

MOBILE CLOUD SYSTEM: SECURITY ENHANCEMENT IN MULTIMEDIA RETRIEVAL VIA
MOBILE PHONES

Mr. Pongsakorn Kingsuwankul



จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science and Information
Technology

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2014

Copyright of Chulalongkorn University

ระบบคลาวด์เคลื่อนที่: การเพิ่มสมรรถนะความปลอดภัยในการค้นคืนสื่อผสมผ่านโทรศัพท์เคลื่อนที่



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการ
คอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2557

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	MOBILE CLOUD SYSTEM: SECURITY ENHANCEMENT IN MULTIMEDIA RETRIEVAL VIA MOBILE PHONES
By	Mr. Pongsakorn Kingsuwankul
Field of Study	Computer Science and Information Technology
Thesis Advisor	Assistant Professor Dr.Pattarasinee Bhattarakosol, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

.....Dean of the Faculty of Science
(Professor Dr.Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

.....Chairman
(Associate Professor Dr.Nagul Cooharajanane, Ph.D.)
.....Thesis Advisor
(Assistant Professor Dr.Pattarasinee Bhattarakosol, Ph.D.)
.....External Examiner
(Lecturer Dr.Kanokwan Atchariyachanvanich, Ph.D.)

พงศกร กิ่งสุวรรณกุล : ระบบคลาวด์เคลื่อนที่: การเพิ่มสมรรถนะความปลอดภัยในการค้นคืนสื่อผสมผ่านโทรศัพท์เคลื่อนที่ (MOBILE CLOUD SYSTEM: SECURITY ENHANCEMENT IN MULTIMEDIA RETRIEVAL VIA MOBILE PHONES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.ภัทรสินี ภัทรโกศล, 72 หน้า.

มีสื่อไร้ลิขสิทธิ์จำนวนมากบนโลกอินเทอร์เน็ตในขณะที่ผู้คนส่วนใหญ่ปราศจากความรู้ในเรื่องของลิขสิทธิ์ กลุ่มผู้ใช้บางคนที่รับรู้เรื่องนี้ก็คิดว่าเป็นประเด็นที่มีความสำคัญเพียงเล็กน้อยและไม่ได้ให้ความสนใจเกี่ยวกับเรื่องนี้เลย คนส่วนใหญ่จะดาวน์โหลดเพลงหรือภาพยนตร์เพื่อความบันเทิงของตนในที่พักซึ่งเป็นที่ยอมรับกันได้โดยทั่วไปแต่จะต้องไม่ใช้การนำมาเพื่อจำหน่าย เผยแพร่ หรือผลิตเป็นอุตสาหกรรม ผู้ใช้ที่สามารถดาวน์โหลดสื่อต่าง ๆ อย่างอิสระอาจส่งผลกระทบต่อผู้เป็นเจ้าของสื่อได้ กล่าวคือ การดาวน์โหลดไฟล์วีดิทัศน์โดยปราศจากการได้รับอนุญาตจากเจ้าของจะถือเป็นอาชญากรรมทางไซเบอร์ การจับบุคคลผู้ซึ่งกระทำผิดเหล่านี้หรือการบล็อกไซต์ต่าง ๆ เพื่อมิให้กระจายสื่อมักใช้เวลาในการดำเนินการนานกว่าจะสืบหาแหล่งที่ดำเนินการเหล่านี้ได้ ดังนั้นประเด็นในการสร้างความเข้มแข็งให้แก่การแพร่กระจายวีดิทัศน์ควรเป็นประเด็นที่นำมาพิจารณาเพื่อเพิ่มความปลอดภัยของกระบวนการระบุตัวตนของผู้มีสิทธิ์ในการใช้ งานวิจัยนี้ได้นำเสนอแนวคิดของการแบ่งส่วนหัวของสื่อสู่ระบบโมบายคลาวด์เคลื่อนที่ ซึ่งผลที่ได้รับแสดงให้เห็นได้ว่า กลไกที่นำเสนอสามารถป้องกันการแพร่กระจายสื่ออย่างผิดกฎหมายได้แล้ว ยังสามารถร่นระยะเวลาการส่งสื่อได้สั้นลงกว่ากระบวนการส่งสื่อโดยทั่วไปอีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ	ลายมือชื่อ อ.ที่ปรึกษาหลัก
ปีการศึกษา	2557	

5572632323 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORDS: COPYRIGHT, CYBER-CRIME, VIDEO DISTRIBUTION, SECURITY, AUTHENTICATION, MOBILE CLOUD, SEEDER, LEECHER.

PONGSAKORN KINGSUWANKUL: MOBILE CLOUD SYSTEM: SECURITY ENHANCEMENT IN MULTIMEDIA RETRIEVAL VIA MOBILE PHONES. ADVISOR: ASST. PROF. DR.PATTARASINEE BHATTARAKOSOL, Ph.D., 72 pp.

There is free media available all over the Internet while most people have no idea about its copyright. Some of those who know perceive it as a minor issue and do not care about it. Most of the people download songs or movies for household entertainment which is fine, but not for those who do it for sale, distribution and manufacture. The users can download the media freely which may have an adverse effect on the media owners. Downloading video files without the permission of the owner is a cyber-crime. Arresting all persons who commit such crime or blocking those sites releasing the media will require complicated and long process to investigate. Hence, strengthening the security issue in video distribution should be practiced. To increase the security of authentication process for the real users, this research proposes the concept of media header file division in Mobile cloud system. The result shows that the proposed mechanism can protect the illegal distribution of media and the transfer period is shorten than general media transfer process.

Department: Mathematics and Student's Signature

Computer Science Advisor's Signature

Field of Study: Computer Science and
Information Technology

Academic Year: 2014

ACKNOWLEDGEMENTS

This Thesis has been completed with supports from many people regarding useful knowledge, information and guidance. I would like to whole-heartedly thank all those who involve in the making of this Thesis even in a slightest portion of it.

I sincerely thank my advisor Assist. Prof. Dr.Pattarasinee Bhattarakosol, for an insistent guidance, motivation and immense knowledge to guide me through any problems occurred during the making of this Thesis.

I would like to thank the Thesis committee, Assoc. Prof. Dr.Nagul Cooharajanane and Lect. Dr.Kanokwan Atcharyachanvanich for an advice in perfecting this Thesis.

I thank my INSET lab juniors, seniors and friends who had helped regarding experiments, necessary devices and documentation.

Finally, I am grateful to every members in Kingsuwankul family for giving me this opportunity and supporting me through thick and thin.

TABLE OF CONTENTS

	Page
THAI ABSTRACT	iv
ENGLISH ABSTRACT	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
List of Tables	x
List of Figures.....	xii
CHAPTER 1 Introduction.....	1
1.1 Problem Statement	2
1.2 Objective.....	3
1.3 Scope of Thesis.....	3
1.4 Definition	3
1.5 Structure of the Thesis.....	4
CHAPTER 2 LITERATURE REVIEW AND RELATED COMMON KNOWLEDGE.....	5
2.1 Intellectual property and Digital Right Management.....	5
2.2 Cloud computing	8
2.3 Mobile Cloud Computing.....	10
2.4 Peer-to-peer files sharing system (P2P files sharing system)	10
2.5 Multimedia File Structure.....	11
2.6 Typical video file structure and its header.....	12
2.7 Asynchronous Transfer Mode	14
CHAPTER 3 METHODOLOGY	15
3.1 Proposed Method	15

	Page
3.1.1 Multimedia file retrieval via a mobile cloud system in [1]: Overview	15
3.1.2 The Intellectual property issue.....	17
3.2 Security Enhancement using the MSSE	18
3.2.1 Media Split	18
3.2.1.1 Comparison between code and third party program split	19
3.2.1.2 File parts protection: Security Enhancement.....	20
3.2.2 Media Request	21
3.2.3 Media Join	21
3.2.4 Media Play and Delete.....	22
3.3 Split-and-Join Code Architecture	23
3.3.1 The Split Code	23
3.3.2 The Join Code	29
3.4 Calculation on the aggregate size of media files part after the split	30
CHAPTER 4 EVALUATIONS AND EXPERIMENTAL RESULTS	33
4.1 File content verification.....	33
4.1.1 Using properties windows	33
4.1.2 Using MP4 Reader.....	35
4.1.3 Using Hash Calculator.....	38
4.1.4 Content verification test results.....	39
4.2 Results on security enhancement	41
4.3 Aggregate size result tables.....	44
4.4 Transfer speed analysis	51
4.4.1 Multimedia file transfer from seeders to leecher.....	51

	Page
4.4.2 The experimental simulation and results.....	53
4.5 User satisfaction evaluation.....	56
4.5.1 Experimental procedures and results	57
4.5.2 User evaluation discussion.....	58
CHAPTER 5 CONCLUSIONS AND DISCUSSIONS.....	60
5.1 Discussion.....	60
5.2 Limitation of the experiment.....	61
5.3 Conclusion	62
REFERENCES	63
APPENDIX.....	69
VITA.....	72



List of Tables

	Page
Table 3.1: Summary of variables definitions.....	31
Table 4.1: ftyp content in hexadecimal value read by MP4 Reader.....	35
Table 4.2: moov content in hexadecimal value	36
Table 4.3: mdat part in hexadecimal value	37
Table 4.4: MD5 and SHA1 hash comparison	38
Table 4.5: MP4 sample file information after undergoing split-join process.....	39
Table 4.6: AVI sample file information after undergoing split-join process.....	40
Table 4.7: 3GP sample file information after undergoing split-join process.....	40
Table 4.8: WMV sample file information after undergoing split-join process	40
Table 4.9: MP3 sample file information after undergoing split-join process.....	41
Table 4.10: Sample video information on security enhancement test.....	41
Table 4.11: Results on security enhancement test.....	42
Table 4.12: MP3 sample file net size increase	45
Table 4.13: 3GP sample file net size increase	46
Table 4.14: MP4 sample file net size increase	47
Table 4.15: 3GP sample file net size increase	48
Table 4.16: WMV sample file net size increase.....	49
Table 4.17: AVI sample file net size increase	50
Table 4.18: Typical smartphone and laptop specification comparison	54
Table 4.19: Multimedia retrieval timing results.....	55
Table 4.20: Users feedback on quality and security after the experiment	58

Table 5.1: Comparison amongst other DRM techniques..... 61



List of Figures

	Page
Figure 3.1 Leecher requests to watch a video file.	16
Figure 3.2 Use case diagram of the original mobile cloud system proposed in [1]	17
Figure 3.3 The media header is split and stored in the server.	18
Figure 3.4 Process of splitting a media content	19
Figure 3.5 Segmentation of a media file using a third party program.....	20
Figure 3.6 Segmentation of a media file using JAVA code.....	20
Figure 3.7 An authenticated user requests for a media file to watch.	21
Figure 3.8 Joining a Media File using Code and Removing an Insertion Part.....	22
Figure 3.9 New seeder is created after media is played.	22
Figure 3.10 Multimedia file samples in the directory	23
Figure 3.11 Pseudocode of SplitHead module.....	24
Figure 3.12 Sample file gets split into header and content file.....	24
Figure 3.13 SplitHead Log Process.....	25
Figure 3.14 Pseudocode of SplitContent Module.....	26
Figure 3.15 A Sample file gets split into equal parts.	27
Figure 3.16 SplitContent Log Process	28
Figure 3.17 Pseudocode explaining JoinAll	29
Figure 3.18 Sample file parts are joined back.	30
Figure 3.19 JoinAll process log data.....	30
Figure 3.20 Explanation of how net size will get increased after the split.....	31

Figure 4.1 (A) Original video properties (B) Joined video properties after split and join process.....	34
Figure 4.2 (A) Original video details (B) Joined video details after split and join process.....	34
Figure 4.3 MP4 file security enhancement result	42
Figure 4.4 3GP file security enhancement result	43
Figure 4.5 AVI file security enhancement result.....	43
Figure 4.6 WMV file security enhancement result.....	43
Figure 4.7 MP3 sample file net size increase graph	45
Figure 4.8 3GP sample file net size increase graph.....	46
Figure 4.9 MP4 sample file net size increase graph	47
Figure 4.10 3GP sample file net size increase graph.....	48
Figure 4.11 WMV sample file net size increase graph.....	49
Figure 4.12 AVI sample file net size increase graph	50
Figure 4.13 Variation in net size on sample videos graph	51
Figure 4.14 A leecher is requesting for media when there is no seeder.....	52
Figure 4.15 A conceptual model of multimedia file retrieval via mobile cloud system.....	52
Figure 4.16 Experimental simulation model	54
Figure 4.17 Mean File Retrieval Time graph.....	56
Figure 4.18 Test users' opinion on quality of sample video.....	57
Figure 4.19 Test users' opinion on security enhancement	58

CHAPTER 1

Introduction

Currently, the matter of media copyright has become one of the main online issues which should be provided more attention. People have been taking things for granted by obtaining free media that they should be paying for through the Internet. Nobody wants to work for free, so as the artists and the movie producers. Imagine those people creating media work with toil and all other people just want to acquire them for free.

Media files scattered all over the Internet are mostly without any copyright protection. As the matter of fact, the best way to stop a cyber-crime is to create ethical minds in Internet people, but that is as hard as telling people to conserve nature. Some people do, the rest of them do not. When asking those online people to stop downloading the media illegally does not make any difference, the downloadable media security authentication itself has to be enhanced.

