the Desoubeaux decoder provides the highest performance, although with more computational cost. Similarly, employing Laarhoven codes decoded using NNS results in decreased complexity and moderate detection as shown in Section 2.2.

To further reduce the complexity, we use hierarchical code creation, which reduces decoding time. We provided detailed literature on the hierarchical structure and conducted several tests based on optimal generator-decoder combinations. In addition, we proposed an alternate hierarchical code construction that considerably improves detection while incurring just a slight increase in computation cost.

Chapter 3

Random Spreading and Collusion Resistant Codes

A robust and secure multimedia content protection solution must utilize a strong and secure watermarking approach to include the user's IDs. However, as far as we know, developing a robust and secure watermarking strategy is challenging, as robustness and security are two distinct concepts in the watermarking industry.

Robust watermarking is a technique that should be robust to routine signal processing operations without a particular strategy. However, some attacks are based on alternating with the watermarking technique. As a result, during these attacks, the attacker takes precise action to perform a strike to erase the watermark with as little distortion as possible. These two features of watermarking affect and restrict one another. For secure watermarking, the messages are covered with additional bits, called redundant data, which are processed and transferred across communication channels.

For secure and robust watermarking with lower BER, we need to add extra redundancy to Tardos codes to obtain greater tracing rates. In this chapter, we investigate approaches for adding redundancy to user ID before embedding it in multimedia. The well-known method is random spreading, which generates a random sequence to represent the single bits in the input message. In our case, the output length depends on the video resolution, so, a higher input message reduces the spreading rate. However, the random spreading has limitations with higher input messages for fixed output. For this reason, we further explore the usage of ECC coupled with random spreading, to reduce BER and then to improve the overall tracking of colluders.

3.1 Spreading Techniques with Tardos Codes

3.1.1 Motivation: Impact of binary errors on Tardos codes

In the collusion-resistant watermarking technique, a Tardos code of length m is hidden in an image to trace the colluders. Tracing a higher number of colluders among many users requires a larger m. The maximum value of m is defined by the video resolution. By default, a $360p$ image corresponds to the image size given by $m_{max} = 360 \times 640$=230400. For this default setting,

considering Tardos code length is $m = m_{max}$, we will not be able to recover the Tardos codes without error, because of higher BER after transmission. This leads to a declined rate of collusion tracing. That's why we have to find a possible trade-off between m and BER.

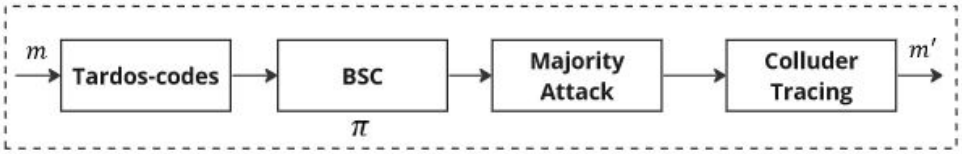
In this thesis, we want the watermarking method to be discrete. To obtain discrete watermarking, the video is blended with the watermark image with the lowest possible capacity, leading to a ratio between our watermark (called the signal in the following) and the video medium (interpreted as the noise), or PSNR of about -20dB, depending on the considered content. For such a low PSNR, many errors occur in the ID codeword, which leads to dramatic performance for collusion tracing with Tardos codes as shown in Figure 3.1(b). We analyzed the performance of Tardos codes using the majority vote attack to trace out the colluders. The simulation model is depicted in Figure 3.1(a): Tardos codes are modified by a BSC with error probability π, which represents the possible errors due to the blending with a low PSNR. A scheme that can help understand errors inside the ID during watermarking can be analyzed by BSC. The BSC [53] is a common communication channel model used in coding theory and information theory. In this channel, the transmitter sends a bit (either a '0' or '1'), and the receiver receives a bit. The bit will be "flipped" with a "crossover probability" of π, but it is otherwise received successfully. This representation could potentially be used here as errors inside IDs. This rearrangement of bits will introduce some level of errors based on π. Figure 3.1(b) illustrates that, whatever the ID length, the number of average detected colluders drops for binary error probabilities higher than $\pi = 2 \times 10^{-1}$, which corresponds to a much higher PSNR than -20dB.

A well-known way to improve PSNR is to add random spreading to the ID length m over the image length k. Let α denote the spreading rate $\alpha = \frac{k}{m}$. A lower α results in fewer binary errors on the watermark image, but as the image length is fixed, it also results in lower ID m, reducing total colluder detecting capacity.

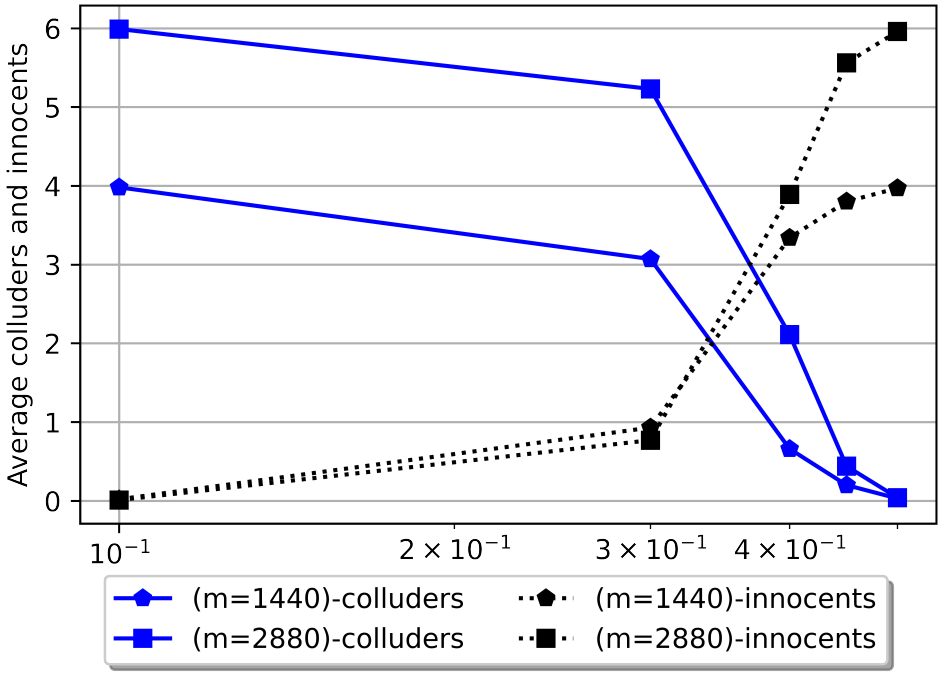
3.1.2 Pseudo-random spreading technique

The first spreading technique was proposed by Cox [54], one of the most important breakthroughs in the field of watermarking, and provided high robustness to different signal processing alterations. In this technique, the pseudo-random generator creates a pseudo-random sequence based on the secret key, which always acts as the secret carrier. Following that, the embedder mixes this secret carrier based on the message to be hidden to generate the watermark signal, which is then added to the original host signal.

In this way, the watermark hides covertly, prohibiting unwanted access or removal. Later on, there was some improvement made by [55, 56], based on the scale and length of sequences. Nowadays, the random spreading approach is frequently utilized; several embedding systems are based on it. This is the reason why numerous studies take advantage of random spreading [57, 58, 59] to hide data into images with low errors after retrieving the data. Considering the need



(a) Simulation model



(b) Simulation results

Figure 3.1 – Colluders tracing without spreading scheme for majority vote attack: (a) Simulation model (b) Average detected colluders with $m = [1440, 2880], n = 1000$ and $\varepsilon_1 = 10^{-3}$.

to create a discrete watermarking method for ID that has higher PSNR due to lower opacity while embedding. So, based on the previous research, we will focus on random spreading in this thesis to improve BER.

The random spreading of x_j is based on α. Here α represents the length of the random sequence for each ID code symbol. (e.g. for a image length $k = w \times h$, $\alpha = \frac{k}{m}$). First, a dictionary $\mathcal{D}$ has to be created so that it contains the random sequences d for the size (ℓ, α) based on the secret key. Each number of symbol ℓ of ID code x_j is represented by d sequences as shown in Figure. 3.2. The resultant matrix Z has dimension $w \times h$ and is partitioned into independent m sub-matrices. Each vector d with length α sequences from $\mathcal{D}$, corresponding to ID code symbols.

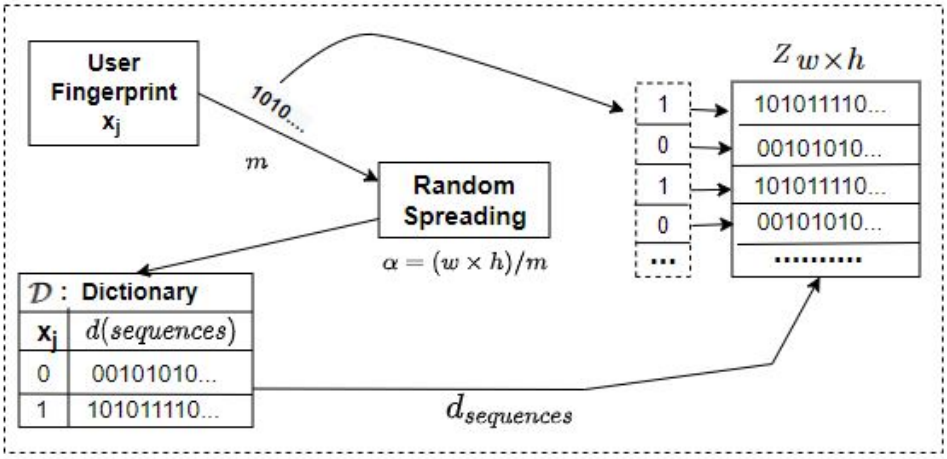


Figure 3.2 – Random spreading of Tardos code with rate α.

To reconstruct the ID of users from extracted sequences $\hat{d}$ in $\hat{x}_j$, perform a Pearson Correlation between extracted sequences $\hat{d}$ and original d sequences from $\mathcal{D}$. The Pearson Correlation for two objects with paired qualities is calculated by adding the product of their differences from their object means and dividing by the product of the squared differences from the object as shown in the Equation. 3.1.

$$c_r(d, \hat{d}) := \frac{\sum (d_i - \hat{d}_i)}{\sqrt{\sum (d_i - \hat{d}_i)^2}} \tag{3.1}$$

The following produces a score that ranges from -1 to +1. Two items with a high score (around +1) are quite similar. Two uncorrelated items would have a Pearson score close to 0. An inverse correlation between two items yields a Pearson score of roughly -1. In this case, we want the correlation between the extracted and original sequences to be close to +1, allowing us to properly rebuild the ID symbols.

3.1.3 Experiments and results

We evaluated the performance of proposed random spreading with Tardos codes when facing a majority vote attack. Figure 3.3(a) depicts the simulation model, in which a BSC modifies the random spreading of Tardos codes with error probability π. The colluders are then traced by applying a majority vote attack.

Figure 3.3(b) shows that irrespective of ID length, the average number of detected colluders increases even when the binary error probabilities are higher than $\pi = 2 \times 10^{-1}$. It shows that random spreading indeed improves the detection rate of colluders.

3.1.4 Conclusion

To build a robust and secure multimedia fingerprinting scheme, we examined the robustness of random spreading to reduce the BER for extracted IDs, followed by an increased detection rate for colluders. Few researchers have used random spreading techniques with protection as a priority and then evaluated their robustness. We showed that random spreading improves BER and provides resilience against errors over BSC when combined with Tardos codes.

Our approach employs a pseudo-random sequence generator to generate random sequences to represent the symbols in the ID. The approach we used improved the BER, resulting in better detection of colluders. But, the random spreading performance decreases as we increase m. To further improve the robustness performance, we explore ECC with random spreading.

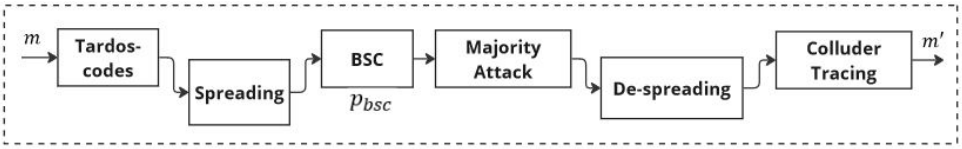
3.2 Error Correcting Codes with Watermarking

The problem is that embedding IDs into videos adds noise yielding binary errors on the ID, and consequently decreasing the performance of Tardos codes. As shown in Section 3.1.3, using random spreading can decrease the BER, resulting in an enhanced colluder detection rate. However, the gain provided by random spreading is decreased as the ID length is increased for a given watermark image size. To improve the performance of random spreading, we propose to use ECC.

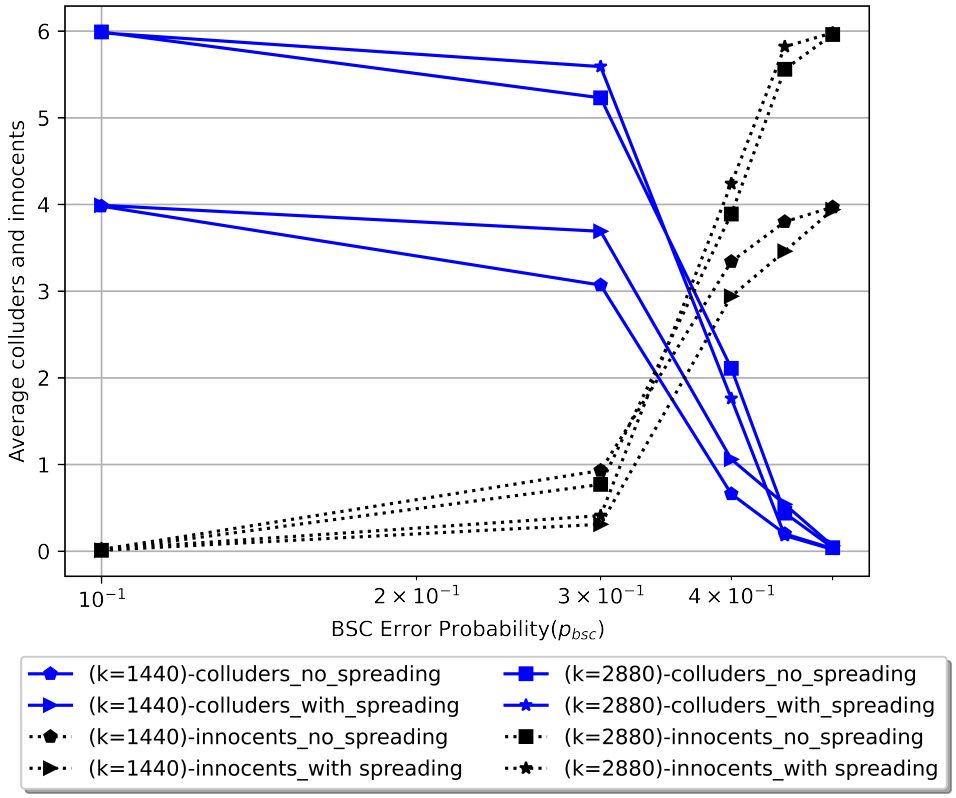
3.2.1 Overview of error correcting codes

An ECC is a solution to retrieve a message after having been transmitted over a noisy channel, by adding parity bits as shown in Figure. 3.4. There have been advancements in the implementation of ECC for digital communications to ensure that communications are secure.

An ECC produces coded data that include additional information in addition to the original data to enable message reconstruction in the event of noise. So, while noise may damage parts of



(a) Simulation model



(b) Simulation results

Figure 3.3 – Colluders tracing with proposed random spreading scheme for majority vote attack: (a) Simulation model (b) Average detected colluders with $m = [1440, 2880], n = 1000$ and $\varepsilon_1 = 10^{-3}$.

the data, the original message may be retrieved to some extent due to the additional information provided by the encoding of the original message.

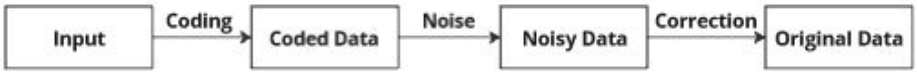


Figure 3.4 – The following chart shows the process in which error-correcting codes are implemented within data communication.

In the communication domain, ECCs are grouped into two types: block and convolutional codes. To secure the message, block codes divide the message into blocks that include redundant bits. The message in convolutional codes consists of data sequences of random length. The four commonly used ECC with watermarking are Hamming, convolutional, Reed-Solomon, and parity check codes. The Hamming codes [60] are block codes that can detect up to two-bit errors and rectify single-bit errors. The Hamming code uses a block parity mechanism. The data is divided into blocks, and parity is added to the block. Hamming code can correct single-bit errors and detect the presence of two-bit errors in a data block. For Hamming codes, the amount of parity data added to Hamming code is given by the formula $2^p \geq d + p + 1$, where p is the number of parity bits and d is the number of data bits.

Similarly, Reed-Solomon [61] codes are block codes that can correct burst errors in received data blocks. A Reed-Solomon code is specified as $RS(n, k)$ with s-bit symbols. This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are $n - k$ parity symbols of s bits each. The Low Density Parity Check (LDPC) codes [62] are also block codes defined by a parity-check matrix with a low density of 1s. They are appropriate for huge block sizes in extremely noisy channels.

The Bose–Chaudhuri–Hocquenghem (BCH) [63] codes form a large class of powerful cyclic error-correcting codes. For BCH codes, the codeword length N must have the form $2^m - 1$, where m is an integer. The message length k is restricted to particular values that depend on N. Lastly, the Hadamard codes [64] can also be referred to as Walsh codes or Hadamard-Walsh codes. The codes are significantly identical. Hadamard has been linked with developing the Hadamard matrix. Walsh later realized that each row of the Hadamard matrix might serve as a code sequence or code word. Turbo codes [65, 66, 67] are characterized by their powerful error-correcting capability while maintaining reasonable complexity and flexibility in terms of coding rates.

3.2.2 Overview of ECC with watermarking

The Hadamard codes have also been vastly utilized in the watermarking domain to improve watermarking performance. The Hadamard code generator block generates a Hadamard code from a Hadamard matrix, with the rows constituting an orthogonal set of codes. In these systems, the de-spreading process works effectively since the codes are completely de-correlated. The authors in [68, 69], employed the Hadamard codes with video watermarking. By merging the Hadamard code with the watermarking algorithm, they were able to strengthen the watermark's resistance against compression attacks.

However, the difficulty with Hadamard codes is their dependency on tight matrix sizes, which might be problematic when the watermark size is limited. Similarly, the authors in [70, 71] used Reed-Solomon codes with watermarking methods to give lower BER with resistance to compression and geometric attacks on the watermarks.

In [72], a watermarking scheme was presented employing three different ECC: BCH, turbo, and convolutional codes. After evaluating and comparing them, they determined that BCH had a higher error-correcting capability than turbo and convolutional. BCH codes provide exact control over the amount of symbol mistakes that may be corrected during code design. It is feasible to create binary BCH codes that can fix multiple-bit faults. However, in the context of watermarking, it does not consider other assaults such as compression. BCH codes provide exact control over the amount of symbol mistakes that may be corrected during code design. It is feasible to create binary BCH codes that can fix multiple-bit faults. However, in the context of watermarking, it does not consider other attacks such as compression.

This thesis investigates convolutional codes. Because Tardos code length is determined by the size of the watermark image, the convolutional code allows us to be more flexible in terms of length than most other codes. These codes are quick and have good results with limited integration costs. The Hamming code is quite small, Reed-Solomon needs too much parity, and the LPD codes are too strict in terms of code length. But, the convolutional codes [73], are favored because the message and code length can be adapted.

The authors from [74, 75, 76] showed that using convolutional codes with watermarking improves the PSNR but also that the scheme is not robust to compression, contrast enhancement, and collusion attacks. Convolutional codes have been surpassed by several other codes. However, convolutional codes are still widely utilized in watermarking. Especially in our case, where we have to deal with higher m of Tardos codes with pseudo-random sequences spreading. Convolutional codes are easier to employ when dealing with pseudo-random spreading while watermarking. In this thesis, we suggest using convolutional codes to improve the resistance and decrease overall BER, which eventually gives enhanced colluder tracing performance.

3.2.3 Convolutional codes and Viterbi decoding

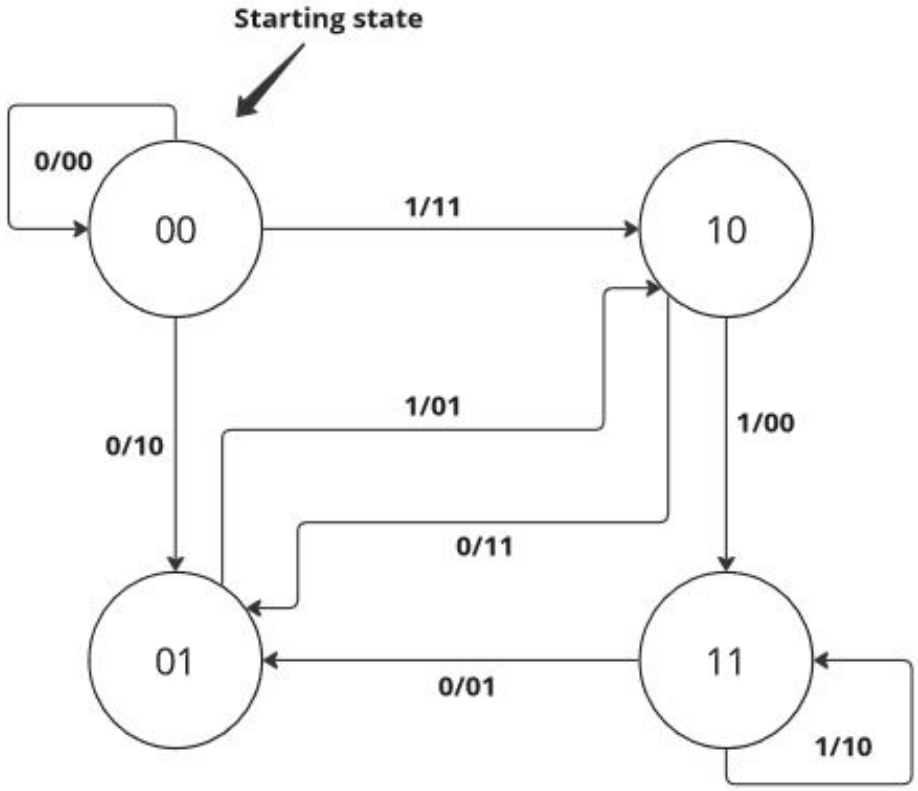
We will explain the convolutional encoder [73] in terms of a state diagram, which may be viewed as a series of states with clear transitions. This will help us comprehend convolutional codes and implement the encoding and decoding techniques. The state diagram for the convolutional encoder is shown in Figure. 3.5(a). The state diagram is identical for all codes with constraint depth S. The labels may change depending on the generator polynomial and the coefficients. For a given input message $x[n]$, each state will be labeled with $x[n-1]x[n-2]....x[n-S+1]$ and each connection is labeled with $x[n]/p_0p_1.....$ where p_i is the parity bit.

This state diagram idea is great for illustrating both how the transmitter operates and how the receiver decodes the message. The transmitter begins with the first state (labeled "STARTING STATE") and processes the message one bit at a time. The state machine switches to a new state based on the value of the input bit and generates the corresponding bits defined by transitions. The receiver is not directly aware of the transmitter's status changes. It only sees the received parity bit sequence, which might include corruptions. The decoder tries to determine the best possible sequence of transmitter states that might have produced the parity bit sequence.

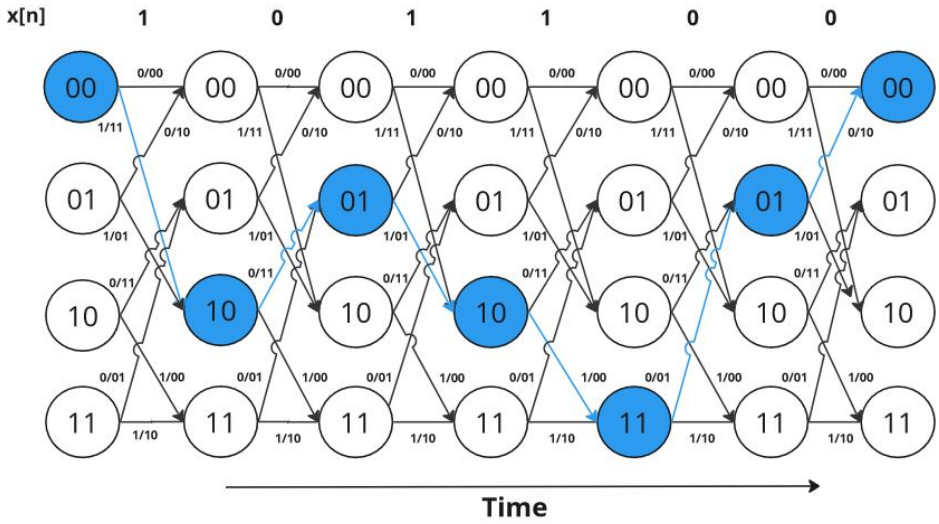
The standard method for convolutional code decoding uses a Viterbi trellis diagram. Trellis is an architecture derived from the state diagram. In the trellis, each column has a collection of states. Each state in a column is linked to two states in the following column, which corresponds to the two states in the state diagram. The topmost connection of each state in a column of the trellis indicates '0' transmissions, whereas the lower links indicate '1'. For example, given the message 101100, the transmitted bits are 111101000110 using the state diagram in Figure. 3.5(a). The links between the states are shown in Figure. 3.5(b), for the message 1001100.

The Viterbi algorithm uses metrics to decode the convolutional codes: the Branch Metrics (BM) and the Path Metrics (PM). The BM is used to measure the distance between the transmitted and received message. It will be defined over each connection in trellis. In hard decision decoding, the branch metric is the Hamming distance between the expected and received bits, while the path metric is associated with a trellis state and corresponds to the Hamming distance between the most likely path from the initial to the current state. We define "most likely" as the path with the shortest Hamming distance between the beginning and current states, calculated across all potential pathways between the two states. The decoded sequence is the result of trace-back. Trace-back is conducted from the end state to the starting point. The Add, Compare and Select (ACS) unit evaluates branch metrics before selecting the branch with the lowest Hamming/Euclidean distance and deleting the other branch as shown in Figure. 3.6.

The path with the shortest Hamming distance reduces the amount of bit errors. The Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, the Hamming distance is the number of bit positions in which the two



(a) State diagram for convolutional code $< 3, 1/2 >$.



(b) Viterbi decoding: the trellis representation of convolutional code$< 3, 1/2 >$.

Figure 3.5 – Understanding how the state diagram evolves and the trellis representation for visualizing the decoding of convolutional codes.

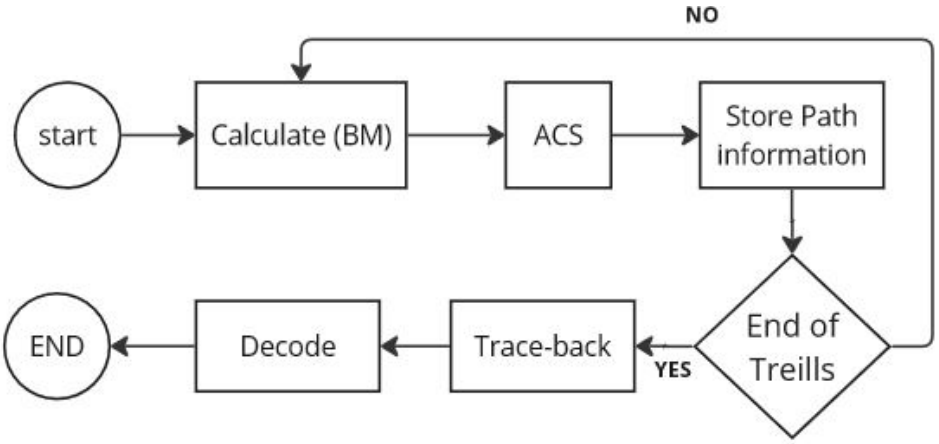


Figure 3.6 – The flowchart for the Viterbi algorithm.

bits are different.

Let u be a global state given in a suitable binary vector encoding, as a vector $(u_1, ..., u_k)$. Further on, let v be the error state we are searching for. One estimate for the number of transitions necessary to get from u to v is called the Hamming distance $h_d(u, v)$ that is defined in Equation 3.2:

$$h_d(u, v) := \sum_{i=1}^{k} (u_i - v_i) \tag{3.2}$$

The Viterbi algorithm's essential finding is that the receiver can calculate the PM for a (state, time) pair by combining the PM of previously computed states and BM. The Viterbi algorithms have two key steps: calculating the BM for the next batch of bits and computing the PM for the following column. First, add the BM to the PM for the old state and compare the sums for connections that arrive at the new state. Finally, choose the route with the least value, breaking connections randomly. This approach leads to the lower BER. So, we will adapt this representation of convolutional with random spreading.

3.2.4 Soft decoding and hard decoding

The function of Viterbi is identical for both hard and soft choice decoders, with the sole variation being that hard decision computes the Hamming distance and soft decision computes the Euclidean distance as the branch metric. Hard decision decoding converts incoming signals by comparing them to a threshold before sending them to the decoder. Consequently, we lose data. For example depending on modulation, if the signal was 0.5001, the level of faith in the digitization would be far lower than if it was 0.9999. However, both are classified as "1," and the decoder handles them equally, even though 0.9999 is far more likely to be a "1" than the other number.

In soft decision decoding, the decoder uses the actual received values (like 0.29) rather than just binary digits (0 or 1) to make more informed decisions about the transmitted bits. This approach captures the nuances of the signal and allows the decoder to take into account the level of confidence in the received values. The square difference between the received and predicted bit is a useful soft decision metric for Additive White Gaussian Noise (AWGN) channel. If the convolutional code generates b bits and associated analog samples ($v = v_1, v_2, ..., v_b$), a soft decision branch metric may be constructed as shown in Equation 3.3. where $u = u_1, u_2, ..., u_b$ are the expected b bits (each a 0 or 1).

$$BM_{(soft)}[u, v] := \sum_{i=1}^{b} (u_i - v_i)^2 \tag{3.3}$$

3.2.5 Convolutional codes with watermarking

The authors in [77] demonstrated that using convolutional codes with DWT watermarking provided enhanced resistance to multimedia compression, but without addressing other crucial attacks, such as collusion, geometric distortions, and cropping. In [78], the authors illustrate the robustness of a watermarking scheme for images using convolutional code embedding, and considering all standard multimedia attacks. However, collusion attacks are not addressed. Also, watermarking in this study is non-blind, meaning that the original image is required. But in our thesis, our focus is on blind watermarking: the original content will not be available on the receiver side.