Peer-to-peer file sharing system is a technique that allows large scale of the Internet users to download as well as upload computer files in a rapid speed. It is apparently almost impossible to control an online distribution of multimedia file. Therefore if people need a copyright protection, it has to be done on the multimedia files themselves.

Digital Right Management is the technology used by the copyright owners to maintain and manage the authentication or use of their multimedia products like songs, movies or other software. The term should not be confused with Copy Protection. Although DRM proposes that downloading media is acceptable for household entertainment, many of them downloaded and distributed as a business.

Some technique like cloud computing system is used to reduce delay of file retrieval over Internet or a computer network. But for mobile cloud system like in [1],

media files are split into many small parts. Each part of media file is accessible by any user and might lead to partial or wholly violation of media copyright. This paper will emphasize on protecting file parts so that each of them will not be playable by any typical video players.

This chapter proposes the problem in Section 1.1, and the objective is described in Section 1.2. In Section 1.3, the scope and constraint of this Thesis will be discussed, followed by definitions of technical terms in Section 1.4. Finally, the structure of this Thesis is detailed in Section 1.6.

1.1 Problem Statement

In mobile cloud system, multimedia retrieval is fast but still not safe regarding the copyright issue. While a multimedia file is divided into smaller parts in which they will be stored in the seeders, the parts stored in the seeders may be accessible and playable by the seeder itself and can lead to the illegal distribution of the media by those who do not have authentication.

Hence, by this way copyright can be violated both partially and wholly. In this Thesis [1], third party program is used to split the media file. As a consequence after the file was split, each individual part that is stored in the seeder can be played normally because the third party file splitter provided the header to each file part. If the user that is not eligible somehow has an access to the file part stored in his device, then he can acquire partial or even full media file. Moreover the total size of all file parts is comparatively larger after the split because the header file is included to each of the parts.

1.2 Objective

The objective of the proposed system is to propose a mechanism to protect multimedia online downloads from breaking intellectual property right and apply cloud mechanism to enhance the download speed for multimedia retrieval via mobile cloud system.

1.3 Scope of Thesis

In this study, the classification system is constrained as follows:

1. Sample video size is not larger than 100 Mb and not smaller than 2 Mb.
2. Wi-Fi connection is necessary in the transferring process.
3. File parts size will range from 64 KB up to $\frac{\text{Nearest size of } 2^n \text{ KB}}{2}$.
4. Assuming the distance between all seeders and a leecher to be the same.
5. Assuming bitrate of all seeders to be equal.

1.4 Definition

User: A member who seeks to watch multimedia file from the system.

Seeder: A registered member of the system who provides storage to file parts that will be requested by a seeder when needed. It is a multimedia sender.

Leecher: A registered member of the system requesting for a multimedia file from a mobile cloud or typically a computer network. It is a multimedia receiver.

Mobile Cloud: A group of active seeders available in the system. Users with smartphones can be considered seeder of the system. Each active seeder contains the parts of multimedia file in the system.

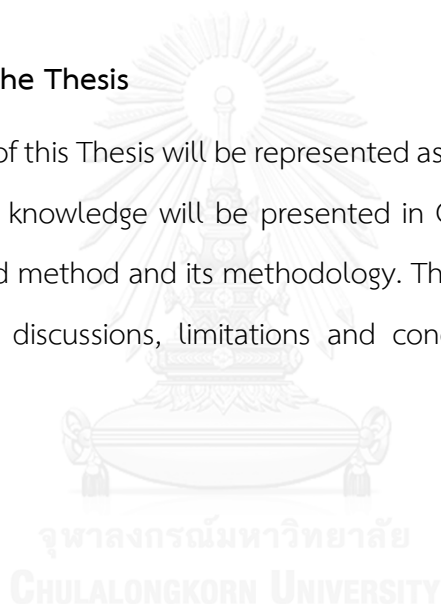
Central Server: A high performance computer acting as a central nervous system that control and update every action in the system. It takes up requests from users and update necessary information with mobile cloud.

Server Database: A physical data storage where the Central Server stores its data if needed.

The MSSE: The Multimedia Segmentation for Security Enhancement, name of the proposed method.

1.5 Structure of the Thesis

The structure of this Thesis will be represented as follows; the literature reviews and related common knowledge will be presented in Chapter 2. The Chapter 3 will describe the proposed method and its methodology. The experimental results will be in Chapter 4. Lastly, discussions, limitations and conclusion will be illustrated in Chapter 5.



CHAPTER 2

LITERATURE REVIEW AND RELATED COMMON KNOWLEDGE

In this chapter, researched literature reviews and necessary fundamental knowledge have been briefed. These include intellectual property (IP) and digital right management (DRM), cloud computing, mobile cloud computing, peer-to-peer file sharing system, media file structure, typical video file structure and its header, Asynchronous Transfer Mode.

2.1 Intellectual property and Digital Right Management

Intellectual property is not physical; it is basically a conceptual and outstanding idea that is a result from brainstorming. It can also be in the form of products that includes the unique style such products, any man-made creativity, or innovative inventions. The intellectual property can be both tangible as well as intangible. The examples of tangible ones are physical products, OTOP products, and electronic hardware specs. The examples of intangible ones are service, idea, business plan, creative style, user interface, computer program mechanism. Some intangible intellectual property needs a core and serious strategy [17], [18] to prevent any violation in such rights.

From the past decades, Information and Technology has taken up major roles in the human's world. Problems, such as product duplication, idea copying, pirating programs have been practiced widely. Although there are several laws protecting it, but most people especially the Internet users are just careless to concern about such thing. Typically, the intellectual property is categorized into two main parts namely; Industrial property and copyright [22].

1. Industrial Property mainly concerns with the idea of the physical products, its unique style, or can say the thing that make a product outstands the others such as taste, smell, packaging, process of manufacture, trademarks, logo, layout, patent etc.
2. Copyright is the right of an individual, a company or an organization to own, publish or distribute their products legally. Some unique things need to have a copyright to protect them from being duplicated or distributed without the owner's permission, such as literature, building design, traditions, musical work, art work, films, and other multimedia.

Idea and creativity do not drop from the sky, some people spent weeks or even years to create one small piece of work. To protect the innovative creation from the copy-paster, copyright is a result from such brainstorm, skills, toils that build up the uniqueness of the products [23]. Hence, the owner needs this kind of right protection.

Some digital products like vector graphic [15], audio [8] [14] [16], video file and other electronic devices [10], also need some protection mechanisms. The copyright owner can range from a single individual to a large corporation. Only the copyright owner will have a sole authority to use their copyright in realizing benefits in any form, such as duplication, voice or video recording, publishing, renting, transferring, authorizing, and distributing, both partly and wholly.

A copyright is obtained automatically right away when the creativity or product is produced without having to register anywhere in a form of document. The creator just needs to have a historical proof showing the actual creation of the creativity or product to claim its possession. Although with that being said, the local Department of Intellectual Property provides the service to register any forms of products or creativity's copyright. This will create a database that can be checked easily in case of copyright violation [23]. The copyright owner can be any individuals or organizations that is related to the creativity or products, such as a creator, a creator's partner/worker, an authorized editor, and a copyright transferee.

Digital Right Management is a set of techniques used to protect any form of digital multimedia from getting its copyright violated. Multimedia includes the file content of images, sound, videos, songs, documents etc. Manufacturers, publishers, copyright holders and individuals used DRM to put constraints to the use and distribution of the digital content and devices. DRM is usually not totally a copy protection, but it is more about preventing leakage partial or whole content of the multimedia files [24]. Unfortunately, no matter many new preventing mechanisms have been implemented, there will always have the backdoor to get away.

It is apparent that the cyber society is corrupted [19] and people do not care when it comes to online world, they just want to do what they want. Although the best way to protect digital right is to rise the ethics in Internet people mind or tackle it with proper strategy [20], but, because very handful people are aware of digital right and multimedia copyright. There exist such people who know but do not care about. Thus, there is a significant need of protection techniques for multimedia files. The mechanism can be performed by any groups or individual; the copyright owner, file hosts, and clients. There is a huge variety of copyright protection mechanisms that have been produced consistently to keep up with the new method of duplicating and different mechanisms serve different purposes. Many techniques have been invented to create such new method. Below are some examples worth mentioning regarding techniques in the field of DRM.

- Re-arranging file structure and joining it back in [5]. MP3 file is disintegrated and rearranged to form a totally playable file, but rather not-smooth to listen to, if the user is unauthenticated.
- Adding a new tag name 'DoRM' used to store deciphering parameters inside the MP3 file structure like in [8] is also another approach to make the file not-playable if the user is unauthenticated by the system.
- A new 'Potato system' in [7] proposed a systematic workflow for media distribution amongst artists, media distributors and consumers/users.

- In [10], digital resources are secured using techniques of file encryption and inserting a private key. Alternatively, storing the file contents behind the firewall.
- Lotspiech [11] also proposed conceptual views of how to apply an encryption that enables a protection mechanism on multimedia distribution. It can be applied to some consumer products too.
- Using fingerprinting mechanism in [12] is also a strong approach for media file verification by embedding a generated fingerprinting code into the file which will be verified whenever the user requests for it.
- DRM involving watermark or Digital watermarking is an effective methodology for content authentication and prevent forgery individual file distributed all over the Internet from an unauthorized users [13][21].
- Watermarking technique is also applicable to an audio file [14][16]. A scheme to protect the audio data by embedding high capacity robust watermark to it is proposed. The watermark removal can be performed using a secret watermarking key with only minimal remaining distortion while the protection watermark embedding and detection is done in the spectral domain.

2.2 Cloud computing

The word 'Clouds' or 'Cloud computing' has been gaining its significance in a past couple of years. Its meaning may vary slightly as for different aspects, but the main idea of the cloud computing stays the same. For the business aspect, it refers to the flexibility, ability to expand, and a kind of "you-pay-what-you-own" system. Due to this property of clouds, many organizations have saved up a huge amount of budget [27]. To make it simple, people can think of cloud computing to be like putting everything to a central server and called it 'Cloud'. Now when a user wants to process anything, there is no need to have a high specific computer to process their complicated tasks since the task will be assigned to compute at the 'Cloud'. While

many users use the same 'Cloud', the efficiency of the system can be improved at once. To increase the efficiency of the whole system, simply improve the specification of the 'Cloud' instead of improving each client's computer. To obtain a clear picture of cloud, the reader can imagine an internet cafe that the client computer does not contain any hard disk, but everything is linked into/from a server computer. The request is sent to the central server to process [25].

Apart from its instant accessibility high security [26], these points below sum up major properties that strengthen the cloud computing.

- **Virtualization** – Cloud computing is a virtualized system which uses the technology to cope up unworthy and inefficient usage of hardware. The system is virtualized up to reduce the maintenance cost, energy consumed, and to be more environmental-friendly. This technique suits up with the increasing of the Internet users and smartphone users presently. While the content or information over websites have been changed all the time, cloud computing provides such flexibility. This virtualization idea creates an unlimited view of the pooling resources.
- **On-demand computing** – The technique was developed to meet an enterprise fluctuating demands. Cloud computing can operate its resource on the judicious basis. Computing resources are made available to the user only when it is needed hence resource is utilized only as much as it has been requested. This helps the system to be able to customize its size and scale up according to the request more effectively. For example, resources like CPU cycles, bandwidth availability, data storage and application software can be arranged to users based on their tasks.
- **Abstract processing** - Cloud computing is an internet-based system that hides its complex mechanism from the clients which means users will never know what happens in the cloud. Another advantage of cloud computing is that each client does not need to install every software into their computer; they just need Internet to process their work.

2.3 Mobile Cloud Computing

Since mobile devices have dominated the IT world recently [2], it has been a big challenge for developers to expand the limit of efficiency of all small-medium devices. The technique of cloud computing allows small devices like smartphones to extend their limitation. Small mobile devices can virtually store a huge amount of data with the help of cloud storages. The device needs to be connected with the Internet and have access to the cloud. The main work of small devices will mobiles here is just to receive the input value and forward to process at the cloud. Hence, the cloud computing has solve many problems regarding of mobile devices like low processing efficiency, small device storage, less RAM etc. [28].

2.4 Peer-to-peer files sharing system (P2P files sharing system)

While in client-server system, there is a central server centralizing and managing the whole system with all clients connected to it, the computers in peer-to-peer files sharing system are similarly the same. The connection and communication in P2P network is established amongst all peers without any needs for a central server. The system allows users to share files or resources amongst one another directly [30].

The advantages of P2P files sharing system over client-server regarding files sharing are:

1. Large scale of clients requesting for files from a server can cause it to overload, while no such problem occurs in P2P files sharing.
2. P2P is more reliable since it is the server-independent, if one computer fails the other can still transfer the requested files.
3. No client computer needs to run at full-time, while in the client-server system, the server must always be online in order to satisfy the users' requests.

4. Overall cost of building and maintaining the system is comparatively lower.

Although a P2P file sharing is fast and comfortable, there are many risks regarding security and copyright issues that need proper measures to tackle with [31].

2.5 Multimedia File Structure

A video format contains mainly of two things; container and codec, which distinguish one kind of video format from another. A format is a structure which information about the file is encoded. A video file needs large space of the storage to store data that will represent its content. The amount of data determines video signal accuracy, resolution, audio level and etc. This information is compressed and stored in a container. The container acts as an empty file and codec acts as a content inside it. If a container can hold a variety of codec, it is considered to be a good container. .mp4, .avi, .mov are the examples of containers that are able to contain huge variety of codec [8].