In [79], ECC was used with random spreading watermarking. They used BCH, Reed-Solomon, and convolutional codes with Viterbi decoding and determined that utilizing convolutional codes with spreading enhances the PSNR and is also very resilient to compression. Similarly, the convolutional encoding and random spreading are also combined in [80] to lower the bit error rate brought on by interference from the host signal. However, in the simulations, no attacks of any kind were taken into account.

In this thesis, we also propose to use convolutional encoding either concatenated with random spreading or jointly, as proposed in [80], to improve colluder tracing in collusion attacks. A convolutional code is specified by its coding rate r_{cc} and to decode a convolutional code, we used the Viterbi algorithm, which finds efficiently the shortest path on the trellis diagram. Two spreading schemes using convolutional encoding are proposed hereafter: the concatenated scheme and the joint scheme.

3.3 Proposed encoding schemes with random spreading

For the concatenated scheme, the convolutional encoder is utilized to encode the m bits of the ID with rate r_{cc}. The classical convolutional encoder is utilized to generate the parity bits and

then pseudo-random sequences to represent that parity bits. The trellis shown in Figure 3.7(a) illustrates the outputs of the convolutional encoder of rate $r_{cc} = 1/2$ with $S = 2$ for each possible transition between 2 states. The output of the encoder is then spread using pseudo-random sequences of rate $\frac{\alpha}{r_{cc}}$ as described in section 3.1.2. In the joint scheme, the m bits of the ID are encoded and spread simultaneously utilizing joint convolutional encoding and spreading with rate α, as proposed by [80]. In other words, the parity bit generated by convolutional code(classical approach) is replaced by a pseudo-random sequence s_r. In this scheme, the outputs are given pseudo-random sequences s_{r_i} with $i \in [1, \ldots, 2 \times 2^N]$ of rate $\frac{1}{\alpha}$, as illustrated in Figure 3.7(b).

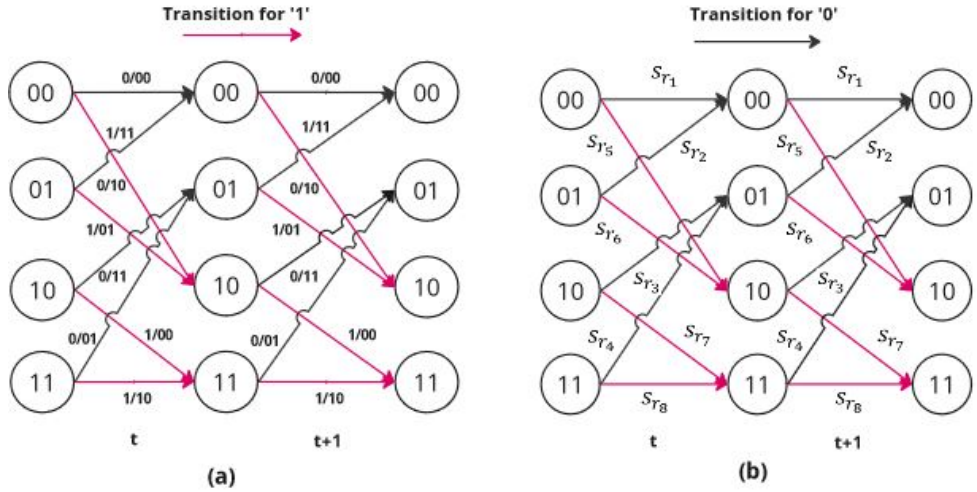


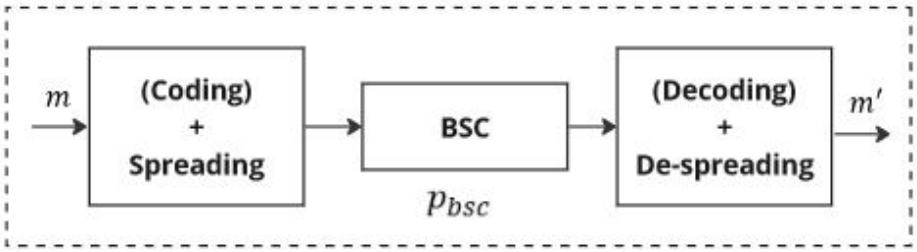
Figure 3.7 – Trellis diagram for 4 states: (a) concatenated scheme with rate $r_{cc} = \frac{1}{2}$ (b) joint scheme with rate $\frac{1}{\alpha}$.

3.3.1 Encoding schemes setup

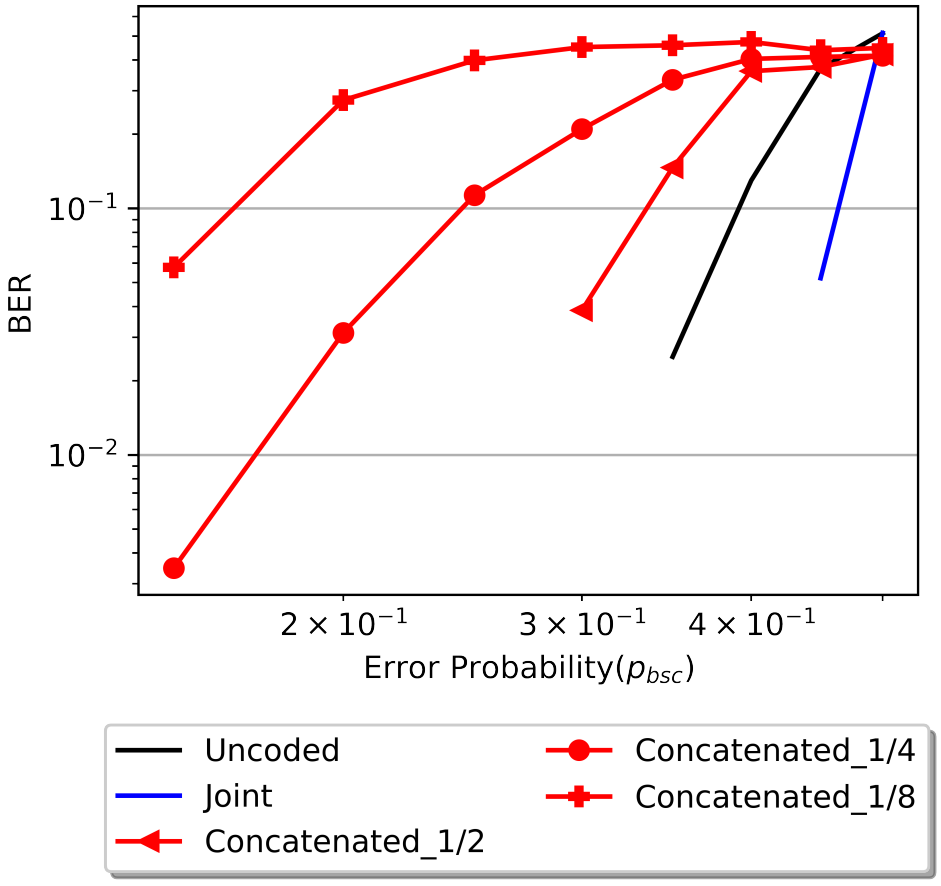
We evaluated the performance of the spreading using a BSC with error probability p_{bsc}, as shown in Figure 3.8(a). For performance evaluation, the concatenated and joint schemes are compared with a random spreading scheme denoted "uncoded" since no convolutional code is used.

For the concatenated scheme, we compare three different numbers of shift register: $S = 3, 5$ and 9, respectively, associated with 3 different rates $r_{cc} = \frac{1}{2}, \frac{1}{4}$ and $\frac{1}{8}$ as suggested in the literature [79]. The random spreading rates after encoding are thus, respectively, $2\alpha, 4\alpha$, and 8α. Simulations have been performed for $\alpha = 1/157$ (m being set to 1440) and are illustrated in Figure 3.8(b). The joint scheme outperforms the two other schemes.

To determine a viable ID length with an acceptable BER, we investigated the trade-off between spreading rate and BER. We performed simulations considering different ID lengths $m = [256, 512, 1024, 1440, 2880]$. Considering the BER target of 2×10^{-1}, the selected possible



(a) Simulation model



(b) Simulation results

Figure 3.8 – BER for spreading schemes combined with convolutional codes and compared to a pure random spreading scheme (uncoded) BSC with error probabilities p_{bsc} for a spreading rate $alpha = 1/157$.

ID lengths are $m = 2880$ for the joint scheme and $m = 1440$ for the uncoded scheme, as shown in Figure 3.9. These trade-offs correspond to a spreading rate of $\alpha = [1/157, 4/315]$, respectively, for lengths $m = [1440, 2880]$.

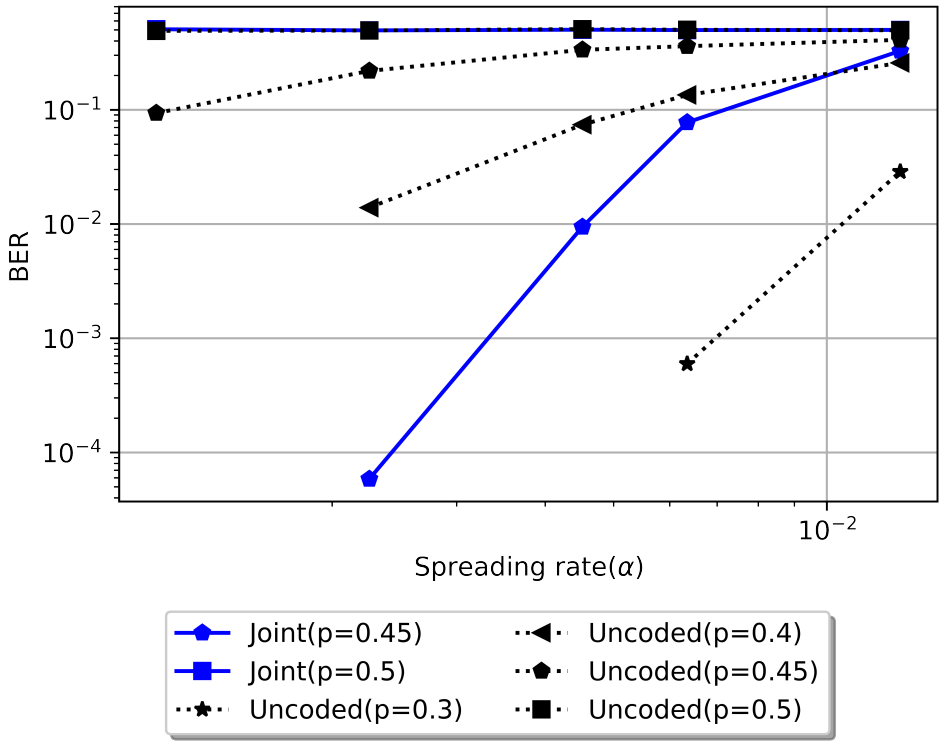
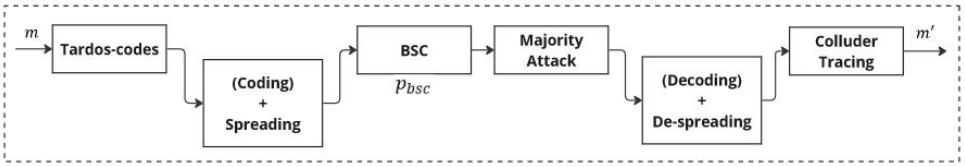


Figure 3.9 – Trade-off between BER and spreading rate for the joint and uncoded scheme with error probability $p_{bsc} \in [0.05, ...0.5]$ for BSC.

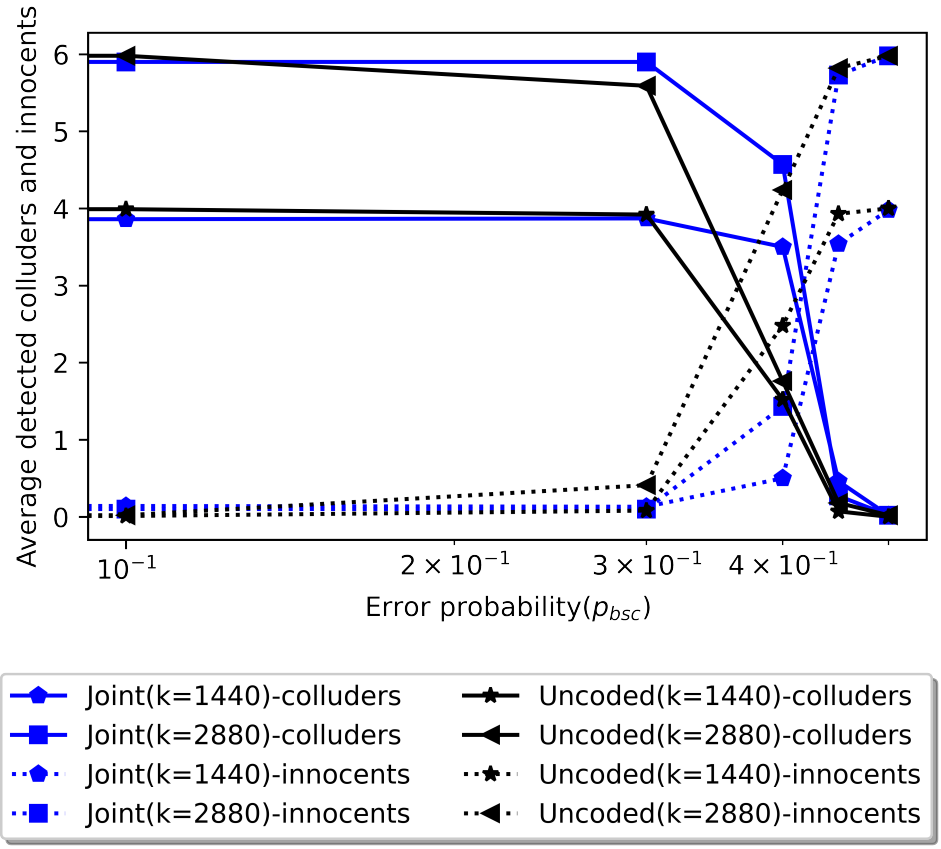
3.3.2 Experiments and results

Taking the two ID lengths of Tardos IDs with $n = 1000$ users and a probability of accusing innocent users set to $\varepsilon_1 = 10^{-3}$, we can trace a maximum of $c_0 = [4, 6]$ colluders. In this section, we address colluder tracing performance by considering majority attack models over BSC. To trace the colluders, we used the best generator-decoder combo from Chapter 2.

To analyze the impact of the spreading scheme for colluder tracing, the ID codeword m is spread and noised over a BSC with error probability π before a majority vote collusion attack is performed, as illustrated in Figure 3.10(a). The performance over the BSC is illustrated in Figure. 3.10(b): we observe that colluder tracing is much improved by the proposed joint scheme compared to the state-of-the-art uncoded spreading scheme, even when binary errors are higher than $\pi = 2 \times 10^{-1}$.



(a) Simulation model



(b) Simulation results

Figure 3.10 – (a): Simulation model for colluders tracing with $m = [1440, 2880], n = 1000$ and $\varepsilon_1 = 10^{-3}$ for the uncoded and the joint coding and spreading schemes; (b) Results for the simulation models of colluders; for majority vote attack over BSC

3.4 Summary and Conclusion

One task of this chapter is to seek a promising technique for constructing a robust and secure multimedia fingerprinting scheme. In this chapter, we demonstrated the significance and impacts of discrete watermarking on IDs. The discrete watermarking is one of the constraints in this thesis. The discrete watermarking generates low PSNR. Because of the low PSNR, the BER decreases for Tardos codes, making it harder to trace the colluders.

The errors might result in the loss of user IDs as well. The spreading schemes on Tardos codes improve the BER, followed by colluders detection rate. But, the spreading eventually has its limitation on the ID length. There is a trade-off between the performance of the spreading and the length of the ID. As the size of the ID increases the performance gain for spreading decreases.

So, this chapter is based on our publication [6], we proposed to combine ECC with random spreading to improve the colluder tracing performance. We analyzed the trade-off between the spreading rate and the bit error rate on the ID code. We then estimated the performance of the proposed joint convolutional code and random spreading, and compared this proposed scheme to the uncoded random spreading scheme. Simulation results were obtained first on a 360p image over a binary symmetric channel with a majority vote attack. These results showed that the proposed joint scheme outperforms the uncoded one in terms of colluder tracing.

In the next chapter, we will discuss how to employ random spreading with a watermarking approach, finally implementing collusion-resistant watermarking in real-time systems considering the results from Chapters 2 & 3.

Chapter 4

Real-time Watermarking with Collusion Resistant Codes

In this chapter, we study a full multimedia fingerprinting system. The system combines the collusion-resistant code generator-decoder from chapter 2 and random spreading with convolutional codes from chapter 3. In addition, we present an approach for watermarking. The real-time collusion attacks are presented and investigated utilizing FFMpeg with near approximation to a binary model. Finally, we suggested a real-time setup to stimulate realistic attacks on videos. The results will be discussed with one studied in the binary domain.

4.1 Proposed full multimedia fingerprinting scheme

To combat piracy, a multimedia fingerprinting system operates similarly to a communication chain, consisting of a transmitter, a channel, and a receiver. The transmitter is responsible for ID creation and embedding, while the receiver is in charge of ID decoding and tracing. Collusion and other watermarking attacks are modeled by transmission channels, as seen in Figure. 4.1.

Our proposed multimedia fingerprinting system has two main layers: fingerprinting layers with collusion-resistant codes and a watermarking layer using an embedding technique. So far, the designs for these two layers have often been generated individually. Since the early studies, the cryptography community has mostly investigated the fingerprinting layer based on collusion models. Those in the image or signal processing fields are mainly interested in the watermarking layer. As a result, it is critical to guarantee that a set of watermarking restrictions conforms to the assumptions established by fingerprinting designers.

In this thesis, we explored both of these layers; the fingerprinting layer in chapter 2 is given to establish the best generator-decoder combo, and the embedding layer in 3 is given the goal of increasing PSNR between embedded and extracted watermark. However, we have yet to explore the watermarking technique. In this chapter, we will propose our watermarking technique.

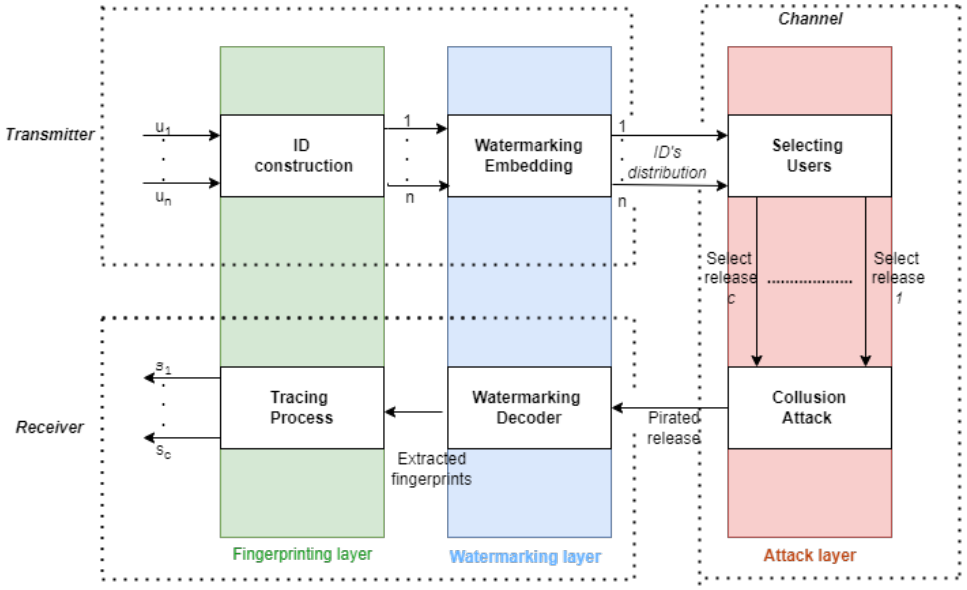


Figure 4.1 – The complete multimedia fingerprinting design scheme.

4.1.1 ID construction

The first step in a multimedia fingerprinting scheme is to generate user IDs. As described in detail in 2, an ID is a set of n different sequences $\{X_j\}_{j=1}^{n}$ with the size of m. These ID(collusion-resistant codes) have the property to observe a mixture of a different ID sequence. The mixed bet vector can be decoded to return a subset of those involved.

In our multimedia fingerprinting scheme, we considered and compared the different ID generators in chapter 2. We employ the Tardos family codes to generate the user IDs. We generated user IDs based on two algorithms 1 and 2. In chapter 2, we provided the best generator for the binary model. In this chapter, we will see if the best generator changes for real-time video embedding.

4.1.2 Overview on the Watermarking schemes

Digital watermarking is an ancient subject of study. However, there is still a lot of research being done in this area since there is no one general watermarking technique that is resistant to all forms of attacks. Watermarking can be done in the spatial or frequency domain. The spatial domain watermarking directly alters the host image pixel values to hide the watermark. This strategy is easier to handle and apply. However, it is less resilient than frequency domain methods, as shown in the brief overview of the literature in the following section 4.1.2.1.

In frequency-based watermarking, the host signal is first converted into the frequency domain using transforms such as DFT, DCT, DWT, and so on. After that, the watermark is embedded in

transformed frequency coefficients. It improves imperceptibility and robustness while increasing complexity. Any of the aforementioned embedding methods can insert the watermark into the images/videos. Similarly, during watermark extraction, the algorithm scans the watermarked content to locate and extract the watermark.

Many watermarking algorithms have been presented in the literature, with overviews provided in various research, each focusing on a certain component. The algorithms presented in [81] are based on the applications, technology, and system requirements of various media kinds, including digital photos, video, audio, and text. They showed that the frequency domain watermarking algorithms are more resistant to different attacks based on the applications. An overview of watermarking for digital images and videos is provided in [82]. In [83], video watermarking was proposed as a development of still picture watermarking. Both studies confirmed the long-term reliability of frequency-based watermarking technologies.

The literature [84, 85] also suggests utilizing frequency-based watermarking methods such as DWT to be resistant against a lot of common attacks such as noise, compression, and bit-rate modification. The recent studies by [86, 87, 88, 89] also summarize digital watermarking strategies based on restrictions such as attacks, visibility, perception, and reconstruction. In conclusion, the literature strongly recommends frequency-based approaches for secure and robust watermarking. Therefore, the study detailed in this chapter relies also on using the DWT frequency-based watermarking method.

4.1.2.1 Why choose "DWT" based watermarking?

The DWT is a frequency domain transform that works on wavelets with variable transform frequencies. The importance of DWT comes from its ability to break down signals at various sizes, which may be selected depending on the goal. In the video domain, many signals rely heavily on the low-frequency component, which includes the signal's characteristics, while the high-frequency component provides the signal's details or distinctions. It is a strong and valuable technique for signal analysis and processing that decomposes a 2D picture into four distinct sub-bands, namely LL (approximate component details), HL (horizontal component details), LH (vertical component details), and HH (diagonal component details) [26].

An image can be decomposed recursively by using DWT to get multi-scale wavelet decomposition, resulting in enhanced approximations and details. The lower coefficient in the low-resolution band LL of DWT represents the most information. The DWT offers multi-resolution and multi-layering abilities and is compatible with the human visual system. There are three main advantages of DWT. The first advantage is that the DWT watermarking is both multiresolution and multilevel. When the received image is not heavily damaged, cross-correlations with the whole image size may not be required, allowing for significant computational savings. The DWT level corresponds to the number of recursions. So, we don't need so much recursion here.

The second advantage is that human eyes are not sensitive to minor changes in an image's edges and textures, but they are extremely sensitive to high-frequency images. With the DWT, edges, and textures are often limited to high-frequency subands such as HH, LH, and HL. Large coefficients in these bands often suggest image edges. As a result, applying watermarks to these large coefficients is difficult for the human eye to detect.

The third advantage is that this method aligns with expected image/video compression standards. These recent studies [90, 91, 92, 93] reveal that this DWT based watermarking is highly resistant to image/video compressions, as well as other typical picture distortions such as additive noise, rescaling/stretching, and halftoning. The DWT technique has an advantage over other frequency-based methods for rescaling. As an example, the DCT coefficients for the rescaled image vary in two directions from those for the original image, which reduces correlation detection performance while decoding the watermark. Because the DWT is localized not just in the time as well as in the frequency domain [26, 94], the decline of correlation detection in the DWT domain is less significant than the one in the DCT domain. We will use the DWT based watermarking method in this thesis because of its benefits.

$$W(a,b) := \ < f(t), \partial(a,b) > \tag{4.1}$$

The DWT can be expressed as follows: where a is the scaling factor and b is the translation factor; $f(t)$ denotes an input signal; and $\partial(a,b)$ is the wavelet function. The wavelet transform returns $W(a,b)$, which is a function of a and b.

The choice of wavelet function $\partial(a,b)$: Despite conventional transform techniques that have set parameters, the primary challenge in DWT is to select a suitable wavelet function because the outputs of a signal change depending on the wavelets function utilized. The primary goal of wavelet transform is to find the similarity between the examined signal and the wavelet function employed [95], but it is difficult to implement in reality, even though wavelet function characteristics are well understood. As a result, it is difficult to select an acceptable wavelet function. Numerous studies have been conducted on this subject. In [96], it is suggested to pick a non-orthogonal wavelet function by considering the "width and shape" similarity of wavelets and coefficients. According to [97], the chosen wavelet function should have continuous and linear phases and be tuned to balance spatial and scaled resolutions.

The DWT employs many sorts of functions that serve as the mother wavelet for wavelet transformation. Wavelet families are separated into two types: orthogonal and biorthogonal. The orthogonal wavelets originate from a single orthogonal basis set, whereas the biorthogonal wavelets originate from several basis sets. Each basis set is weighted to create filters, either highpass or lowpass, which are the building blocks of Quadrature Mirror Filter (QMF) banks. These filters may be used to create wavelets, with varying weighted parameters affecting the performance of transformations in applications. As shown in [98], orthogonal wavelets can't

mix orthogonality with symmetry, resulting in the creation of biorthogonal wavelets. It was also revealed that biorthogonal wavelets outperform orthogonal wavelets in image compression applications because orthogonal wavelets lack the symmetric filters needed to resolve image edges. Thus, biorthogonal wavelets have a more symmetrical nature, whereas orthogonal wavelets are more periodic [99]. In literature, several wavelets linked with DWT include Haar, Daubechies, Symlet, and Coiflet.

The Haar wavelet [100] is one of the simplest and orthogonal types of wavelets. Haar wavelets are discontinuous and resemble step functions. Haar wavelet is known as the "mother of all wavelets" since it generates all wavelet functions needed for transformation, including translation and scaling functions. The Daubechies [101] family wavelets are referred to as dbN, where N represents the order and db denotes the wavelet name. Daubechies wavelets are bi-orthogonal, allowing for useful discrete wavelet analysis. The dbN family's main characteristics include easily supported wavelets with external phases and the highest number of vanishing moments for a given support width. Symlet wavelets [102] are also types of orthogonal wavelets. It can be generated by modifying the symmetry of Daubechies wavelets, and their characteristics are very similar to those of Daubechies wavelets. These are symmetrical wavelets, often known as symlets. Symlet wavelets are represented as $SymN$. Coiflet wavelets(CoifN) [103] are easily maintained wavelets having the greatest number of vanishing moments for any given support width. In some texts, it is referred to as 2N. The coiflet wavelets are more symmetric than the dbN's.

As a result, biorthogonal wavelets are best suited for image watermarking, but orthogonal wavelets might be useful in other applications such as multicarrier modulation. Since its publication, various publications [104, 105, 106], have used Daubechies wavelets for image watermarking due to their resistance to compression, detection, and denoising. Daubechies wavelets were enhanced in [107] using additional filters to improve smoothness and vanishing moments. It is known as CDF9/7 wavelets. According to [108], the popular CDF9/7 wavelet outperforms all the other wavelets for image watermarking. In this thesis, we will also rely on biorthogonal CDF9/7 wavelets. They are more symmetrical and close to orthogonality. This is a critical element in coding that ensures that the reconstruction error is extremely close to the quantization error in terms of mean squared error.

Decomposition level choice for DWT: In two-dimensional applications, for each level of decomposition, the DWT is carried out first in the vertical and then in the horizontal direction. Following the initial level of breakdown, there are four subbands: LL1, LH1, HL1, and HH1. The LL subband from the previous level serves as the input for each subsequent level of decomposition. To perform DWT on two levels, apply DWT to LL1. Apply DWT on LL2 for three levels to create four subbands (LL3, LH3, HH3, and HL3) as shown in Figure.4.2.