Container

A container specify the structure of a file where the information will be stored inside, and the types of codec will be used with the file. It also determines both audio and video codec.

Codec

A codec stands for coder/decoder is a method or format to encode video and audio into a container. The quality of video is determined by codec. Each codec provides different features and quality of the video output.

Size and the quality of video files

There are some factors responsible for the size and quality of video. Roughly a digital video needs 200 MB for a minute video. FPS or frame per sec is a counting of how many images can still handle in one second. FPS of the video determines its smoothness while the standard FPS is at 29.97. Bitrate is a measurement of the number of bits that are transmitted over a set of time. Resolution is the number of pixels of the video, higher the resolution, smoother the video.

A default or installed media player cannot play every video file format. It depends on what type of decoder that media player is having. It is worth to mention that VLC (VideoLan Client) is a free video player that can play almost all video codecs.

2.6 Typical video file structure and its header

While most video file formats have similar structure, they all are different and have to be decoded differently. The discussion about the structure of typical video file format like MP4 and 3GP are drawn below.

The basic video structure and some specific terms are defined as follows [2] [6] [9].

- ftyp – file type
- moov – a tag containing important data
- free - empty area to separate header and data
- mdat - contains the audio and video frames

ftyp

The file structure consists of a series of 4-byte lengths followed by multiple 4 byte tags.

Example:

00 00 00 00 00 00 00 00 18 66 74 79 70 33 67 70 35 00 00 03 00 *ftyp3gp5*

00 00 00 1C 66 74 79 70 69 73 6F 6D 00 00 02 00 *ftypisom*

The file started with the header of 4-byte length (usually with 0x14, 0x18, 0x1c 0x20 or 0x24) followed by the string 'ftyp'. After that, the next 4 bytes specify the file type like 3gp, or mp4.

moov

Example:

00000010 6D 70 34 32 6D 70 34 31 00 00 0B E6 6D 6F 6F 76 *mp42mp41 æmoov*

The next logical tag is often a 'moov'. A moov tag contains meta-data pointers, such as song title, authors, url, other infos, etc. The structure is complex and it can allow multiple tracks, typically one for video, and the other for audio.

mdat

Followed by the last tag, it is called 'mdat'. It stores video and audio data.

Remaking the files will require the process of analyzing moov tags and check where the video frame starts. Usually most of the videos start with the fixed set of data. So it is easy to check data cluster whether they contain a frame start or not.

On many files, the start of the video frame is the fixed data, and so data cluster can be searched for and tested to make sure that it contains a starting frame (or starts) at a known offset within the cluster.

2.7 Asynchronous Transfer Mode

Asynchronous Transfer Mode or ATM is a high-speed transfer standard used to transfer variety types of files like audio, video, document etc. ATM transfers files in many small parts but in fixed size blocks called cells. ATM is developed to be used with the system that needs high-speed file transfer. The transfer rate ranges from a few Mbps upto several Gbps. ATM is sometimes called cell relay technology.

ATM generally refers to the traffic dealing with fix-size packets called ATM cells. These cells measured to be 53 bytes long; 5 bytes for Header part and the rest 48 bytes for Payload. The two biggest advantages which have made ATM so successful are:

- Queueing delays have been reduced significantly and consequently reducing wait time due to its fix-size transfer. This allows real-time audio and video streaming to run a lot smoother.
- ATM cells can be transferred in a fast rate in switch due to its short fix-size.

CHAPTER 3

METHODOLOGY

This chapter describes the proposed method that enhances the security in multimedia file sharing and storing in emphasis on P2P file sharing via a mobile cloud system. The content will include overall idea and methodology that will be implemented to create new DRM technique.

3.1 Proposed Method

The objective of the proposed method is to control the intellectual property of the multimedia file parts stored in the seeds in a mobile cloud system. The technique can be implemented to enhance the performance of Kaewmahingsa and Bhattarakosol [1] in both security [29] and file transfer speed aspect. This method involves splitting files into many fixed size blocks to make the transfer rate faster as used in Asynchronous Transfer Mode. In this paper, the proposed method will be referred to as Multimedia Segmentation for Security Enhancement or the MSSE. Although the MSSE can be applied to a typical file transfer technique but this technique will emphasis on the multimedia retrieval via mobile cloud system in [1]. Before getting into details of this Thesis, the overview of Kaewmahingsa and Bhattarakosol [1] must be understood.

3.1.1 Multimedia file retrieval via a mobile cloud system in [1]: Overview

In a P2P file transfer, especially in a mobile cloud system, multimedia file parts will be stored in each mobile called a seeder. Only the member that is registered to the system will be allowed to have access to the lists of multimedia files stored in the mobile cloud system. When a media requester, called leecher, requests to watch a video clip, it sends a request to a central

server which will locate the nearest seeders to stream video file that they are storing to the leecher, as shown in Figure 3.1.

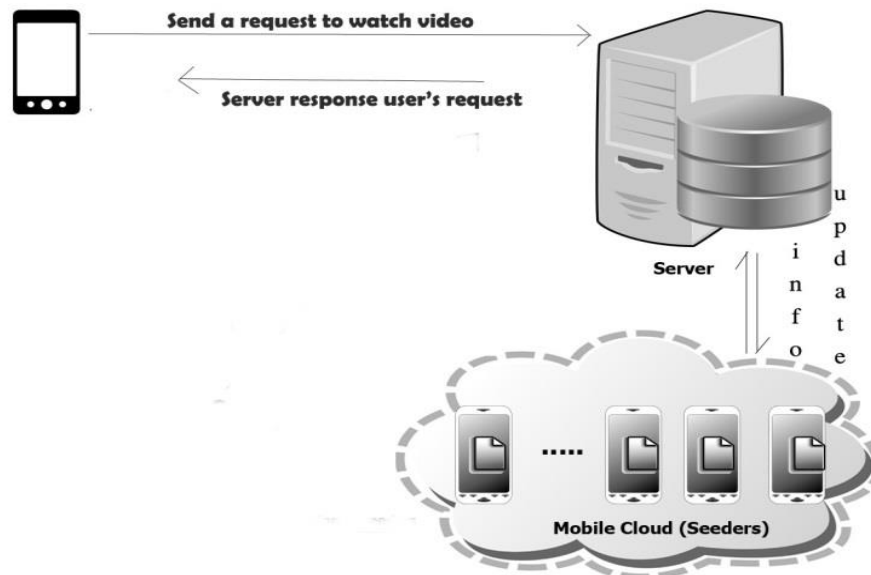


Figure 3.1 Leecher requests to watch a video file.

After playing the media or turning off the application, the requested media file will be erased from the new leecher, leaving only a small part of it in the leecher's mobile. As a consequence, the leecher changed its role to be as a new seeder for the system. Data of the part stored in this new seeder is updated to the central server.

When a new media file is added to the mobile cloud system, it will be split using a third party program and stored into seeders under the mobile cloud system. Use case diagram shown in Figure 3.2 shows the overview of the previous mobile cloud system [1].

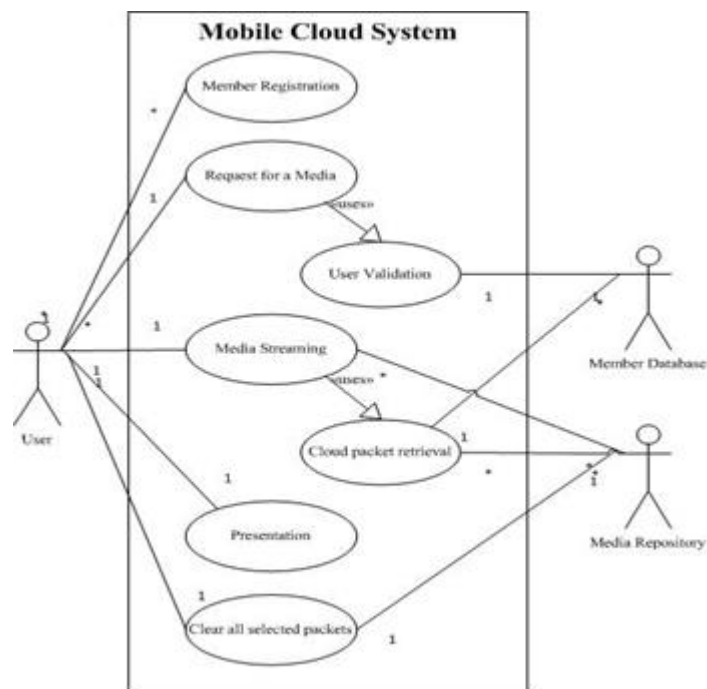


Figure 3.2 Use case diagram of the original mobile cloud system proposed in [1]

3.1.2 The Intellectual property issue

In order for playing a media, the header of the media file is checked by the media player software. If there is no header then a media player cannot play it. Each seeder will contain parts of video files which can be accessed by the seeders themselves. This may violate copyright of the media files stored in them both partially or wholly. In order to protect media files, the protection should be embedded in the file parts during splitting the media file. This Thesis mainly emphasizes on the point that file parts stored in each seeder should not be playable. In the other word, the chance that each part of the divided video files can be played individually is eliminated. This can be considered as another way to control the media copyright and strengthens the Digital Right Management.

With the overview of Kaewmahingsa et al. being cleared out, this paper's methodology will be discussed in Section 3.2.

3.2 Security Enhancement using the MSSE

The process will be divided into four phases as described below:

3.2.1 Media Split

3.2.2 Media Request

3.2.3 Media Join

3.2.4 Media Play and Delete

3.2.1 Media Split

When a new media file is added to the mobile cloud system, the file was split into small parts and stored in different seeders. Using JAVA code, this research splits files into an equal size. The first part will be the header of the video file having size of 4 bytes; this part will be sent to store in the server database, as shown in Figure 3.3.

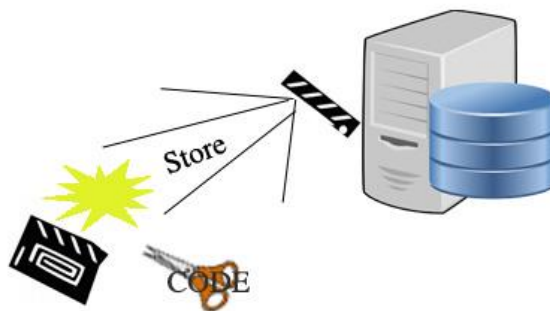


Figure 3.3 The media header is split and stored in the server.

The rest of the file content is divided into parts of size 2^n KB (i.e. 512 KB) and transferred to each seeder. The last part of the file will always be the remainder part and usually will not be equal to 2^n KB size. The size of this last part will be set to be 2^n KB, the same as the others. For the example, if all the other parts are of 512 KB and the last part is 322 KB, an insertion part of size 190 KB will be filled, as shown Figure 3.4 below.

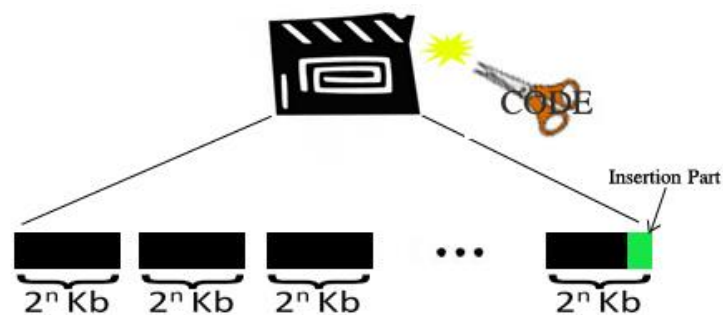


Figure 3.4 Process of splitting a media content

The size and the name of the file will be recorded precisely to the server database; this process is essential in joining these file parts back.

3.2.1.1 Comparison between code and third party program split

Directly splitting the file using code is different from using any third party programs. While the typical media splitter provides each file parts media header, Figure 3.5, using JAVA code to split does not. Each part will have no header and the content will not be playable by the media player, as shown in Figure 3.6.