In DWT, low-level wavelet decomposition for image denoising has little influence on noise reduction and compression. On the other hand, a high-level wavelet decomposition may exclude

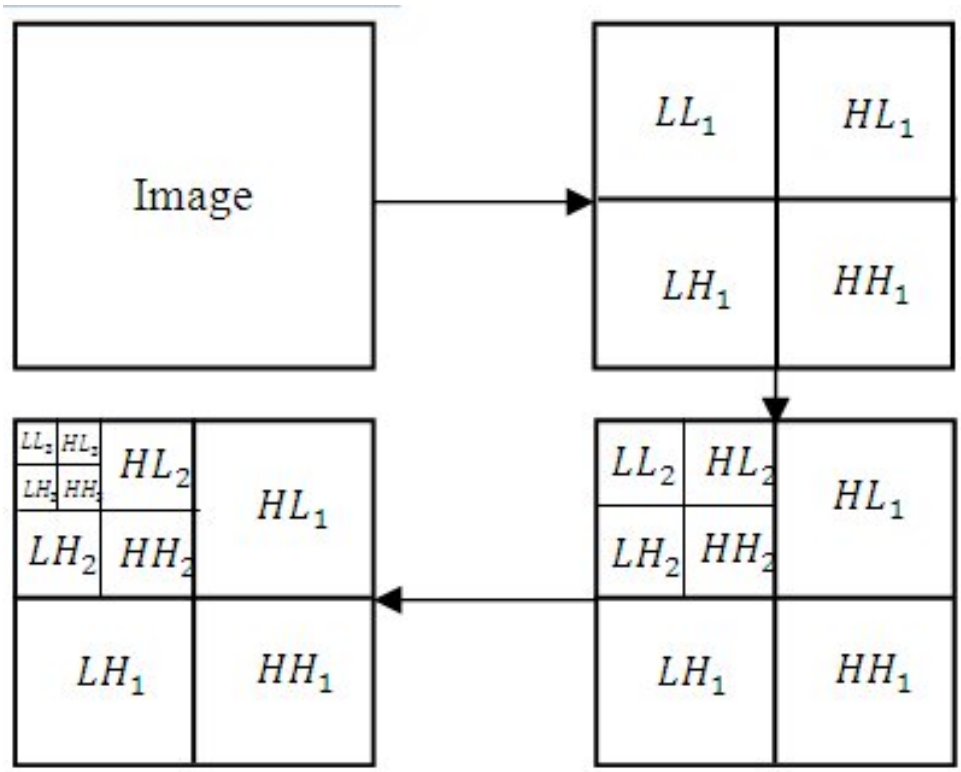


Figure 4.2 – Flow of DWT process (3-level decomposition: The HVS is more sensitive to the low-frequency coefficients and less sensitive to the high-frequency coefficients.

important information from the image. Therefore, it is necessary to find the optimum wavelet decomposition level. Recent studies [109, 110, 111] suggest using a 3-level decomposition of DWT to recreate the watermark without loss more effectively. So, in this thesis, we will also rely on the 3-level decomposition. The higher the level of decomposition of the image, the higher the level of blurness in the resulting image as shown in Figure 4.3.

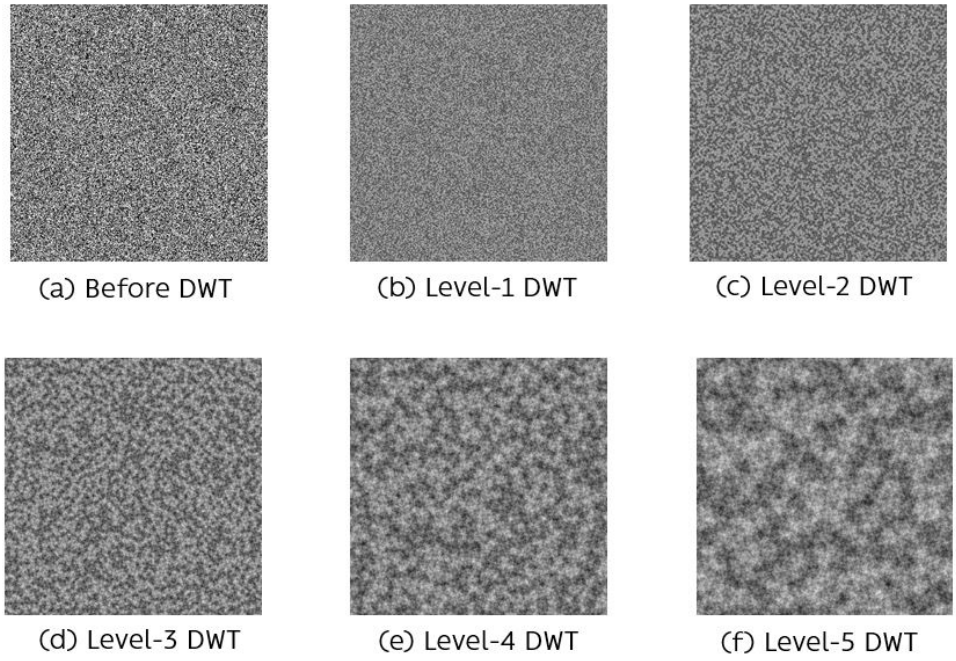


Figure 4.3 – An impact on image blurness for different levels decomposition with DWT.

4.1.2.2 Overview of watermarking with Tardos codes

Since their publication, Tardos codes have been employed in several types of embedding. Zafar et al. [112] used a spread spectrum robust watermarking method to identify H.264/AVC. They are, however, unsuitable for real-time applications due to considerable memory limitations and long code lengths. Tardos codes and zero-bit broken arrows watermarking method for photos were proposed in [113]. They've shown that this combination has ruled out fusion attacks.

In [114], a collusion-secure Tardos code-based fingerprinting scheme for 3D videos was presented, using a conventional LSB replacement for all the 2D video and the depth map components. However, here again, higher-length codes were employed without considering real-time attacks on the videos such as compression. The embedding of Tardos code with DWT watermarking was shown in [115], but the resilience has only been proven with an average attack on gray-scale images. Tardos codes were also used as QR codes to embed into images [116], and they propose numerous advantages such as providing a large quantity of information in a small

manner, reducing resilience, and solving concerns with computational costs and time of current tracing codes during the accusation process.

[117] gives an overview of DWT watermarking techniques using Tardos codes and including the ECC. They showed that ECC improves watermark detection and reconstruction. In [118], a DWT-based audio and video watermarking employing Tardos code is outlined, suggesting that the proposed combination delivers increased robustness against the security of multimedia content. In this thesis, we will use Tardos code as user IDs and embed them into a video using DWT while integrating ECC with random spreading.

4.1.3 Proposed DWT based collusion-resistant watermarking

In our proposed DWT based collusion-resistant video watermarking, a user ID is randomly spread while enduring convolutional codes. After spreading and coding, a watermark image is embedded in a video using DWT. Using alpha blending, we use FFMpeg to blend the watermark image into a video. Our suggested watermarking method consists of three basic stages. The details for each stage are provided below.

— Generate a watermark image: A watermark image will be created based on user ID. In addition, random spreading of user ID with the addition of convolutional codes.

— Transform: A DWT is performed based on CDF9/7 wavelets function. We utilized a 3-level decomposition of DWT. After DWT, results are quantized(0-255) to create the RGB watermark image.

— Embedding and extraction: An embedding of a watermark image into a video is based on FFmpeg. For the extraction we utilized IDWT with de-spreading and de-coding as shown in Figure 4.4.

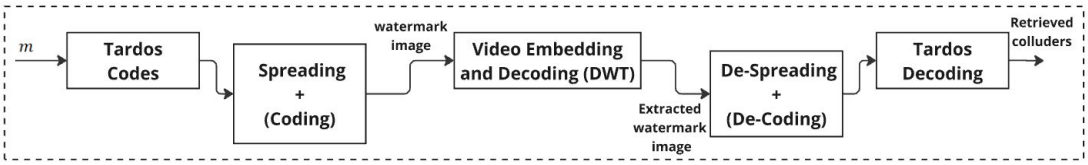


Figure 4.4 – The proposed video watermarking scheme using Tardos codes with random spreading with ECC.

4.1.3.1 Watermark image generation

We created a $360p = (360 \times 640)$ watermark image by utilizing user ID(Tardos codes). The random spreading of user ID can be done by creating a random dictionary $\mathcal{D}$ containing the random sequences d with length corresponding to the size of the watermark image as presented in Section 3.1.2. With random spreading, we also utilize the convolutional codes for more robustness

as given in Section 3.3. Then, we perform a DWT based on CDF9/7 wavelets function. To create the watermark image before blending it into the video, it is critical to specify the choice of wavelet function and level of decomposition.

Details on the choice of wavelets function and level of decompositions: we utilized the CDF9/7 wavelets function as shown in Figure. 4.5. For the level of decomposition, we use a 3-level decomposition of DWT as indicated in the literature (details in Section 4.1.2.1). Note that we omit the LL3 (lowest frequency) part of the 3-level DWT decomposition since it cannot be reproduced in a blind detection of the watermark image [119]. We thus come up with the final watermark image size $img_{len} = (360 \times 640) - LL3 = 226800$.

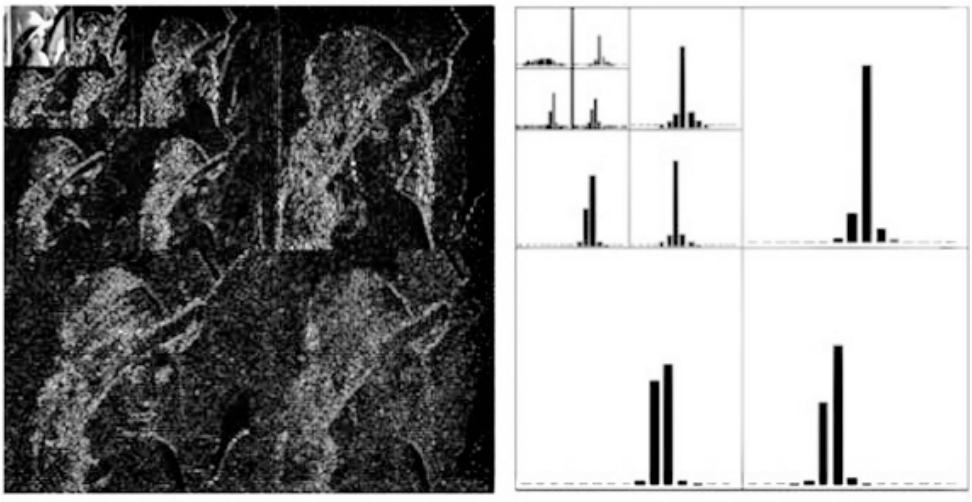


Figure 4.5 – Result of 3-level DWT decomposition of Lena using CDF 9/7 and its coefficient distribution in different sub-bands (Ref. [7], CC 2005, IEEE)

Lastly, it will be rescaled (according to the resolution of the source video) and quantized (0-255) to create a gray-scale image for embedding into the video.

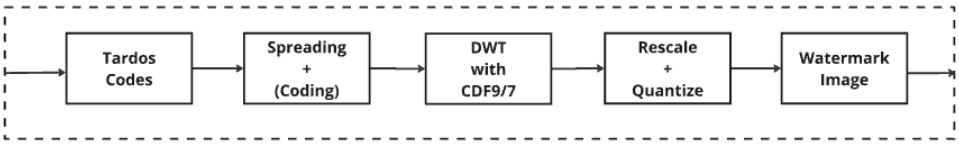


Figure 4.6 – Creation of watermark image based on user ID(Tardos codes).

4.1.3.2 Watermark embedding and extraction

We use the alpha blending technique to embed the watermark image into a video. In alpha blending, the watermarked video V_{wt} is obtained by alpha blending $V_{wt} = \mathcal{O}V_i + (1 - \mathcal{O})I_w$, where $\mathcal{O}$ is the opacity ranging between 0 and 1, I_w is the watermark image and V_i is the

source video. The $\mathcal{O}$ affects the perception or visibility of the watermark image. There is a balance between perception and robustness, with $\mathcal{O}$ values very close to 1. A robust video watermarking approach based on 3-level DWT was proposed in [120, 121]. These studies also employ alpha blending to produce discrete watermarking as well. Their results show that the opacity value (near 1) influences the embedding and extraction of watermarks. To create a discrete watermarking method, we will assume that the opacity $\mathcal{O}$ is very close to 1, as indicated in the literature. For embedding, we employ FFMpeg to blend the watermark image with the original video using the $\mathcal{O}$.

To extract the watermark image $\hat{I}_w$, collect frames from the watermarked video V_w and use 3-level DWT. This will return the random sequences used for random spreading to represent the user IDs. The recovered random sequences are compared to the original random sequences stored in $\mathcal{D}$ using Pearson Correlation, as illustrated in section 3.1.2.

4.1.4 Real-time collusion attacks

A collusion of c colluders can use their ID sequences X_j, where $j \in c$ creates a new pirate copy y using a different binary collusion attack model such as the majority and minority vote model as shown in previous Chapter 2. In real systems, the most prevalent collusion attacks are the darken and lighten blending filters of the FFMpeg as shown in Table 4.1.

Table 4.1 – FFmpeg blending filters for collusion attacks

FFmepg function	**Mathematical method**	**Attacks**
Lighten	$max(A, B)$	Majority
Darken	$min(A, B)$	Minority
Average	$\frac{A+B}{n}$	Average

The blending filter of FFMpeg takes two input video streams and outputs one stream, the first input is the "top" layer and the second input is the "bottom" layer. By default, the output terminates when the longest input terminates. FFMpeg can be used to execute a collusion attack on videos. Figure 4.7(a) shows the command lines for two source videos and it can be extended to multiple sources (e.g., four), Figure. 4.7(b).

4.1.5 ID decoding and accusation

A collusion attack (darken and lighten) on videos using FFMpeg results in a colluded copy of the video. This colluded video copy (or a pirated video) may be illegally redistributed. Analyzing pirated video reveals a colluding bit vector y, which is a combination of the user IDs involved in the collusion attack. So, the last step in the fingerprinting scheme is to analyze y to accuse the colluders(pirates). For accusation, we consider four different decoders in Chapter 2. We showed

```
ffmpeg -y -i {vidSrc1} -i {vidSrc2} {"-filter_complex [0:v][1:v]blend=all_mode={darken} -to 120.037".format(f=f)"} {vidDst}
ffmpeg -y -i {vidSrc1} -i {vidSrc2} {"-filter_complex [0:v][1:v]blend=all_mode={lighten} -to 120.037".format(f=f)"} {vidDst}
```

(a) Blending of two video streams

```
ffmpeg -y -i {vidSrc1} -i {vidSrc2} –i {vidSrc3} –i {vidSrc4}
{"-filter_complex [0:v][1:v]blend=all_mode={darken}[a];[2:v][3:v]blend=all_mode={darken}[b];[a][b]blend=all_mode={darken} -to 120.037"} {vidDst}
ffmpeg -y -i {vidSrc1} -i {vidSrc2} –i {vidSrc3} –i {vidSrc4}
{"-filter_complex [0:v][1:v]blend=all_mode={lighten}[a];[2:v][3:v]blend=all_mode={lighten}[b];[a][b]blend=all_mode={lighten} -to 120.037""} {vidDst}
```

(b) Blending of four video streams

Figure 4.7 – Command lines in FFMpeg to perform collusion attack (darken and lighten) with videos: (a) Collusion attack using two source videos, and (b) Collusion attacks for four source videos

that using *Laarhoven-Desoubeaux* is the best generator-decoder combo in the binary model. In this chapter, we look for the best decoder in real systems for video embedding. In the end, we present the best ID generator-decoder combo for video embedding.

4.2 Results and Experiments

The experimental study analyzes the performance of the proposed DWT based multimedia fingerprinting scheme. Firstly, we investigate the overall performance of the DWT based multimedia fingerprinting scheme with the different ID generator-decoder combo. In this section, we will show if the ID generator-decoder combo changes in real-time.

Following that, we investigate the overall performance of the proposed DWT-based multimedia fingerprinting scheme using random spreading with convolutional codes. In chapter 3, we proposed two different coding schemes(joint and concatenated schemes) of utilizing random spreading with convolutional codes. We showed that using a joint scheme reduces BER, which finally enhances accusation as compared to a concatenated scheme. Furthermore, the performance of uncoded (without ECC) is roughly comparable to the suggested concatenated method. In this section, we will examine the performance of the proposed joint scheme with an uncoded in real-time.