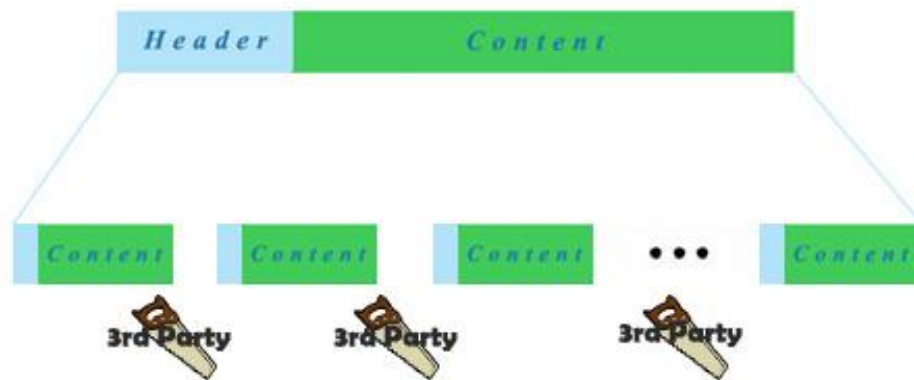


Figure 3.5 Segmentation of a media file using a third party program

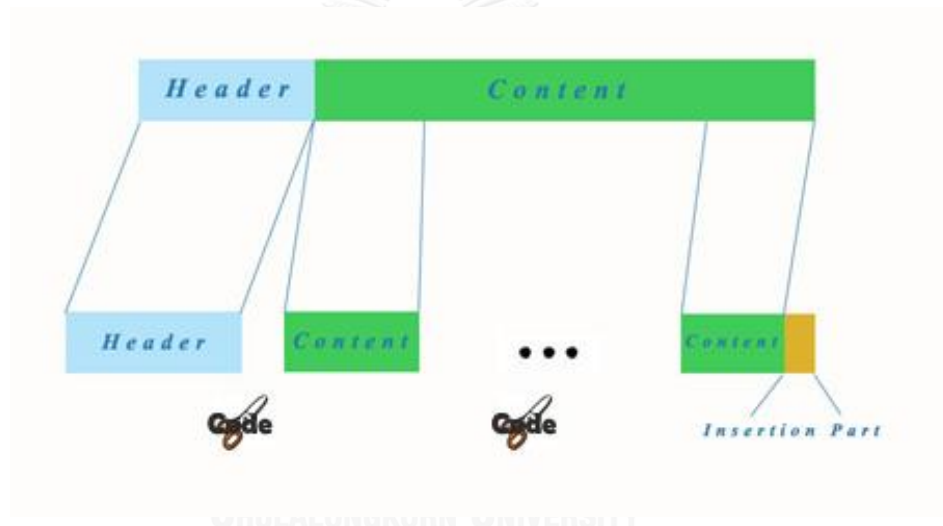


Figure 3.6 Segmentation of a media file using JAVA code

3.2.1.2 File parts protection: Security Enhancement

The following video sample had been split using both mentioned method and the parts are tested independently to play using variety of well-known video players. The third party program used is a versatile free-ware, YAMB, which can split and join a video file. In the end, all file parts are joined back and the quality of audio, video resolution, video size are equally the same as the original video. The test result will be discussed in Chapter 4.

3.2.2 Media Request

When an authenticated user wants to acquire a media, the request is sent to the central server which locates the required parts of video file in the nearby seeders. The media header will be downloaded from the server and the rest of the parts will be downloaded from the seeders in the mobile cloud system, see Figure 3.7.

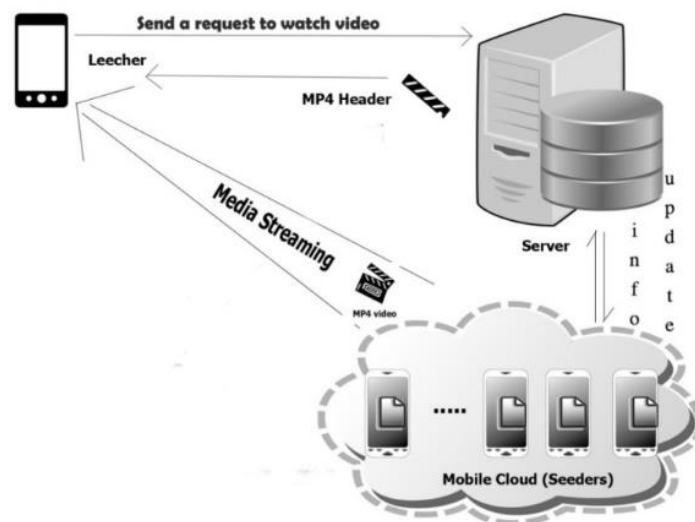


Figure 3.7 An authenticated user requests for a media file to watch.

3.2.3 Media Join

Once all video file parts and header are assembled, the file parts will be appending together using the join code. The central server updates the leecher about the name of the last part and its original size. The join code will identify and convert the last part back to its original size before the joining phase begins. An insertion part is removed in this process. The process is shown in Figure 3.8.

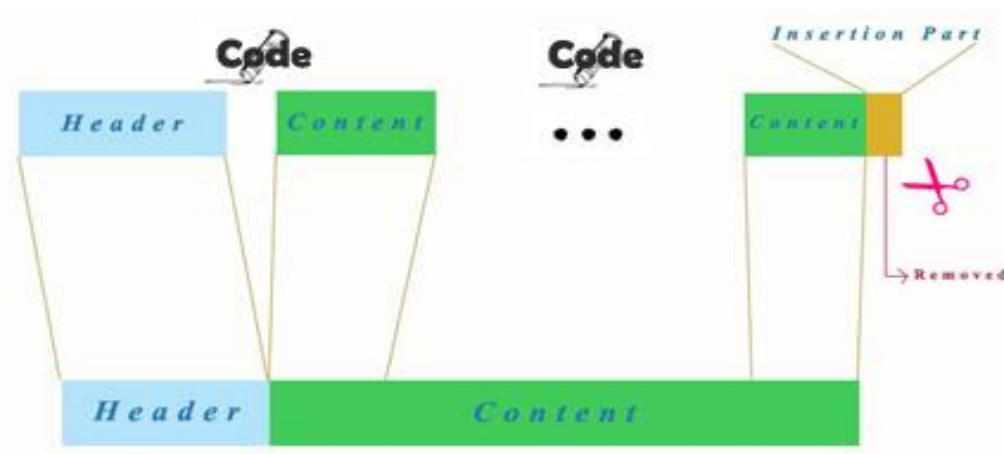


Figure 3.8 Joining a Media File using Code and Removing an Insertion Part

3.2.4 Media Play and Delete

Once the video has been played and the application is terminated, video content will be deleted, leaving only small parts of the video in this leecher. So the leecher will be as a new seeder to the mobile cloud system. The header of the media is ensured to be deleted to prevent any unauthenticated accessibility of the video content. Data of the parts stored in a new seeder is updated to the central server so that it has a record of which media is stored in this seeder. Typically in P2P file sharing, more of the seeders, faster the transfer rate. The results will be in Chapter 4.



Figure 3.9 New seeder is created after media is played.

3.3 Split-and-Join Code Architecture

Split-and-join-file modules can be written on almost any platforms and coding languages. It constitutes of a simple file dissection and re-assembly but the value must be precise to get a high accuracy performance. In this paper, the following codes have been written in JAVA but translated to a set of Pseudocode for a better understanding. Eclipse is used to run the code on the sample media files. The OS of the split-join testing pc is Windows 7. The following sample will simulate the situation in a pc directory when a video file will be split and joined back. Figure 3.10 shows the sample files initial status in the directory.









Name	Date	Type	Size	Length
 Original Sample	14/3/2558 12:22	GOM Media file(.mp4)	37,681 KB	00:01:29
 Sample_3GP	14/4/2558 14:04	GOM Media file(.3gp)	12,816 KB	00:01:50
 Sample_AVI	14/4/2558 14:06	GOM Media file(.avi)	82,954 KB	
 Sample_MP3	1/2/2554 16:56	KMP - MPEG Layer3 Audio File	5,390 KB	00:02:50
 Sample_MP4	14/3/2558 12:22	GOM Media file(.mp4)	37,681 KB	00:01:29
 Sample_WMV	14/7/2552 12:32	GOM Media file(.wmv)	25,631 KB	00:00:30
 thisfile	6/3/2556 14:53	GOM Media file(.3gp)	7,869 KB	00:13:01
 video.name	5/1/2557 20:45	GOM Media file(.mp4)	9,531 KB	00:05:35

Figure 3.10 Multimedia file samples in the directory

3.3.1 The Split Code

The split code is separated into 2 main modules called SplitHead and SplitContent. The two modules will perform their tasks as the name suggests. The split code will be activated when a new media file enters the system.

SplitHead

The SplitHead module is responsible to cut first 4-byte length from the media file and name it to be the first part (by adding '1_' as a prefix). The rest of the content uses the file's original name and will get split consecutively by the next module, so called SplitContent. Figure 3.11 shows Pseudocode of SplitHead.

SplitHead

Initial status: Targeted video file in a specific directory

```

Get String Video.name
StreamSize <-0
SplitSize <-4                (For 4-bytes file part)
While File.InputStream is not equal to -1 (End of the file)
  If SpliSize is not equal to StreamSize Then
    Do
      `Read data from a target video file
      `Write a new file up to SplitSize while incrementing
      StreamSize
      `Save the file and add prefix '1_' to Video.name
    Else
      `Read data from a target video file
      `Write data until the end of the file
      `Save the file as the same name Video.name
    End If
  End While
End While

```

Figure 3.11 Pseudocode of SplitHead module

The media file 'video.name.mp4' has been split into two parts. The first part is the header and the second is the media content file. Figure 3.12 shows the sample files status in the directory.



Name	Date	Type	Size	Length
 01_Video.name	22/4/2558 2:07	GOM Media file(.mp4)	1 KB	
 Video.name	22/4/2558 2:07	GOM Media file(.mp4)	9,531 KB	

Figure 3.12 Sample file gets split into header and content file

Process log of the process has been recorded to archive the activities when SplitHead module was executed. Log of the process SplitHead is shown in Figure 3.13.

Process Log data

```
Directory Created -> C:\Documents\Videos_Split\Video.name  
File Created Location:  
C:\Documents\Videos_Split\Video.name\01_Video.name.mp4  
File Created Location:  
C:\Documents\Videos_Split\Video.name\Video.name.mp4
```

Figure 3.13 SplitHead Log Process

SplitContent

SplitContent is coded similarly to SplitHead. This module will locate the media file content to undergo dissection. It will equally divide the rest of the file content into file parts of equal size, which will be in 2^n fix size blocks. At the end, it will identify the last part and record its name and size to the central server. Then, it will resize the last part to be the same as the rest of the parts. This insertion part will be removed later during the joining process when the file is requested to be played. Pseudocode of SplitContent will be represented in Figure 3.14.

SplitContent

Initial status: 2 file parts in a specific directory; one is 4-byte header file namely 1_video.name; the other one contains the rest of video file content having same original name video.name

```

Get String Video.name

StreamSize <-0

SplitSize <-512 * 1024      (For each part to be 512 Kb)

File.Count<- 2              (Started with 2 files)

While File.InputStream is not equal to -1 (End of the file)
If SplitSize is not equal to StreamSize Then
Do
    'Read data from the original file
    'Write a new file up to SplitSize while incrementing StreamSize
    'Save the file and add prefix 'File.Count_' to Video.name
    'Increment File.Count
'Reset StreamSize to zero
Else
    'Read data from a target video file
    'Write data until the end of the file
    'Save the file and add prefix 'File.Count_' to Video.name
End If
End While

Delete Video.name          (Delete original content file)

Get String Lastpart.name

Get Size Lastpart.size

    'Store file name and size of the last file part

```

Figure 3.14 Pseudocode of SplitContent Module

While having the first part of the file (01_Video.name) remains the same, the media content chunk got further split by SplitContent module. This process created 20 file parts in total each will have the prefix of number series. Note that the last part was not originally 512 KB, but it was filled to be 512 KB. Figure 3.15 shows the sample file parts' status in the directory after the split.





















Name	Date	Type	Size	Length
 01_Video.name	22/4/2558 2:07	GOM Media file(.mp4)	1 KB	
 02_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 03_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 04_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 05_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 06_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 07_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 08_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 09_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 10_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 11_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 12_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 13_Video.name	22/4/2558 2:11	GOM Media file(.mp4)	512 KB	
 14_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 15_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 16_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 17_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 18_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 19_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	
 20_Video.name	22/4/2558 2:12	GOM Media file(.mp4)	512 KB	

Figure 3.15 A Sample file gets split into equal parts.

Like the previous module, log data has been recorded to keep track during the execution of SplitContent module. The last part size and name was recorded to be used when the joining time. Process log data of SplitContent will be shown in Figure 3.16.

Process Log data

```
File Created Location:
C:\Documents\Videos_Split\Video.name/02_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/03_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/04_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/05_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/06_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/07_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/08_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/09_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/10_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/11_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/12_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/13_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/14_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/15_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/16_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/17_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/18_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/19_Video.name.mp4
File Created Location:
C:\Documents\Videos_Split\Video.name/20_Video.name.mp4
20 parts are created . . .
The size of the last part is 322458 bytes
20_Video.name.mp4 is resized
```

Figure 3.16 SplitContent Log Process

3.3.2 The Join Code

When all the file parts are assembled including media header, the join code will append all parts together using JoinAll module. Firstly, JoinAll will identify the last part using name and its original size obtained from the central server and resize it back. This process must be performed precisely meaning the name and size value obtained must be accurate in order to obtain the perfectly joined media. JoinAll will be explained in Figure 3.17.

JoinAll

Initial status: All file parts have been assembled and are ready to be appended.

Get String Lastpart.name

Get Size Lastpart.size

`Identify the file part and resize it to its original size

While All file parts still exist in the directory

`Read data from file parts starting from 1_Video.name

`Write data to a new file

End While

`Replace '1_' prefix with 'Joined_'

Figure 3.17 Pseudocode explaining JoinAll

The last file part was located with name. The insertion chunk from the last file part gets removed to its original size before the joining process. All file parts have been integrated back. The prefix 'Joined_' has been added to make it easier to distinguish. Figure 3.18 shows the sample files status in the directory.


Name	Date	Type	Size	Length
 Joined_Video.name	22/4/2558 2:13	GOM Media file(...	9,531 KB	00:05:35

Figure 3.18 Sample file parts are joined back.

Process log data has been recorded during the execution of JoinAll module. Figure 3.19 shows the log process of JoinAll module.

Process Log data

```
The last part original size is 322458 bytes
The last file part was 20
20_Video.name.mp4 is resized back
Total files to be joined: 20
Join file created with name -> Joined_Video.name.mp4
Created Directory -> C:\Documents\Videos_Join\Video.name
Reading the file -> 01_Video.name.mp4
Reading the file -> 02_Video.name.mp4
Reading the file -> 03_Video.name.mp4
Reading the file -> 04_Video.name.mp4
Reading the file -> 05_Video.name.mp4
Reading the file -> 06_Video.name.mp4
Reading the file -> 07_Video.name.mp4
Reading the file -> 08_Video.name.mp4
Reading the file -> 09_Video.name.mp4
Reading the file -> 10_Video.name.mp4
Reading the file -> 11_Video.name.mp4
Reading the file -> 12_Video.name.mp4
Reading the file -> 13_Video.name.mp4
Reading the file -> 14_Video.name.mp4
Reading the file -> 15_Video.name.mp4
Reading the file -> 16_Video.name.mp4
Reading the file -> 17_Video.name.mp4
Reading the file -> 18_Video.name.mp4
Reading the file -> 19_Video.name.mp4
Reading the file -> 20_Video.name.mp4
Joined file saved at ->
C:\Documents\Videos_Join\Video.name\Joined_Video.name.mp4
```

Figure 3.19 JoinAll process log data

3.4 Calculation on the aggregate size of media files part after the split

The dissection process of media files using the above methodology does create some changes in an aggregate size of file parts. Splitting a file into an equally fixed-size

block will produce a remainder block almost all the time. Hence, deciding a proper 2^n size determines the increase in file parts aggregate size. It means the smaller an insertion part to the last file part; there will be less or no increase in the aggregate file size at all. The below derivation will show an increase in the file size. Definitions of variables used in the derivation have been listed in Table 3.1.

Table 3.1: Summary of variables definitions

Variable	Definition
X	Size of a sample video in KB
2^n	Fixed Split Size in KB
$TRUNC\left(\frac{X}{2^n}\right)$	Truncate the value of $\frac{\text{SampleSize}}{2^n \text{Fixed size block}}$ to obtain only a whole number
i	Size of an insertion part in KB
R	Size of a remainder part in KB

Assume a sample video of size X KB and the file will undergo a split of size 2^n KB for each part.

Thus, there will be $TRUNC\left(\frac{X}{2^n}\right)$ of 2^n size block and one remainder block of size R KB which makes total file parts $(TRUNC\left(\frac{X}{2^n}\right) + 1)$ parts or $ROUNDUP\left(\frac{X}{2^n}\right)$.KB

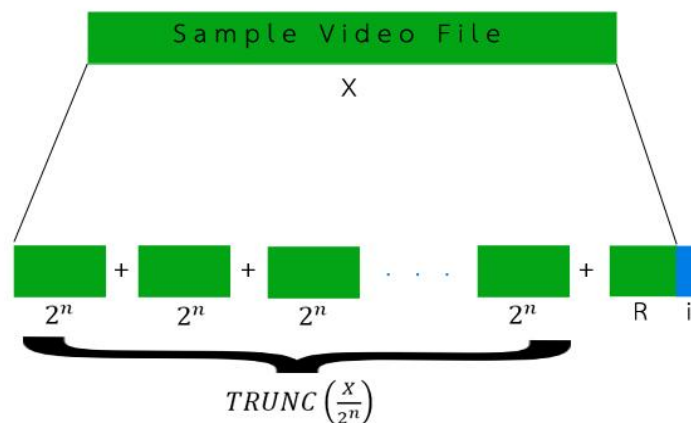


Figure 3.20 Explanation of how net size will get increased after the split.

Since the whole sample file size is equivalent to all 2^n fixed size block and 1 remainder size block, the master equation can be written as;

$$\text{SampleSize} = \text{FixedSplitSize} \cdot (\text{no - of - } 2^n \text{ - parts}) + \text{RemainderSize} \cdot 1$$

By putting related variables, the equation becomes;

$$X = 2^n \cdot \text{TRUNC} \left(\frac{X}{2^n} \right) + R$$

To make the last part to be the same size as the others, an insertion part of size i KB is added to it. The size of 2^n can be written as;

$$2^n = R + i, \text{ or, } R = 2^n - i$$

By substituting this into the main equation,

$$X = 2^n \cdot \text{TRUNC} \left(\frac{X}{2^n} \right) + (2^n - i)$$

Rearrange it and the following equation provides an increasing size of the remaining file parts as follows;

$$i = 2^n \cdot \text{TRUNC} \left(\frac{X}{2^n} \right) + 2^n - X$$

This final equation will be used to calculate the increase in aggregate size of the file. The results, graphs and discussion will be shown in Chapter 4.

CHAPTER 4

EVALUATIONS AND EXPERIMENTAL RESULTS

This chapter will contain the evaluations of the test results on the sample video files using the methodology from the previous chapter. This chapter is divided into 4 sections; File content verification is in Section 4.1, results on security enhancement in Section 4.2, aggregate size result tables in Section 4.3 and finally transfer speed analysis in Section 4.4.

4.1 File Content verification

The most important thing in splitting a file is to have exactly the same media file after all parts are joined. Joined file parts must possess exactly the same properties compared to its original one. Therefore, this section will elaborate the verification process and the result to show that the joined files will be the same as it originally was.

4.1.1 Using properties windows

Properties like, size, size on disk, length, frame width, frame height, total bitrate, frame rate and audio sample rate are taken into account. If they are all shown exactly the same, then the file retains its original properties. The test results on other files will be shown in the next section. Figure 4.1 and 4.2 show the comparison of video properties after undergoing the split-join process.

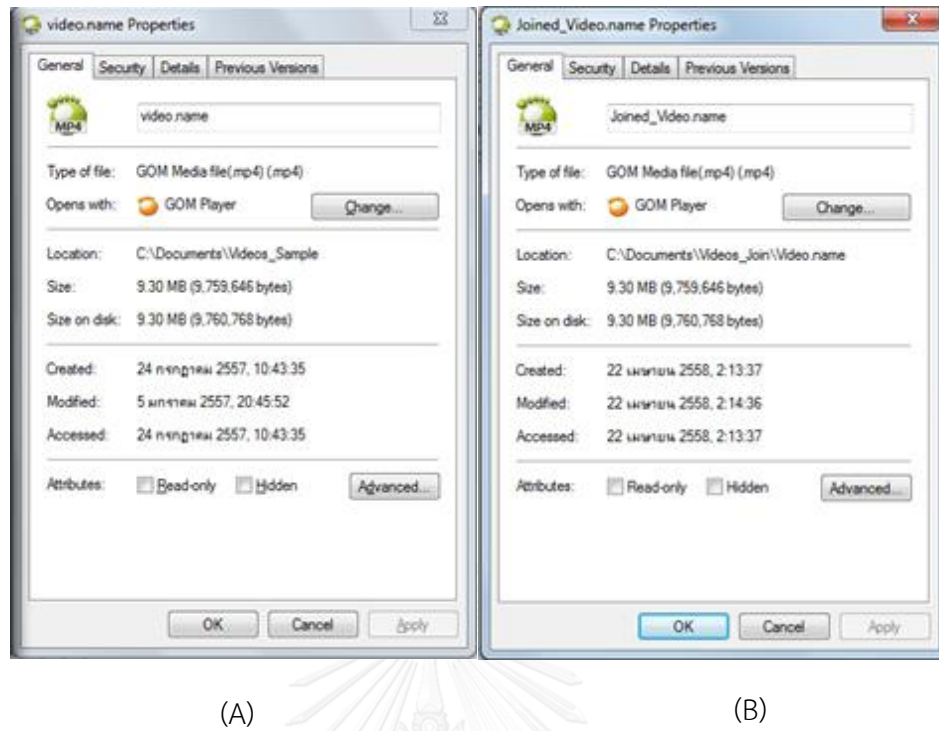


Figure 4.1 (A) Original video properties (B) Joined video properties after split and join process

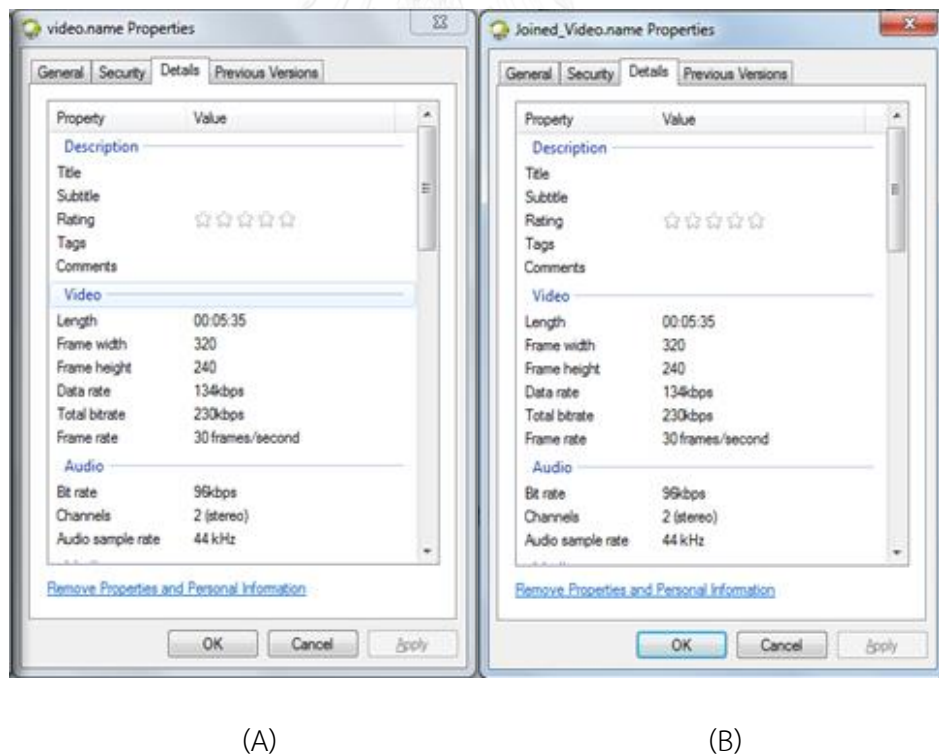


Figure 4.2 (A) Original video details (B) Joined video details after split and join process

4.1.2 Using MP4 Reader

The following comparisons have been processed to verify multimedia file content using MP4 Reader program. Since there are too many lines of video content in hexadecimal values, the lines have been picked randomly to make a comparison between those of original and joined videos. The tables below will show content information in hexadecimal numbers copied from the program's log console.

ftyp part

ftyp part is the first part of video file. The video content in hexadecimal value is shown in Table 4.1. If the two files are not exactly the same, these sets of data will be totally different. 'ftypmp42' shows that the file format is mp4.

Table 4.1: ftyp content in hexadecimal value read by MP4 Reader

File Name	ftyp content in hexadecimal value
Video.name.mp4	00 00 00 18 66 74 79 70 6D 70 34 32 00 00 00 00 ;ftypmp42... 69 73 6F 6D 6D 70 34 32 ; isommp42
Joined_video.name.mp4	00 00 00 18 66 74 79 70 6D 70 34 32 00 00 00 00 ;ftypmp42... 69 73 6F 6D 6D 70 34 32 ; isommp42

ftyp console and raw data happen to be exactly the same. Data of moov and mdat part can further be checked for more content confirmation.

moov part

moov console information of both files are compared. Since there is a lot of lines shown in the console, data lines have randomly been picked from line 5-15. If both files possess different sets of data, then content in both the files will be different. Table 4.2 shows raw data picked from line 5-15.

Table 4.2: moov content in hexadecimal value

File Name	moov content
Video.name.mp4	<pre> 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 ; 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 ; ...@..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 00 03 00 00 00 15 69 6F 64 73 00 00 00 00 ;iods... 10 07 00 4F FF FF 29 15 FF 00 00 C4 C7 74 72 61 ; ...O.).....tra 6B 00 00 00 5C 74 6B 68 64 00 00 00 0F 00 00 00 ; k...\tkhd..... 00 CE C5 AD D0 00 00 00 01 00 00 00 00 00 03 11 ; A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 40 00 00 00 01 40 00 00 00 F0 00 00 00 00 00 ; .@...@..... </pre>
Joined_video.name.mp4	<pre> 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 ; 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 ; ...@..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 00 03 00 00 00 15 69 6F 64 73 00 00 00 00 ;iods... 10 07 00 4F FF FF 29 15 FF 00 00 C4 C7 74 72 61 ; ...O.).....tra 6B 00 00 00 5C 74 6B 68 64 00 00 00 0F 00 00 00 ; k...\tkhd..... 00 CE C5 AD D0 00 00 00 01 00 00 00 00 00 03 11 ; A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 ; 00 40 00 00 00 01 40 00 00 00 F0 00 00 00 00 00 ; .@...@..... </pre>

Table 4.3: mdat part in hexadecimal value

File name	mdat in hexadecimal value
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;
	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ;
	00 00 00 00 30 70 ;0p

ftyp, moov, and mdat data of original and joined files are shown to be the same. Hence, this media file retains its originality after undergoing split-join process.

4.1.3 Using Hash Calculator

File content can undergo cryptographic algorithms and produces irreversible hashes. Hash is like a fingerprint; any strings or computer files can produce a unique hash value. It is the rule that no two files can have the same hash value (although in some rare circumstances they can). Therefore, the file content identity can be checked using the hash calculator. The resulting values have been pulled out from a hash calculator when both original and joined files are loaded into the program. The value is shown in the following table.