4.2.1 Experimental real-time setup

For the real-time setup, we will use the open-source Tear of Steel video [122] in 1080p resolution. A 360p watermark image is re-scaled to 1080p. Because when we use FFMpeg's blending filter for embedding, the size of the two sources should be identical. The experiments were carried out considering darken and lighten collusion attacks using FFMpeg, as presented in Section 4.1.4. The simulation was executed with a computer equipped with a Core-i5-CPU@1.6GHz (8 CPUs) and 16 GB RAM.

4.2.1.1 Best generator-decoder combo in real-time

We will compare ID generators and decoders for the proposed DWT-based fingerprinting scheme without any coding (ECC). The simulation setup for ID generation is the same as the binary model given in section 2.2.1. The averages detected colluders from the different decoders are given in Table 4.2 with Tardos ID generator and in Table 4.3 for *Laarhoven* ID generator.

In the case of the Tardos generator, we can detect several colluders using *Desoubeaux* and NNS decoders with average and darken attacks for $n = 100$ users. However, we can only discover one colluder for a lighten attack. In contrast, with *Laarhoven* generator, we can find multiple colluders for all attacks utilizing *Desoubeaux*'s, *Laarhoven*, and NNS decoders as shown in Table 4.3.

Table 4.2 – Real-time average detected colluders using different decoders with Tardos generator for $\mathcal{O} = 0.99$

Filter	n	**Tardos score**	***Laarhoven* score**	***Desoubeaux* score**	**NNS score**
Darken	30	3	5	5	6
	50	3	3	3	4
	100	1	1	2	2
Lighten	30	3	4	4	5
	50	3	4	3	4
	100	1	1	1	1
Average	30	5	6	6	5
	50	2	2	2	3
	100	1	1	2	2

Table 4.3 – Real-time average detected colluders using different decoders with *Laarhoven* generator for $\mathcal{O} = 0.99$

Filter	n	**Tardos score**	***Laarhoven* score**	***Desoubeaux* score**	**NNS score**
Darken	30	3	6	5	5
	50	4	4	4	4
	100	1	2	2	2
Lighten	30	5	5	7	5
	50	3	4	4	4
	100	1	1	2	2
Average	30	4	6	7	5
	50	5	6	6	5
	100	1	2	2	2

For the proposed DWT fingerprinting scheme in a real-time setup, we showed that the best

generator is *Laarhoven*, the same as the binary model. However, the decoder can be changed mainly because of decoding time and real setup. For real setup, estimating the collusion attack (θ_b) is not viable, which is the key parameter of the *Desoubeaux* decoder. As a result, we are unable to predict the real collusion attack. We endorse the NNS decoder for real-time setup since it has an equivalent accusation rate to *Desoubeaux*. Finally, we showed that the *Laarhoven* generator and NNS decoder are the best combination for real-world setups.

4.2.1.2 Joint scheme in real-time

Considering the findings of the previous section with the best generator-decoder combo for real setup without any coding, now we evaluate the performance of the proposed DWT-based fingerprinting system utilizing the joint coding scheme as shown in Figure 4.8. The *Laarhoven* generator, generates two IDs with parameter $m = [1440, 2880]$ and $n = 1000$. These IDs are converted into a watermark image after spreading with a joint coding scheme. This watermark image is embedded into the video with FFMpeg for the opacity ranging from 0.90 to 0.99.

The simulation results are depicted in the following of Figures 4.9, 4.10, and 4.11 corresponding to *darken*, *lighten*, and *average* collusion attacks. As the opacity approaches 1 (reducing the PSNR), the performance of the two schemes becomes equivalent for both ID lengths. However, with lower opacities (higher PSNR), the joint scheme continues to outperform the uncoded scheme for any type of attack.

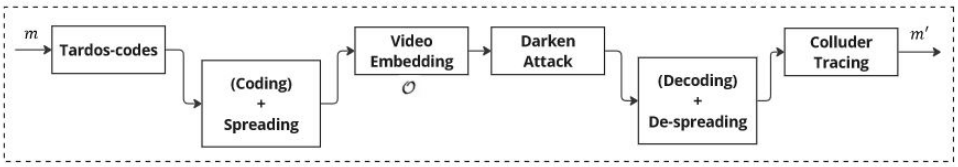


Figure 4.8 – Simulation model:

4.2.2 Conclusion

In this chapter, we proposed a complete DWT based multimedia fingerprinting scheme. This proposed multimedia fingerprinting system has two main layers: fingerprinting layers with collusion-resistant codes and a watermarking layer using an embedding technique.

In the fingerprinting layer, we suggested and compared several generators/decoders in chapter 2 in a binary model. We showed that the best generator-decoder combo is the *Laarhoven* generator with the *Desoubeaux* decoder. This combo results in a higher accusation rate but requires more decoding time. In this chapter, we examined the same generators/decoders to see if this combo changes in real systems (with video embedding).

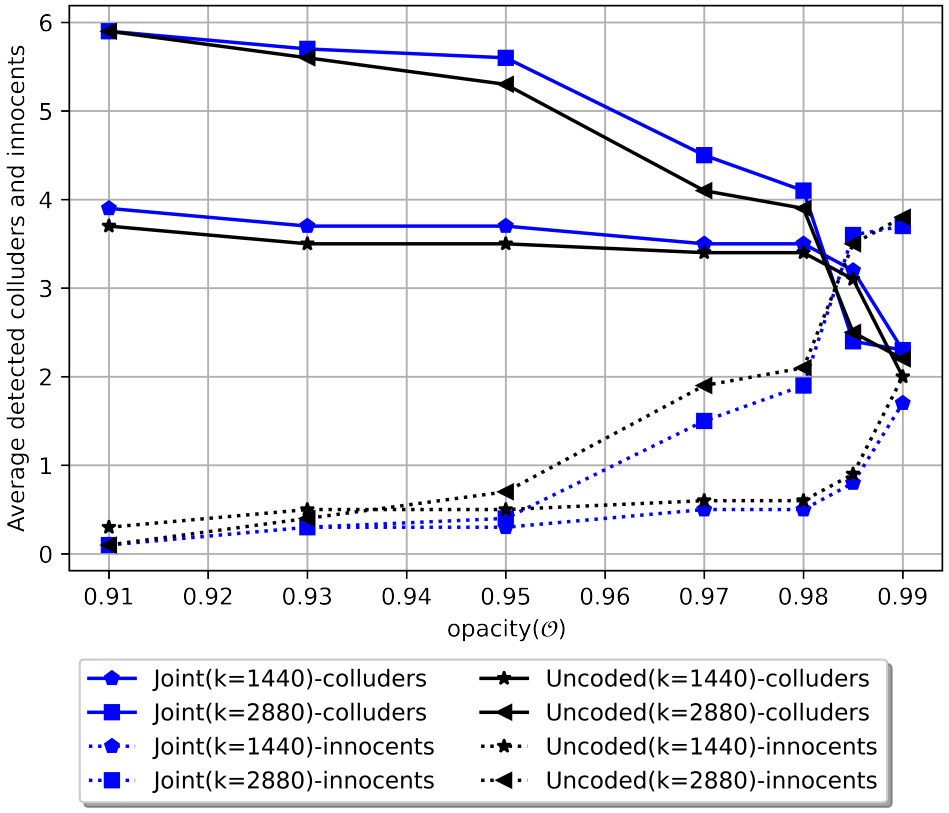


Figure 4.9 – The colluders for the uncoded and the joint coding scheme over *darken attack* on video with FFmpeg.

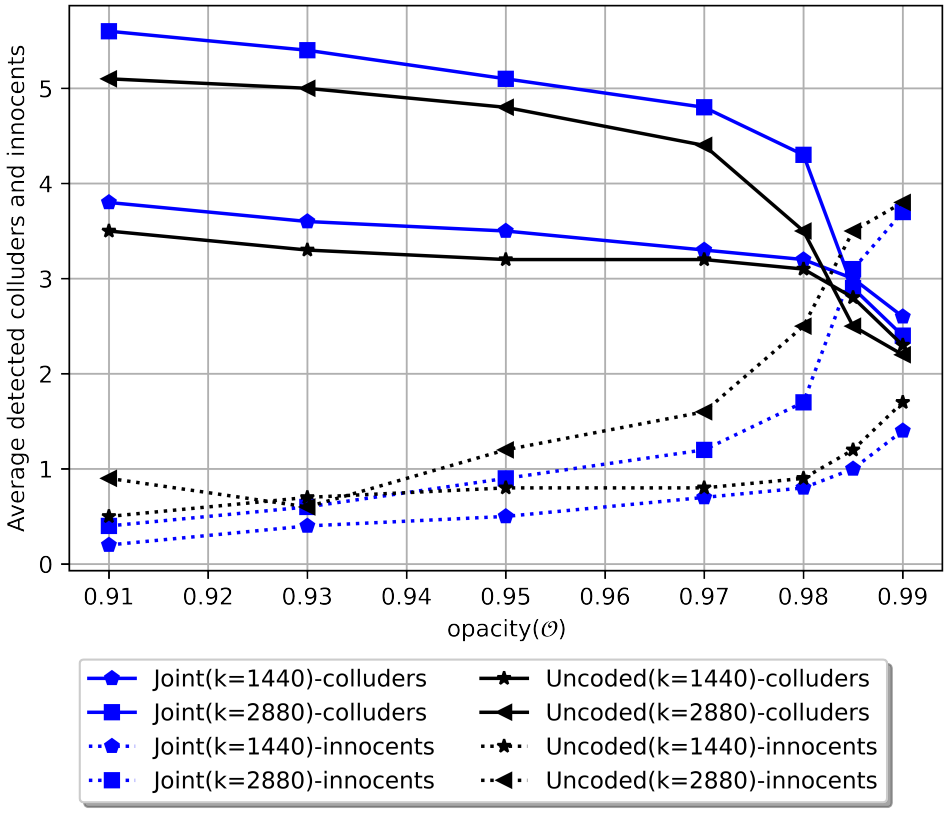


Figure 4.10 – The colluders for the uncoded and the joint coding scheme over *lighten attack* on video with FFmpeg.

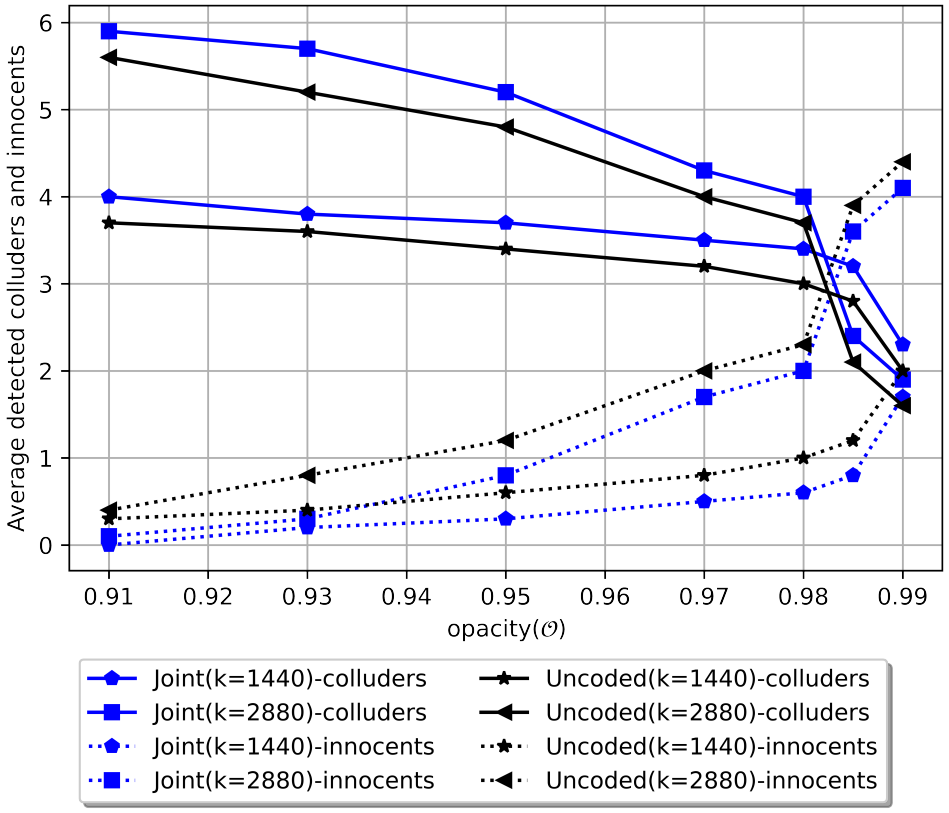


Figure 4.11 – The colluders for the uncoded and the joint coding scheme over *average attack* on video with FFmpeg.

In the watermarking layer, we use the DWT based watermarking approach combined with a joint coding scheme (random spreading with convolutional codes) from chapter 3. In addition, we provided a detailed overview of watermarking techniques and why we chose DWT. Also, the real-time collusion attacks are presented and investigated utilizing FFMpeg with near approximation to a binary model.

Finally, we suggested a real-world setup and performed experiments with real videos. we examined the performance of both fingerprinting and watermarking layers in our proposed DWT based fingerprinting scheme.

For the fingerprinting layer in a real-world setup, we examined the same generators/decoders and showed that the best generator-decoder combination changes from the one selected for binary models. We showed that the *Laarhoven* generator with NNS decoder is the best combo. Both the binary and real-world generator-decoder combinations provide a similar accusation rate. The main difference, however, is the time required for decoding. This is critical in real systems because we need to discover the colluder quickly. We recommend using the NNS decoder over *Desoubeaux*, because the *Desoubeaux* decoder is more complex and challenging to implement.

For the watermarking layer in a real-world setup, we pick the most suitable generator-decoder combination (*Laarhoven* and NNS). Then, after comparing the joint coding scheme's performance to that of the uncoded one, we concluded that the joint coding scheme outperforms alternative solutions in real systems, as with the binary model.

Conclusions and Future Perspective

In this thesis, we have studied various aspects of multimedia fingerprinting for traitor tracing. This thesis started from the study of the background and the state of the art of the multimedia fingerprinting field. We provide a generic framework for multimedia fingerprinting schemes and explore the many flaws they encounter. Furthermore, we examine the two primary components in the design of multimedia fingerprinting schemes: fingerprint generation and watermarking. Our purpose in this thesis is to take into account the design of the anti-collusion code as well as the design of the watermarking system that will embed the code in content. This provides us with an overall view of the fingerprint embedding's strength and the efficiency of the accusing procedure. We especially focused on the significance and impacts of discrete watermarking with user IDs, which is one of the constraints in this thesis. Such constraints lead to low PSNR and, so to the significant BER, which impacts Tardos capabilities.

In chapter 1, we discussed the necessity for internet content protection. Then we defined video piracy and discussed several approaches to combat it One approach is to use a multimedia fingerprinting scheme, consisting of two key components: Tardos codes at first and embedding in the video content. We also discussed several transformations that may affect the watermarking robustness. Additionally, we have studied several collusion attack models that illustrate several acts of piracy. Then, we identified various thesis restrictions in contrast to the project's goal. The restrictions include discrete watermarking, blind detection, and resilience.