Table 4.4: MD5 and SHA1 hash comparison

File Name	MD5	SHA1
Joined_video.name.mp4	8d30a948aef14c36a7f8f5e78ff4a8 09	d4ba529885c2812cd21d326104584123 b571bc03
Video.name.mp4	8d30a948aef14c36a7f8f5e78ff4a8 09	d4ba529885c2812cd21d326104584123 b571bc03

More test results of content verification on different multimedia file samples are shown in Section 4.2.2.

4.1.4 Content verification test results

By running split and join code on different video clip, it is highly expected to have no change in video resolution, audio quality, size of video clip, frame rate and other video properties. Tests are run on the sample video files to compare the properties of the original and those of split-join video. Questionnaire [appendix] is given to people based on inquiry regarding quality of video resolution and audio quality. Questions are inquired from users of different multimedia background based on their interest in Digital Right Management also. The test users' evaluation will be discussed in Section 4.5. Many multimedia files of different formats and sizes have been tested to ensure that the content is perfectly the same as what it was. The below resulting tables show that splitting and joining multimedia files does not intervene any physical changes in the files themselves.

Video Sample no. 1

Table 4.5: MP4 sample file information after undergoing split-join process

	Original Video	After Split and Join with code
Name	Sample_MP4	Joined_Sample_MP4
Format	mp4	mp4
Duration	89 seconds	89 seconds
Size	38,584,350 bytes	38,584,350 bytes
Other video properties	Same	
MP4 Reader Raw Data	Same	
MD5 and SHA1 checksum	Same	

Video Sample no. 2

Table 4.6: AVI sample file information after undergoing split-join process

	Original Video	After Split and Join with code
Name	Sample_AVI	Joined_Sample_AVI
Format	AVI	AVI
Duration	263 seconds	263 seconds
Size	84,944,154 bytes	84,944,154 bytes
Other video properties	Same	
MP4 Reader Raw Data	Same	
MD5 and SHA1 checksum	Same	

Video Sample no. 3

Table 4.7: 3GP sample file information after undergoing split-join process

	Original Video	After Split and Join with code
Name	Sample_3GP	Joined_Sample_3GP
Format	3gp	3gp
Duration	110 seconds	110 seconds
Size	13,123,351 bytes	13,123,351 bytes
Other video properties	Same	
MP4 Reader Raw Data	Same	
MD5 and SHA1 checksum	Same	

Video Sample no. 4

Table 4.8: WMV sample file information after undergoing split-join process

	Original Video	After Split and Join with code
Name	Sample_WMV	Joined_Sample_WMV
Format	Wmv	wmv
Duration	30 seconds	30 seconds
Size	26,246,026 bytes	26,246,026 bytes
Other video properties	Same	
MP4 Reader Raw Data	Same	
MD5 and SHA1 checksum	Same	

Video Sample no. 5

Table 4.9: MP3 sample file information after undergoing split-join process

	Original Video	After Split and Join with code
Name	Sample_MP3	Joined_Sample_MP3
Format	mp3	mp3
Duration	170 seconds	170 seconds
Size	5,519,241 bytes	5,519,241 bytes
Other video properties	Same	
MP4 Reader Raw Data	Same	
MD5 and SHA1 checksum	Same	

All sample media files have been verified to be the same. The joined file possess same content as its original one.

4.2 Results on security enhancement

File parts split by code and a third party program are tested independently using variety of media players. The details of video sample are shown in the Table 4.10 and the results on security enhancement test are shown in Table 4.11. The results determine the security enhancement of the MSSE.

Video sample details

Table 4.10: Sample video information on security enhancement test

Name:	Original_Sample
Format:	mp4
Size:	38,584,350 bytes
Size on disk:	38,588,416 bytes
Duration:	89 seconds

Table 4.11: Results on security enhancement test

Media Player	Using Third Party Program Split	Using Code Split
Window Media Player	Playable	Not Playable
GOM Player	Playable	Not Playable
VLC Media Player	Playable	Not Playable
KM Player	Playable	Not Playable
QuickTime Player	Playable	Not Playable
Default Android Video Player	Playable	Not Playable
MX Player (Android app)	Playable	Not Playable
Video Player Ultimate (HD)	Playable	Not Playable

The following figure show you feedback after the media file part is set to be played in VLC media player. It shows an error window and the file is not playable.

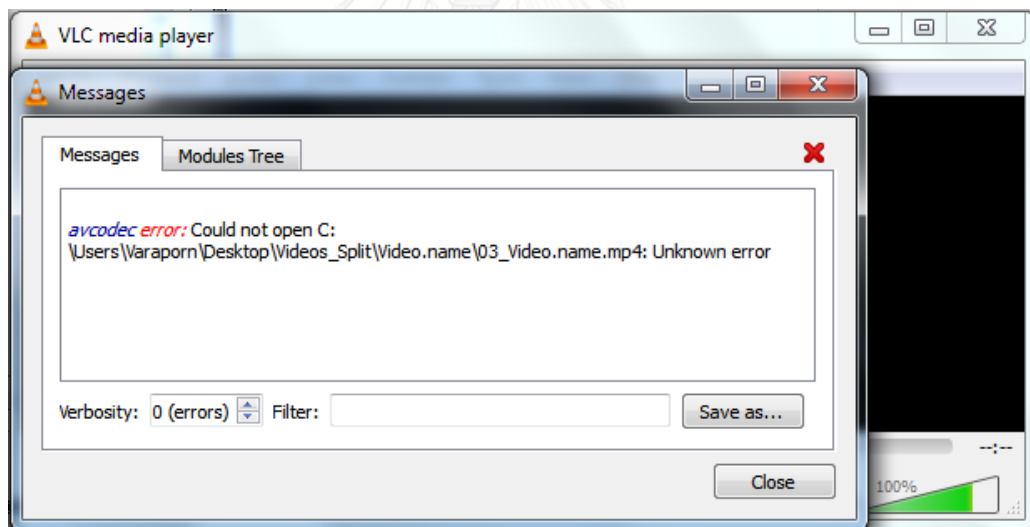


Figure 4.3 MP4 file security enhancement result

From the results, the file parts are not at all playable by common end users using well-known video players available. Even VLC player that claims to be a universal video player is not able to play those parts. The following figures show more result on other file formats after implementing of the MSSE. The message window shows an error and warning messages after a media file is put into a VLC Media Player.

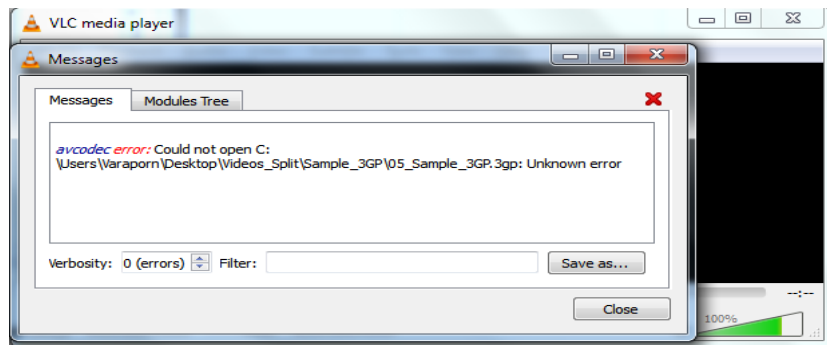


Figure 4.4 3GP file security enhancement result

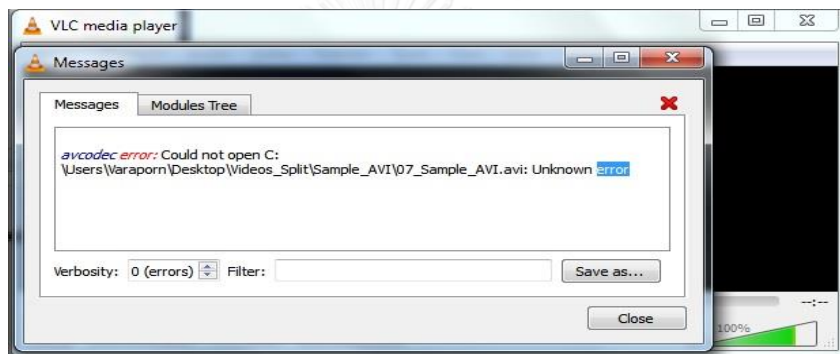


Figure 4.5 AVI file security enhancement result

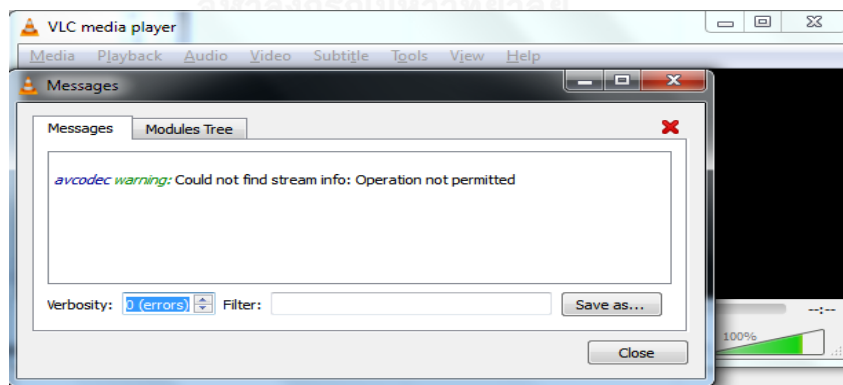


Figure 4.6 WMV file security enhancement result

As a result, the implementation of the MSSE can also be applied to other file format samples.

4.3 Aggregate size result tables

Due to the insertion part discussed in the previous chapter, the aggregate size of file parts has been increased. The following tables will show how much does the net size of file parts increase after an insertion part is added to the last part. Various sample files are tested and results have been statistically graphed. Original sample size and its aggregate size after the split should remain the almost the same. The increase in size can be calculated using the formula that has been discussed in the previous chapter. Finally the last graph will show aggregate size patterns of all sample file parts.



Table 4.12: MP3 sample file net size increase

	2^n	$TRUNC\left(\frac{X}{2^n}\right)$	$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$	
S.no.	Size of one file part (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	4096	2	8192	2802
2	2048	3	6144	754
3	1024	6	6144	754
4	512	11	5632	242
5	256	22	5632	242
6	128	43	5504	114
7	64	85	5440	50
8	32	169	5408	18
9	16	337	5392	2
10	8	674	5392	2
11	4	1348	5392	2
12	2	2695	5390	0

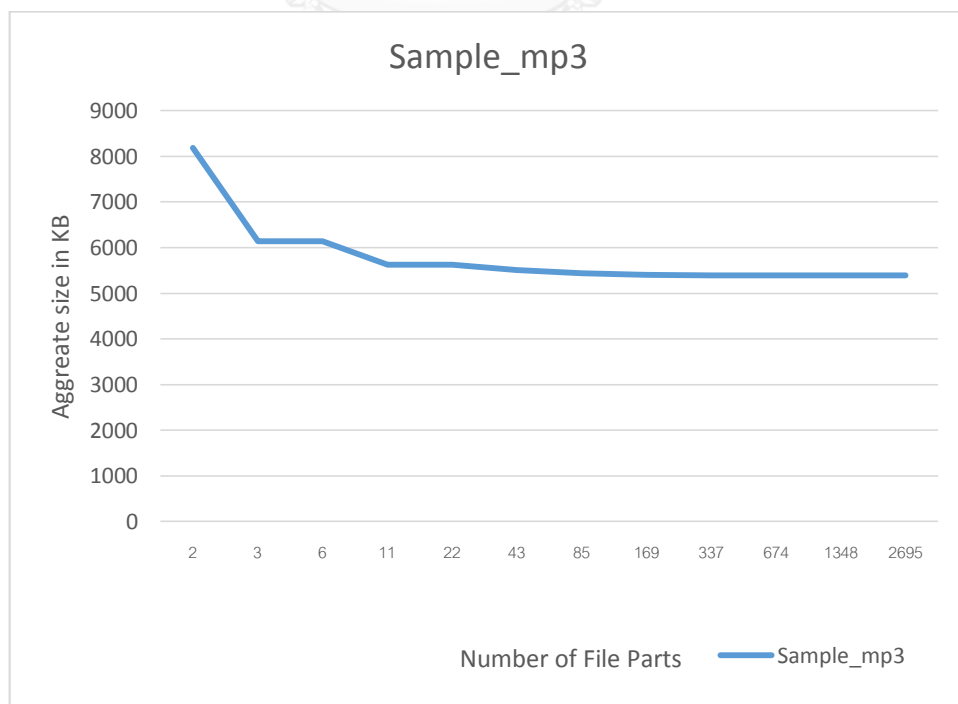


Figure 4.7 MP3 sample file net size increase graph

Table 4.13: 3GP sample file net size increase

	2^n	$TRUNC\left(\frac{X}{2^n}\right)$		$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$
S.no.	Size of one file part (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	4096	2	8192	323
2	2048	4	8192	323
3	1024	8	8192	323
4	512	16	8192	323
5	256	31	7936	67
6	128	62	7936	67
7	64	123	7872	3
8	32	246	7872	3
9	16	492	7872	3
10	8	984	7872	3
11	4	1968	7872	3
12	2	3935	7870	1

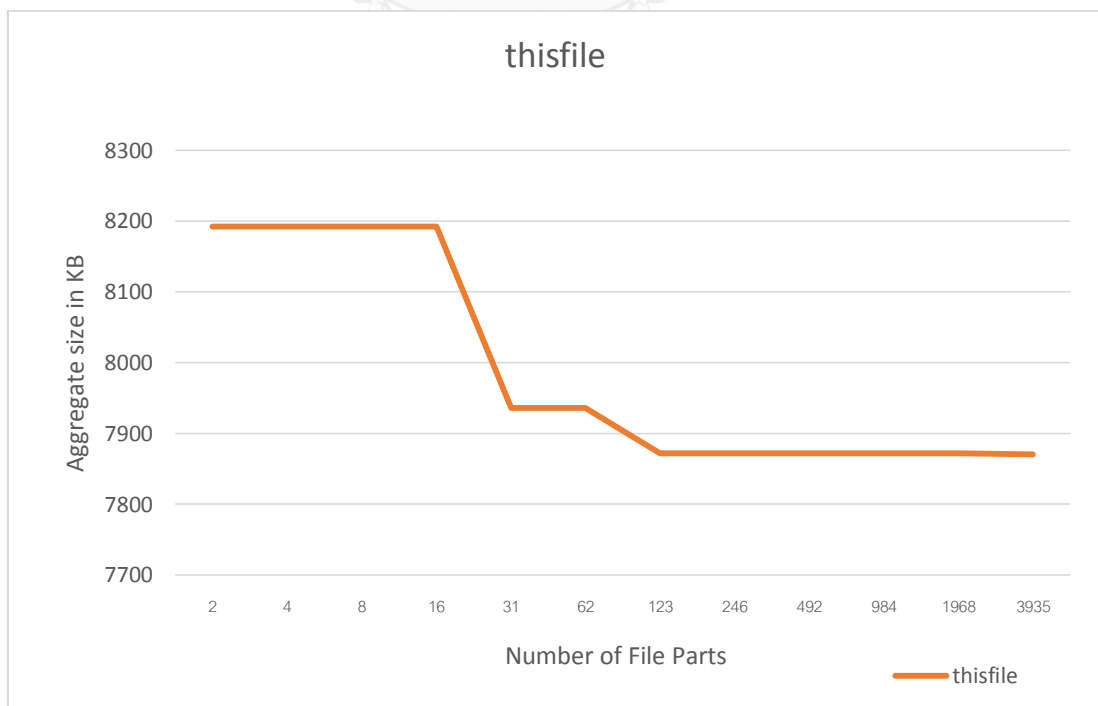


Figure 4.8 3GP sample file net size increase graph

Table 4.14: MP4 sample file net size increase

	2^n	$TRUNC\left(\frac{X}{2^n}\right)$	$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$	
S.no.	Size of one file part (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	8192	2	16384	6853
2	4096	3	12288	2757
3	2048	5	10240	709
4	1024	10	10240	709
5	512	19	9728	197
6	256	38	9728	197
7	128	75	9600	69
8	64	149	9536	5
9	32	298	9536	5
10	16	596	9536	5
11	8	1192	9536	5
12	4	2383	9532	1
13	2	4766	9532	1

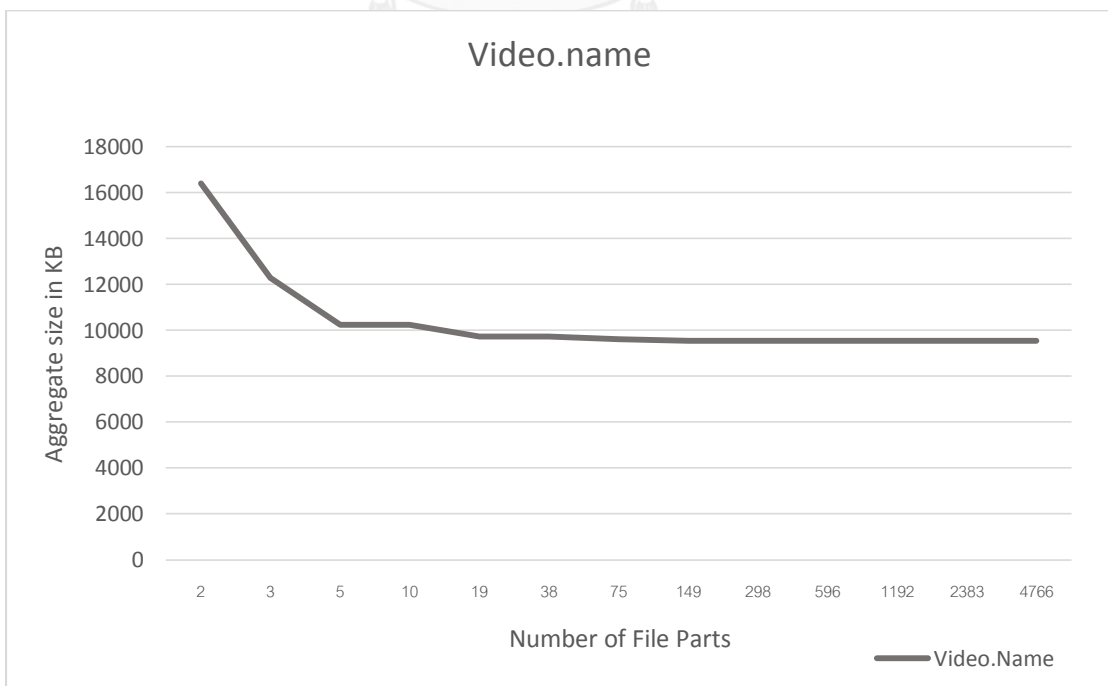


Figure 4.9 MP4 sample file net size increase graph

Table 4.15: 3GP sample file net size increase

X		$TRUNC\left(\frac{X}{2^n}\right)$	$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$	
S.no.	Sample Video file size (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	12816	2	16384	3568
2	12816	4	16384	3568
3	12816	7	14336	1520
4	12816	13	13312	496
5	12816	26	13312	496
6	12816	51	13056	240
7	12816	101	12928	112
8	12816	201	12864	48
9	12816	401	12832	16
10	12816	801	12816	0
11	12816	1602	12816	0
12	12816	3204	12816	0
13	12816	6408	12816	0

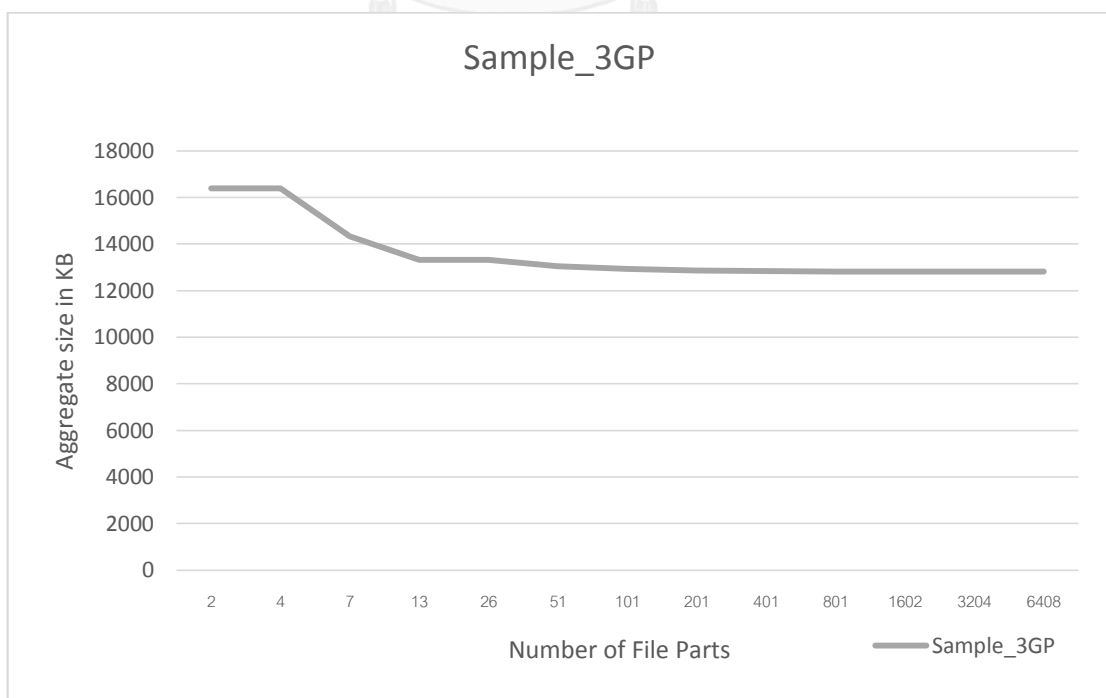


Figure 4.10 3GP sample file net size increase graph

Table 4.16: WMV sample file net size increase

	2^n	$TRUNC\left(\frac{X}{2^n}\right)$	$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$	
S.no.	Size of one file part (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	16384	2	32768	7137
2	8192	4	32768	7137
3	4096	7	28672	3041
4	2048	13	26624	993
5	1024	26	26624	993
6	512	51	26112	481
7	256	101	25856	225
8	128	201	25728	97
9	64	401	25664	33
10	32	801	25632	1
11	16	1602	25632	1
12	8	3204	25632	1
13	4	6408	25632	1
14	2	12816	25632	1

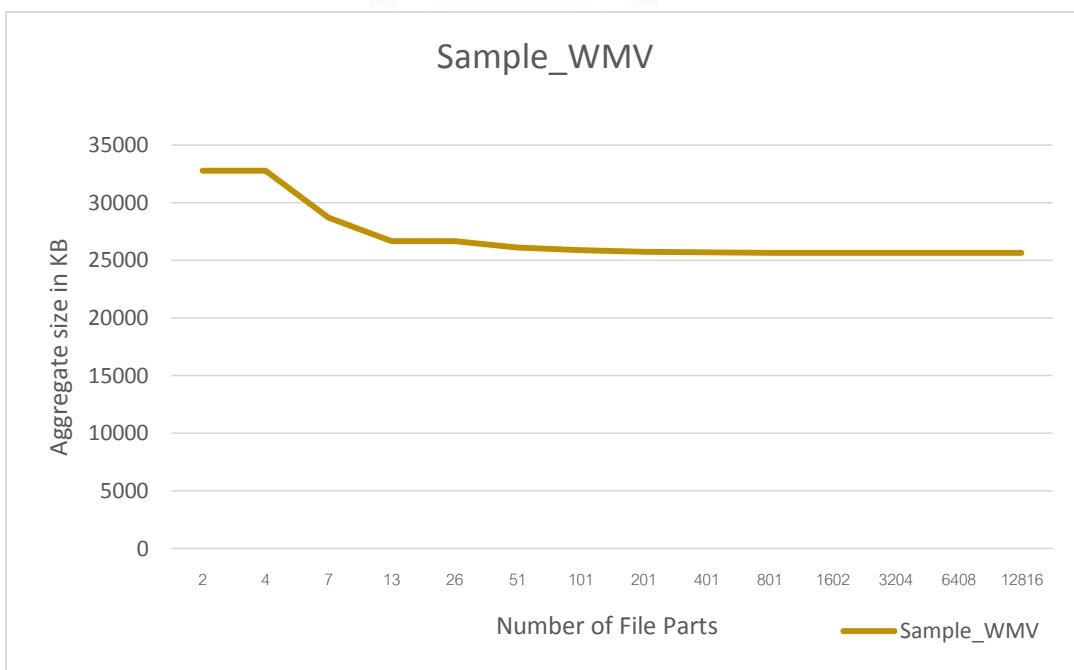


Figure 4.11 WMV sample file net size increase graph

Table 4.17: AVI sample file net size increase

	2^n	$TRUNC\left(\frac{X}{2^n}\right)$	$i = 2^n \cdot TRUNC\left(\frac{X}{2^n}\right) + 2^n - X$	
S.no.	Size of one file part (KB)	Number of file parts After the Split	Net size (KB)	Net Size Increase (KB)
1	65536	2	131072	48119
2	32768	3	98304	15351
3	16384	6	98304	15351
4	8192	11	90112	7159
5	4096	21	86016	3063
6	2048	41	83968	1015
7	1024	82	83968	1015
8	512	163	83456	503
9	256	325	83200	247
10	128	649	83072	119
11	64	1297	83008	55
12	32	2593	82976	23
13	16	5185	82960	7
14	8	10370	82960	7
15	4	20739	82956	3
16	2	41477	82954	1