In chapter 2, we studied different collusion pairs of code generator-decoders based on Tardos work. We compared Tardos and Laarhoven as generators with Tardos, Laarhoven, Desoubeaux, and NNS as decoders. To select the best generator-decoder, we compared performance in terms of detection rate and complexity in simple binary (theoretical mode) with a theoretical attack model. We showed that the best generator-decoder combo is the Laarhoven generator with the Desoubeaux decoder. This combo results in a higher accusation rate but requires more decoding time, and Desoubeaux needs to know the attack model, which may not be clear on pirated content. To further reduce the complexity, we established a dynamic setup for hierarchical code construction that considerably improves detection while decreasing decoding time as shown in Figure. 2.9.

In chapter 3, we described the importance and impact of discrete watermarking with user IDs. While decoding, Tardos codes with higher BER, making tracing colluders more challenging. The

errors may potentially cause the loss of user code and accusing an innocent. Spreading methods on Tardos codes improve the PSNR, followed by collusion detection rate. However, the spreading ultimately limits the Tardos code length.

There is a trade-off between spreading performance and ID length. As the size of the ID increases, the performance of spreading decreases. In addition to spreading, we propose to use ECC, to reduce BER for Tardos codes. We compared the performance of the proposed joint coding scheme (convolutional code with random spreading) with the uncoded scheme (no ECC). In terms of colluder tracing, the proposed joint technique beats the uncoded one for both the majority and minority vote attack models as shown in Figure. 3.10.

In the chapter 4, we proposed a complete DWT based multimedia fingerprinting scheme. This scheme is composed of two layers, a coding layer and a watermarking layer. In the coding layer, we examined generators/decoders from theory (see chapter 2), but, showed different winner combinations. Indeed, Chapter 2 favors the Laarhoven generator with NNS decoder as the best combo. Both the binary and real-time generator/decoder combinations provide a similar accusation rate. The main difference, however, is the time required for decoding. This is critical in real-time because we need to discover the colluder quickly.

We propose utilizing the NNS decoder over the Desoubeaux decoder because the latter is more complex and difficult to implement in real-time. Furthermore, Desoubeaux needs the attack model to decode, which will not be available in real-time. In the watermarking layer, we proposed a DWT based watermarking approach combined with a joint coding scheme(random spreading with convolutional codes). We used the most suitable generator-decoder combination (Laarhoven and NNS) and compared the joint coding scheme's performance to the uncoded one. We concluded that, as with the binary model, the joint coding scheme outperforms alternative coding schemes in real-time.

Despite the work carried out in this thesis, there are still several interesting research directions that are worth further exploring.

1. In this thesis, we tried to design a DWT based multimedia fingerprinting that can resist collusion attacks. It can resist certain robustness attacks, security attacks, and fusion attacks as found in the literature (details in section 4.1.2.1). However, certain other attacks can affect the colluder accusation, such as the de-synchronization attack [123], which can generate unintentional geometric distortion of the host video. The pirate trades visually identical frames to mislead the ID identification process. While DWT offers advantages in terms of frequency localization and multi-resolution analysis, it remains vulnerable to synchronization and geometric attacks, making it challenging to ensure the integrity and robustness of watermarked content under these conditions. We should also put our proposed multimedia fingerprinting system through comparable attacks and provide some protection.

2. When working with error-correcting codes, it's important to identify the type of errors you're dealing with—such as burst errors, random errors, or specific patterns of corruption. Understanding the nature of the errors can guide you toward the most effective coding scheme. For instance, if you're dealing with burst errors, codes like Reed-Solomon might be more suitable, while Hamming codes might better handle random errors. Video watermarking can encounter several types of errors that affect the visibility and integrity of the watermark. Here are some common error types: compression artifacts, transmission errors, cropping and resizing, Gaussian noise, and Format conversation. We didn't explore several advanced ECC such as Turbo codes [124]. They might be more efficient and more robust ECC with video watermarking. We think this study would be an interesting direction if we want to get more robustness.
3. We have not studied rare fingerprint schemes such as asymmetric fingerprinting [125] and anonymous fingerprinting [126]. These schemes often involve complex cryptographic protocols that can be difficult to implement correctly. Ensuring security while maintaining usability can be a balancing act. We are unsure if the proposed watermarking scheme in this thesis will perform well for these rare schemes; it would be fascinating to test this in future work.

BIBLIOGRAPHY

[1] http://en.wikipedia.org/wiki/PPLive. [Online; accessed 09-September-2024].

[2] http://en.wikipedia.org/wiki/PPStream. [Online; accessed 09-September-2024].

[3] Parks Associates. Eu study: Online piracy rebounds, but not due to covid-19. *Parks Associates*, 2023.

[4] Naoyuki Akashi, Minoru Kuribayashi, and Masakatu Morii. Hierarchical construction of tardos code. In *2008 International Symposium on Information Theory and Its Applications*, pages 1–6, 2008.

[5] Abdul Rehman, Gaëtan Le Guelvouit, Jean Dion, Frédéric Guilloud, and Matthieu Arzel. Dwt collusion resistant video watermarking using tardos family codes. In *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, volume Five, pages 1–6, 2022.

[6] Abdul Rehman, Gaëtan Le Guelvouit, Jean Dion, Frédéric Guilloud, and Matthieu Arzel. Collusion resistant watermarking using convolutional encoding and random spreading. In *ICWMC 2024, The 20th International Conference on Wireless and Mobile Communications, IARIA, Athènes, Greece*, 2024.

[7] P. Kumsawat, K. Attakitmongcol, and Srikaew. A new approach for optimization in image watermarking by using genetic algorithms. *IEEE Transactions on Signal Processing*, 53(12):4707–4719, 2005.

[8] Miguel Godinho de Matos, Pedro Ferreira, and Michael D. Smith. The effect of subscription video-on-demand on piracy: Evidence from a household-level randomized experiment. *Management Science*, 64(12):5610–5630, 2018.

[9] Nicolas Dias Gomes, Pedro André Cerqueira, and Luís Alçada-Almeida. Determinants of worldwide software piracy losses. *Technological and Economic Development of Economy*, 24(1):48–66, 2018.

[10] Arthur S De Vany and W David Walls. Estimating the effects of movie piracy on box-office revenue. *Review of Industrial Organization*, 30:291–301, 2007.

[11] R.T. Watson and Erich Schwartzel. Hollywood Movies Flood Piracy Sites Hours After Release. *The Wall Street Journal*, August 2021.

[12] Ernesto Van der Sar. Streaming piracy market and ecosystem strategies. *TorrentFreak article*, 2023.

[13] S.R. Subramanya and B.K. Yi. Digital rights management. *IEEE Potentials*, 25(2):31–34, 2006.

[14] Rahul Telang. Does online piracy make computers insecure? evidence from panel data. *Evidence from Panel Data (March 12, 2018)*, 2018.

[15] Behrouz Zolfaghari and Pinaki Mitra. A survey on piracy protection techniques in digital cinema watermarking schemes. In *Recent Trends in Communication Networks*, page 11. IntechOpen, 2020.

[16] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3.4):313–336, 1996.

[17] Rongsheng Xie, Keshou Wu, Jiangbo Du, and Chunguang Li. Survey of public key digital watermarking systems. In *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing - Volume 02*, SNPD '07, page 439–443, USA, 2007. IEEE Computer Society.

[18] Takaaki Yamada, Hiroshi Yoshiura, I. Echizen, Kazuto Ogawa, Itsuro Murota, Go Ohtake, and Seiichi Gohshi. Watermarking applications for broadcast content copyright protection. *The Journal of the Institute of Image Information and Television Engineers*, 57:1155–1167, 09 2003.

[19] Mir Shahriar Emami, Khairuddin Omar, Shahnorbanun Sahran, and SNHS Abdullah. Spatial domain approaches for real-time ownership identification. *Journal of Advances in Information Technology*, 5(1):1–4, 2014.

[20] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 49–56, 1999.

[21] Jean-Francois Delaigle, Christophe De Vleeschouwer, and Benoit MM Macq. Digital watermarking. In *Optical Security and Counterfeit Deterrence Techniques*, volume 2659, pages 99–110. SPIE, 1996.

[22] S. Voloshynovskiy, S. Pereira, T. Pun, J.J. Eggers, and J.K. Su. Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE Communications Magazine*, 39(8):118–126, 2001.

[23] Chih-Chin Lai and Cheng-Chih Tsai. Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE Transactions on Instrumentation and Measurement*, 59(11):3060–3063, 2010.

[24] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun. A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1750–1753, 2007.

[25] Qiang Li, Chun Yuan, and Yu-Zhuo Zhong. Adaptive dwt-svd domain image watermarking using human visual model. In *The 9th International Conference on Advanced Communication Technology*, volume 3, pages 1947–1951, 2007.

[26] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[27] Junxiu Liu, Jiadong Huang, Yuling Luo, Lvchen Cao, Su Yang, Duqu Wei, and Ronglong Zhou. An optimized image watermarking method based on hd and svd in dwt domain. *IEEE Access*, 7:80849–80860, 2019.

[28] Abdul Rehman, Gaëtan Le Guelvouit, Jean Dion, Frédéric Guilloud, and Matthieu Arzel. Dwt collusion resistant video watermarking using tardos family codes. In *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, volume 5, pages 1–6, 2022.

[29] Gábor Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), may 2008.

[30] Mahbuba Begum and Mohammad Shorif Uddin. Digital image watermarking techniques: A review. *Information*, 11(2), 2020.

[31] P. Bassia, I. Pitas, and N. Nikolaidis. Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia*, 3(2):232–241, 2001.

[32] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

[33] Yacov Yacobi. Improved boneh-shaw content fingerprinting. *Lecture Notes in Computer Science*, 2020:378–391, 04 2001.

[34] Oded Blayer and Tamir Tassa. Improved versions of tardos' fingerprinting scheme. *Des. Codes Cryptography*, 48:79–103, 07 2008.

[35] M. Celik B. Škorić, S. Katzenbeisser. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. *An International Journal of Design Code and Cryptography*, 46(2):137–166, 2008.

[36] Manabu Hagiwara, Goichiro Hanaoka, and Hideki Imai. A short random fingerprinting code against a small number of pirates. In *Proceedings of the 16th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, AAECC'06, page 193–202, Berlin, Heidelberg, 2006. Springer-Verlag.

[37] K. Nuida M. Hagiwara H. Watanabe and H. Ima. An improvement of discrete Tardos fingerprinting codes. *An International Journal of Designs, Codes and Cryptography*, 52(3):339–362, 2007.

[38] Ehsan Amiri and Gábor Tardos. High rate fingerprinting codes and the fingerprinting capacity. In *ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[39] N. Prasanth Anthapadmanabhan, Alexander Barg, and Ilya Dumer. On the fingerprinting capacity under the marking assumption. *CoRR*, abs/cs/0612073, 2006.

[40] Teddy Furon and Luis Pérez-Freire. Worst case attacks against binary probabilistic traitor tracing codes. *CoRR*, abs/0903.3480, 2009.

[41] Thijs Laarhoven and Benne de Weger. Discrete distributions in the tardos scheme, revisited. *Association for Computing Machinery*, page 13–18, 2013.

[42] Tatsuya Yasui, Minoru Kuribayashi, Nobuo Funabiki, and Isao Echizen. Near-optimal detection for binary tardos code by estimating collusion strategy. *IEEE Transactions on Information Forensics and Security*, 15:2069–2080, 2020.

[43] M. Desoubeaux C. Herzet W. Puech and G. Le Guelvouit. Enhanced Blind Decoding of Tardos Codes with New Map-Based Functions. *IEEE 15th International Workshop on Multimedia Signal Processing (MMSP),*, October 2013. Pula, Italie,.

[44] Laarhoven. Capacities and Capacity-Achieving Decoders for Various Fingerprinting Games. *In ACM Workshop on Information Hiding and Multimedia Security*, 2014.

[45] Thijs Laarhoven. Nearest neighbor decoding for tardos fingerprinting codes. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, page 182–187, New York, NY, USA, 2019. Association for Computing Machinery.

[46] Faten Chaabane, Maha Charfeddine, and Chokri Ben Amar. A multimedia tracing traitors scheme using multi-level hierarchical structure for tardos fingerprint based audio watermarking. In *2014 International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pages 289–296, 2014.

[47] A. Ben Hamida, M. Koubaa, C. Ben Amar, and H. Nicolas. Hierarchical traceability of multimedia documents. In *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 108–113, 2011.

[48] Faten Chaabane, Maha Charfeddine, and Chokri Ben Amar. Clustering impact on group-based traitor tracing schemes. In *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 440–445, 2015.

[49] Faten Chaabane, Maha Charfeddine, and Chokri Ben Amar. An enhanced hierarchical traitor tracing scheme based on clustering algorithms. In Francisco Javier Martinez de Pisón, Rubén Urraca, Héctor Quintián, and Emilio Corchado, editors, *Hybrid Artificial Intelligent Systems*, pages 379–390. Springer International Publishing, 2017.

[50] Joonho Choi, Abu Ahmed S. Reaz, and Biswanath Mukherjee. A survey of user behavior in vod service and bandwidth-saving multicast streaming schemes. *IEEE Communications Surveys and Tutorials*, 14:156–169, 2012.

[51] Joost Poort, João Pedro Quintais, Martin A. van der Ende, Anastasia Yagafarova, and Mathijs Hageraats. Global online piracy study. *Amsterdam Law School Research*, pages 2018–21, 2018.

[52] Tjiptono, Fandy, Arli, Denni, and Viviea. Gender and digital piracy: Examining determinants of attitude toward digital piracy among youths in an emerging market. *International Journal of Consumer Studies*, 40, 08 2015.

[53] Shashi Kiran Chilappagari, Sundararajan Sankaranarayanan, and Bane Vasic. Error floors of ldpc codes on the binary symmetric channel. In *2006 IEEE International Conference on Communications*, volume 3, pages 1089–1094, 2006.

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

[55] H.S. Malvar and D.A.F. Florencio. Improved spread spectrum: a new modulation technique for robust watermarking. *IEEE Transactions on Signal Processing*, 51(4):898–905, 2003.

[56] P. Moulin and A. Ivanovic. The zero-rate spread-spectrum watermarking game. *IEEE Transactions on Signal Processing*, 51(4):1098–1117, 2003.

[57] Zhu Huaihong, Tian Li, Ma Jingwen, Wei Yongzhuang, Sun Tie, and Zhu Huaihong. A dual pseudo random sequence digital watermarking algorithm based on dct. In *2011 International Conference on E-Business and E-Government (ICEE)*, pages 1–4, 2011.

[58] Z. Liu and A. Inoue. Audio watermarking techniques using sinusoidal patterns based on pseudorandom sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8):801–812, 2003.

[59] Suresh. C. Kuri and G.H. Kulkarni. Robust wavelet-based color image watermarking using pseudo random numbers. In *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pages 307–310, 2015.

[60] William Rurik and Arya Mazumdar. Hamming codes as error-reducing codes. In *2016 IEEE Information Theory Workshop (ITW)*, pages 404–408, 2016.

[61] Stephen B Wicker and Vijay K Bhargava. *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[62] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[63] R. C. Bose and Dwijendra K. Ray-Chaudhuri. On A class of error correcting binary group codes. *Inf. Control.*, 3(1):68–79, 1960.

[64] W.K. Pratt, J. Kane, and H.C. Andrews. Hadamard transform image coding. *Proceedings of the IEEE*, 57(1):58–68, 1969.

[65] R. Wadekar and L. Fagoonee. Beyond third generation (b3g) mobile communication: challenges, broadband access and europe. In *Proceedings of the 3rd International Conference on Mobile Technology, Applications & Systems*, Mobility '06, page 5–es, New York, NY, USA, 2006. Association for Computing Machinery.

[66] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, 1996.

[67] Corina Nafornita, Alexandru Isar, and Maria Kovaci. Increasing watermarking robustness using turbo codes. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 113–118, 2009.

[68] Michael Windisch, Jakob Wassermann, Monica Leba, and Olimpiu Stoicuta. Hadamard error-correcting codes and their application in digital watermarking. *Sensors*, 24(10), 2024.

[69] Dziech, Andrzej and Wassermann, Jakob. Application of enhanced hadamard error correcting code in video-watermarking and his comparison to reed-solomon code. *MATEC Web Conf.*, 125:05007, 2017.

[70] J. S. Y. Jeedella, H. Al Ahmad, and O. Al Shehhi. Watermarking mobile phone colour images with reed solomon error correction code. In *2012 16th IEEE Mediterranean Electrotechnical Conference*, pages 375–378, 2012.

[71] Ntitiseng Moloi, Khmaies Ouahada, and HaiLing Zhu. Performance analysis for reed solomon codes and bose chaudhuri-hocquenghem codes in digital image watermarking. In *2019 IEEE AFRICON*, pages 1–4, 2019.

[72] Yuk Ying Chung, Fang Fei Xu, and Faith Choy. Development of video watermarking for mpeg2 video. In *TENCON 2006 - 2006 IEEE Region 10 Conference*, pages 1–4, 2006.

[73] A. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Transactions on Communication Technology*, 19(5):751–772, 1971.

[74] Kunliang Yu, Liquan Chen, Zhangjie Fu, Yu Wang, and Tianyu Lu. A coding layer robust reversible watermarking algorithm for digital image in multi-antenna system. *Signal Processing*, 199:108630, 2022.

[75] Md. Ahasan Kabir. An efficient low bit rate image watermarking and tamper detection for image authentication. *SN Applied Sciences*, 3(4):400, 2021.

[76] Juan R. Hernandez, Jean-Francois Delaigle, and Benoit M. M. Macq. Improving data hiding by using convolutional codes and soft-decision decoding. In Ping Wah Wong and Edward J. Delp III, editors, *Security and Watermarking of Multimedia Contents II*, volume 3971, pages 24 – 47. International Society for Optics and Photonics, SPIE, 2000.

[77] Hicham Tribak, Zaz. Youssef, and Houria Kelkoul. Advanced video watermarking approach based on convolutional encoding : Search for new solution against cinematography piracy traffic. In *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, pages 1–7, 2018.

[78] Yun Tan, Qin. Jiaohua, Xuyu Xiang, Ma. Wentao, Pan. Wenyan, and Neal N. Xiong. A robust watermarking scheme in ycbcr color space based on channel coding. *IEEE Access*, 7:25026–25036, 2019.

[79] T. Brandão, M. P. Queluz, and A. Rodrigues. On the use of error correction codes in spread spectrum based image watermarking. In Heung-Yeung Shum, Mark Liao, and Shih-Fu Chang, editors, *Advances in Multimedia Information Processing — PCM 2001*, pages 630–637, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[80] Gaetan Le Guelvouit and Stephane Pateux. Wide spread spectrum watermarking with side information and interference cancellation. In Edward J. Delp III and Ping Wah Wong, editors, *Security and Watermarking of Multimedia Contents V*, volume 5020, pages 278 – 289. International Society for Optics and Photonics, SPIE, 2003.

[81] C.I. Podilchuk and E.J. Delp. Digital watermarking: algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–46, 2001.

[82] G.C. Langelaar, I. Setyawan, and R.L. Lagendijk. Watermarking digital image and video data. a state-of-the-art overview. *IEEE Signal Processing Magazine*, 17(5):20–46, 2000.

[83] Jean-Luc DOERR, Gwenaël; DUGELAY. A guide tour of video watermarking. *Signal processing. Image communication*, 2003.

[84] Yaxun Zhou and Wei Jin. A robust digital image multi-watermarking scheme in the dwt domain. In *2012 International Conference on Systems and Informatics (ICSAI2012)*, pages 1851–1854, 2012.

[85] Dazhi Zhang, Boying Wu, Jiebao Sun, and Heyan Huang. A new robust watermarking algorithm based on dwt. In *2009 2nd International Congress on Image and Signal Processing*, pages 1–6, 2009.

[86] Alejandra Menendez-Ortiz, Claudia Feregrino-Uribe, Rogelio Hasimoto-Beltran, and Jose Juan Garcia-Hernandez. A survey on reversible watermarking for multimedia content: A robustness overview. *IEEE Access*, 7:132662–132681, 2019.

[87] Oleg Evsutin and Kristina Dzhanashia. Watermarking schemes for digital images: Robustness overview. *Signal Processing: Image Communication*, 100:116523, 2022.

[88] Shaik Kashifa, Sravani Tangeda, Ummadisetty Kavya Sree, and V. M. Manikandan. Digital image watermarking and its applications: A detailed review. In *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–7, 2023.

[89] Farnaz Arab, Mazdak Zamani, Sasan Karamizadeh, Mojtaba Alizadeh, Saman Shojae Chaeikar, and Touraj Khodadadi. Comparison of data hiding techniques for video watermarking applications. In *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, pages 168–173, 2022.

[90] Sanjay Kumar and Binod Kumar Singh. Dwt based color image watermarking using maximum entropy. *Multimedia Tools Appl.*, 80(10):15487–15510, apr 2021.

[91] Fares Kahlessenane, Amine Khaldi, Redouane Kafi, and Salah Euschi. A dwt based watermarking approach for medical image protection. *Journal of Ambient Intelligence and Humanized Computing*, 12(2):2931–2938, 2021.

[92] Lei Wang and Huichao Ji. A watermarking optimization method based on matrix decomposition and dwt for multi-size images. *Electronics*, 11(13):2027, 2022.

[93] Priyank Khare and Vinay Kumar Srivastava. A reliable and secure image watermarking algorithm using homomorphic transform in dwt domain. *Multidimensional Systems and Signal Processing*, 32:131–160, 2021.

[94] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.

[95] J.S. Walker. A primer on wavelets and their scientific applications. *CRC Press*, (1st ed.):179, 1999.

[96] Christopher Torrence and Gilbert P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79:61 – 78, 1998.

[97] B. Schaefli, Douglas Maraun, and Matthias Holschneider. What drives high-flow events in the swiss alps? recent developments in wavelet spectral analysis and their application to hydrology. *Elsevier*, pages 2511–2525, 01 2007.

[98] Jérôme Lebrun and Martin Vetterli. Balanced multiwavelets theory and design. *IEEE Transactions on Signal Processing*, 46(4):1119–1125, 1998.

[99] S Rout and AE Bell. Narrowing the performance gap between orthogonal and biorthogonal wavelets. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, volume 2, pages 1757–1761. IEEE, 2004.

[100] Radomir S Stanković and Bogdan J Falkowski. The haar wavelet transform: its status and achievements. *Computers & Electrical Engineering*, 29(1):25–44, 2003.

[101] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE transactions on information theory*, 36(5):961–1005, 1990.

[102] B Vijayakumari, J Ganga Devi, and M Indhu Mathi. Analysis of noise removal in ecg signal using symlet wavelet. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pages 1–6. IEEE, 2016.

[103] Mohamed Elgendi, Mirjam Jonkman, and Friso De Boer. R wave detection using coiflets wavelets. In *2009 IEEE 35th Annual Northeast Bioengineering Conference*, pages 1–2. IEEE, 2009.

[104] Xiangyi Zhong, Hongxu Jiang, Haiheng Cao, and Rui Yang. Efficient lifting based cdf9/7 wavelet transform using fixed point. In *2010 3rd International Congress on Image and Signal Processing*, volume 7, pages 3094–3097, 2010.

[105] Wen-Bo Wang, He-Long Li, Xu-Ming Yi, and Pi-Sheng Fei. Sar image speckle reduction algorithm based on second wavelet packet transform. In *2009 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3628–3632, 2009.

[106] Guoan Yang, Nanning Zheng, Cuihua Li, and Shugang Guo. Extensible jpeg2000 image compression systems. In *2005 IEEE International Conference on Industrial Technology*, pages 1376–1380, 2005.

[107] Albert Cohen, Ingrid Daubechies, and J-C Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

[108] Nicholas Hopper, David Molnar, and David Wagner. From weak to strong watermarking. *IACR Cryptology ePrint Archive*, 2006:430, 01 2006.

[109] Thai Hung Pham and Minh Thanh Ta. Invariant zero-watermarking algorithm in dwt-dct domain using robust features matching. *Journal of Science and Technique-ISSN*, 1859:0209, 2024.

[110] Peter Awonnatemi Agbedemnab, Mohammed Akolgo, and Moses Apambila Agebure. A new image watermarking scheme using genetic algorithm and residual numbers with discrete wavelet transform. *Journal of Information Security*, 14(4):422–436, 2023.

[111] HAWEEZ SHOWKAT and NISA DA ROHUN. Systematic review and simulative comparison of video watermarking schemes. *Journal of Theoretical and Applied Information Technology*, 102(4), 2024.

[112] William Puech Zafar Shahid, Marc Chaumont. H.264/AVC video watermarking for active fingerprinting based on Tardos code. *Signal, Image and Video Processing, Springer Verlag*, 7(4), 2013.

[113] Fuchun Xie, Teddy Furon, and Caroline Fontaine. On-Off Keying Modulation and Tardos Fingerprinting. *Proc. ACM Multimedia and Security*, January 2008.

[114] Karama Abdelhedi, Faten Chaabane, William Puech, and Chokri Ben Amar. Toward a Novel LSB-based Collusion-Secure Fingerprinting Schema for 3D Video. In *Springer International Publishing*, Computer Analysis of Images and Patterns, pages 58–68, Cham, 2021.

[115] Minoru Kuribayashi and Hans Georg Schaathun. Image fingerprinting system based on collusion secure code and watermarking method. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2120–2124, 2015.

[116] Jalel Baaouni, Hedi Choura, Faten Chaabane, Tarek Frikha, and Mouna Baklouti. Design of multiprocessor architecture for watermarking and tracing images using qr code. In *Intelligent Decision Technologies: Proceedings of the 14th KES-IDT 2022 Conference*, pages 109–122. Springer, 2022.

[117] Faten Chaabane, Maha Charfeddine, and Chokri Ben Amar. A survey on digital tracing traitors schemes. In *2013 9th International Conference on Information Assurance and Security (IAS)*, pages 85–90, 2013.

[118] Houria Kelkoul, Youssef Zaz, Hicham Tribak, and Gerald Schaefer. A robust combined audio and video watermark algorithm against cinema piracy. In *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, pages 1–4, 2018.

[119] Sanjay Kumar and Binod Kumar Singh. DWT based color image watermarking using maximum entropy. *Multimedia Tools and Applications*, 80(10):15487–15510, 2021.

[120] Nikita Kashyap and GR Sinha. Image watermarking using 3-level discrete wavelet transform (dwt). *International Journal of Modern Education and Computer Science*, 4(3):50, 2012.

[121] Akhil Pratap Singh and Agya Mishra. Wavelet based watermarking on digital image. *Indian Journal of Computer Science and Engineering*, 1(2):86–91, 2011.

[122] Blender Foundation. Tears of steel: Open movie free to share and show. https://mango.blender.org/download/, 2012. [Online; accessed 19-July-2024].

[123] Mauro Barni. Shedding light on some possible remedies against watermark desynchronization: a case study. In Edward J. Delp III and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents*, volume 5681 of *Proceedings of SPIE*, pages 106–113. SPIE, 2005.

[124] Dariush Divsalar and Fabrizio Pollara. On the design of turbo codes. *The Telecommunications and Data Acquisition Progress*, pages 42–123, 1995.

[125] Birgit Pfitzmann and Matthias Schunter. Asymmetric fingerprinting (extended abstract). In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 1996.

[126] Birgit Pfitzmann and Michael Waidner. Anonymous fingerprinting. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 88–102. Springer, 1997.

COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Filigrane pratique pour la recherche de trafiquants multimédias

Mot clés : Attaques de collusion, tatouage vidéo, codes d'empreintes digitales, Codes correcteurs d'erreurs, étalement aléatoire

Résumé : La popularité des téléphones portables, des appareils photo numériques et des ordinateurs personnels a modifié la création, la consommation et le partage des contenus multimédias. Cela pose le problème de la duplication à l'infini, du téléchargement non autorisé et de la redistribution illégale. À l'ère de la distribution généralisée des informations numériques, il est plus important que jamais de développer des solutions fiables et solides pour éviter la distribution illégale. Un système d'empreinte multimédia est un moyen efficace de protéger le contenu multimédia et d'empêcher la distribution illégale. L'objectif de cette thèse est de trouver les individus qui ont participé à la production et à la distribution illégale de contenus multimédias. Nous avons proposé un système de filigrane vidéo aveugle par transformée en ondelettes discrète associé à des codes d'empreintes probabilistes pour contrer les attaques de collusion parmi les vidéos à haute résolution. Le filigrane robuste et aveugle conduit à un taux d'erreur binaire plus élevé, nous devons ajouter une redondance supplémentaire aux codes d'empreintes digitales pour obtenir des taux de traçage plus élevés. Pour cela, nous proposons un schéma de codage dans lequel l'étalement aléatoire est utilisé avec des codes correcteurs d'erreurs. Nous avons utilisé FFMpeg pour intégrer l'image du filigrane dans la vidéo, ainsi que pour mener une série d'attaques de collusion (par exemple, moyenne, assombrissement et éclaircissement) sur des vidéos à haute résolution et nous avons comparé les générateurs-décodeurs de codes d'empreintes digitales les plus souvent suggérés dans la littérature pour trouver l'auteur de l'attaque de collusion. L'étude expérimentale montre que notre conception est très performante en termes de repérage des fraudeurs et de temps.

Title: Practical watermarking for multimedia traitor tracing

Keywords: Collusion attacks, Video watermarking, Fingerprinting codes, Error-correcting codes, Random spreading

Abstract: The popularity of mobile phones, digital cameras, and personal computers has changed multimedia content creation, consumption, and sharing. This raises the issues of endless duplication, unauthorized uploading, and unlawful redistribution. In this era of widespread digital information distribution, it is more important than ever to develop dependable and strong solutions to avoid illegal distribution. A multimedia fingerprinting scheme is an efficient means of protecting multimedia content and preventing illegal distribution. The goal of this thesis is to find individuals who were engaged in the production and illegal distribution of multimedia content. We proposed a Discrete Wavelet Transform blind video watermarking scheme tied with probabilistic fingerprinting codes to counter collusion attacks among higher-resolution videos. The robust and blind watermarking leads to a higher bit error rate, we need to add extra redundancy to fingerprinting codes to obtain greater tracing rates. For that, we propose a coding scheme, in which the random spreading is utilized with error-correcting codes. We utilized FFMpeg to embed the watermark image into the video, as well as to conduct a range of collusion attacks (e.g., average, darken, and lighten) on high-resolution video and compared the most often suggested fingerprinting code generator-decoders in the literature to find the colluder. The experimental investigation shows that our design has high performance in terms of colluder tracing, and time.