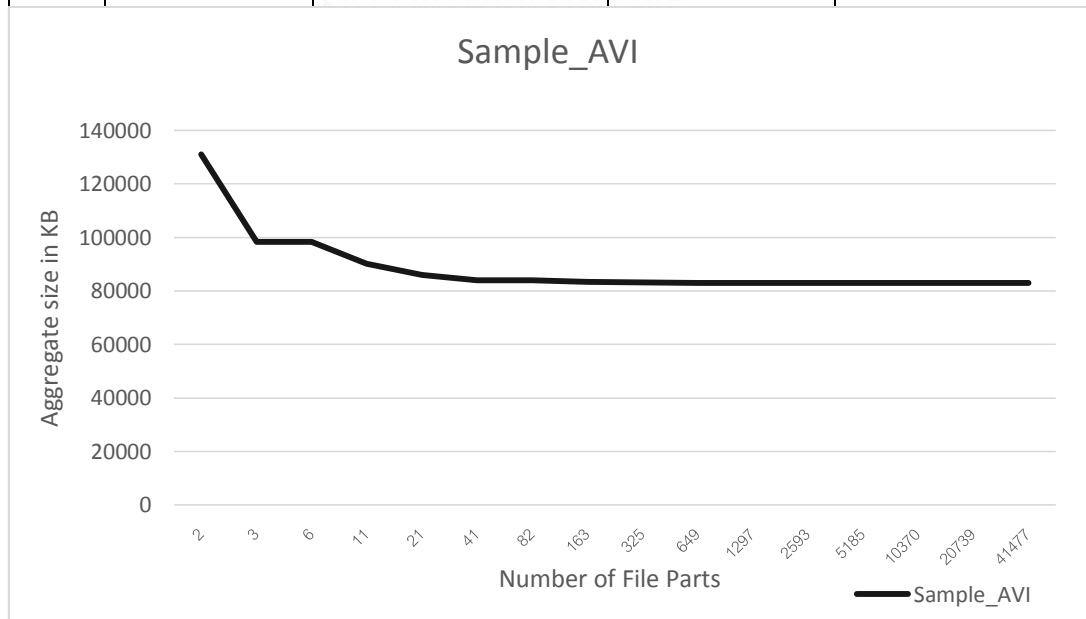


Figure 4.12: AVI sample file net size increase graph

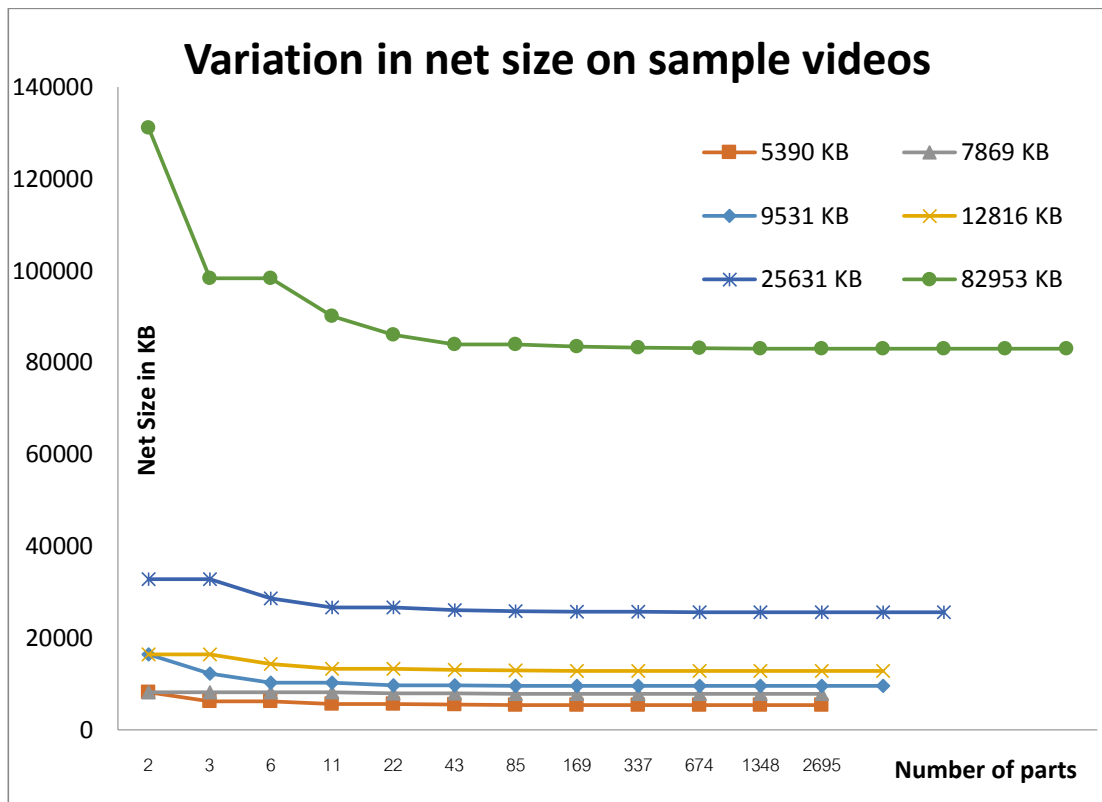


Figure 4.13 Variation in net size on sample videos graph

The graph shows that the more file parts in a mobile cloud system, the value of file aggregate size will tend towards its original size.

4.4 Transfer Speed Analysis

This section determines the result of experiments regarding time taken to retrieve media file parts.

4.4.1 Multimedia file transfer from seeders to leecher

This section is to show that when the MSSE was applied, it does not increase any files' mean retrieval time. A similar environment to that of [1] has been simulated to obtain retrieval timing pattern which will be discussed later on in this section.

Theoretically, when there is no seeder in mobile cloud, leecher and seeder will send the file in one-to-one pattern. Media requesting time (T_{req}) is negligible, so majorly T_{send} is considered. Figure 4.14 shows a file transfer process when there is only one seeder in the system.

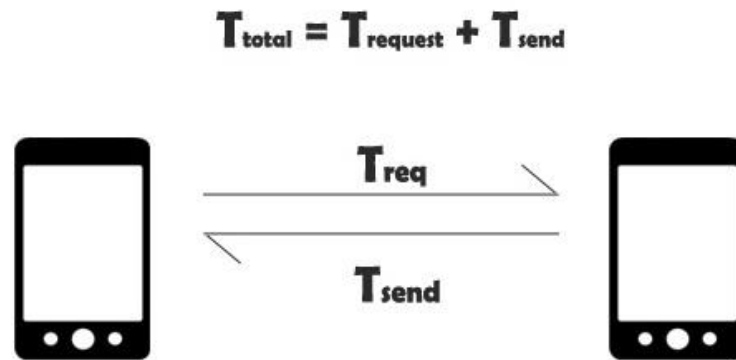


Figure 4.14 A leecher is requesting for media when there is no seeder

When there are more seeders, the system become as in [1], and the aggregate file size containing in each seeder reduces when there are more file parts. By considering the time taken for Media requesting time (T_{req}), time to transfer header of the media file (T_{head}) and (S_{req}) are negligible; since it occurred almost instantaneously. Let $t_1, t_2, t_3 \dots t_n$, be time taken for file parts from each seeder to reach a media requester. This process of file parts transferring from seeders happens concurrently. The time taken for all media file parts to reach a media requester is measured to be $\text{MAX} \{ t_1, t_2, t_3 \dots t_n \}$. Figure 4.15 shows the process of requesting media file from a mobile cloud.

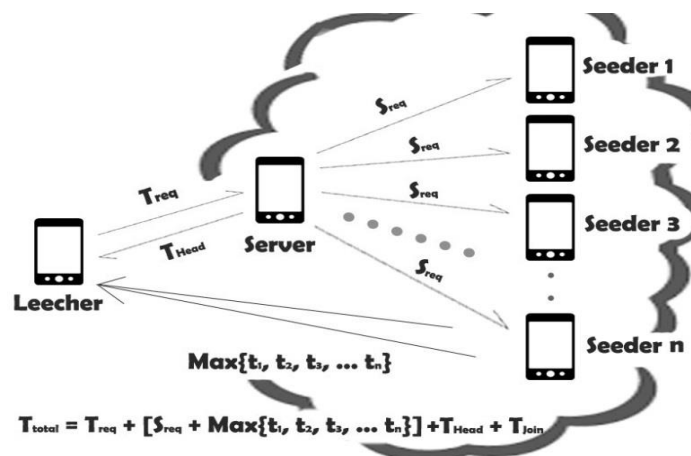


Figure 4.15 A conceptual model of multimedia file retrieval via mobile cloud system

4.4.2 The experimental simulation and results

This simulation majorly emphasis in the file transfer part from seeders in mobile cloud to the leecher. The simulation has been carried out to measure the transfer time in multimedia retrieval. A sample video file is split into 3, 10, 19, 75 and 149 parts. These part numbers have no pattern since the split has been done based on fixed-size of file parts. In each test case, all parts are divided equally amongst three seeders. If the part number is not a multiple of 3, the first seeder takes up the remainder based on the Round Robin concept.

The experiment has been carried out using 3 laptops and 1 computer. All laptops are connected into same local wireless network. The sending module in the seeding system has been coded in JAVA to transfer multiple files from a targeted directory to the destination. The receiving module in the receiving system uses multithreading technique to perform parallel file transfer. Assuming all systems are within the same room and are connected to the same wireless local area network.

At present, a smartphone possesses high-spec properties that its hardware performance is no less efficient than a personal computer or a typical laptop. CPU, RAM, Wifi and 3G specification are the major factors that affect file transfer rate. High-end smartphones can even be compared side by side with typical laptops. In this experiment, computers and laptops have been used. Since the size of file parts transferred during this process are small, the transfer time of small fix-sized file parts do not make any difference in laptop and in smartphone. Even typical smartphones in a medium price range possess a respectable specification that transferring small files should not be a problem. Table 4.18 shows specification of typical laptop and smartphone in a various price range.

Table 4.18: Typical smartphone and laptop specification comparison

Device	CPU	RAM	WLAN
Laptop			
Macbook Pro 2015	2.7GHz dual-core Intel Core i5	8GB	Wi-Fi IEEE 802.11 a/b/g/n
Dell Inspiron 5420	Intel Core i3 2.4 GHz	4GB	Wi-Fi IEEE 802.11 b/g/n
Acer Aspire ES1	Intel Pentium N3540	2GB	Wi-Fi IEEE 802.11 a/b/g/n
Smartphone			
Samsung Galaxy Note 4	Quad-core 2.7 GHz Krait 450	3 GB	Wi-Fi IEEE 802.11 a/b/g/n/ac
Samsung Galaxy S3	Quad-core 1.4 GHz Cortex-A9	1 GB	Wi-Fi IEEE 802.11 a/b/g/n
Apple iPad 2 Wifi + 3G	Dual-core 1 GHz Cortex-A9	512 MB	Wi-Fi IEEE 802.11 a/b/g/n

Assuming that all three seeders contain all necessary file parts. Also assuming that each client contains approximately equal numbers of file parts. After the sending module in each seeder is executed, the seeders await for the receiving client to connect. It can be said to be in a ready mode. Once the receiving module in the receiving client is executed, it connects to all seeders as shown in Figure 4.16.

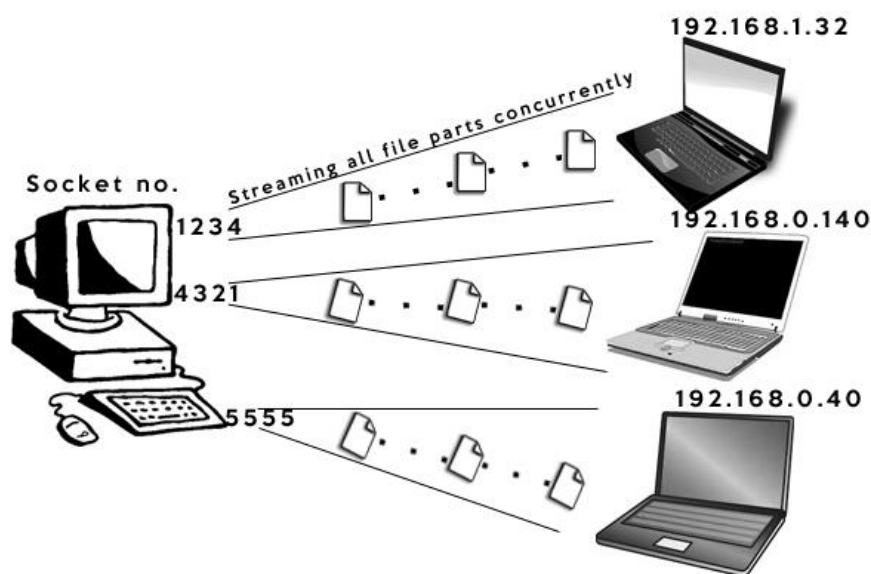


Figure 4.16 Experimental simulation model

The leecher locates all three seeders by IP address and connects to them via independent sockets. All file parts in the target directory are then transferred to it concurrently. The timing has been measured in nanosecond and then been converted into second for the record. Data has been collected as shown in Table 4.19. Content in Table 4.19 will be explained thereafter.

Table 4.19: Multimedia retrieval timing results

No. of file parts in each client	Total file parts sent concurrently in this process	Size of each file part in KB	tClient1	tClient2	tClient3	tClient
1+0+0	1	9523	13.2	13.45	13.68	13.44
1+1+1	3	4096	8.84	9.5	8.74	9.027
4+3+3	10	1024	7.59	7.2	7.71	7.50
7+6+6	19	512	6.59	6.78	6.54	6.64
25+25+25	75	128	6.04	6.13	5.98	6.05
50+50+49	149	64	5.48	5.66	5.74	5.63

Since there are three seeders in this experiment, the number of file parts in each client is represented by $X + Y + Z$ where X , Y and Z are the number of file parts in the seeder 1, 2 and 3 respectively. The sum makes up the total file parts sent concurrently in this process. For the first case, only the first seeder contains the media file to be transferred in the process.

According to typical statistical theory, the mean of observation value is the best representative data. So the experiment has been repeated 3 times for each case and the average time has been calculated. tClient1 signifies the time taken when the leecher have received all file parts from all three seeders in the first test. tClient2 and tClient3 are the timings for 2nd and 3rd try of the process taken to create mean value of tClient, which is the average time taken to retrieve multimedia file parts from all seeders.

Figure 4.17 shows the relationship between the time measured at the receiving system and total number of file parts transferred in the process.

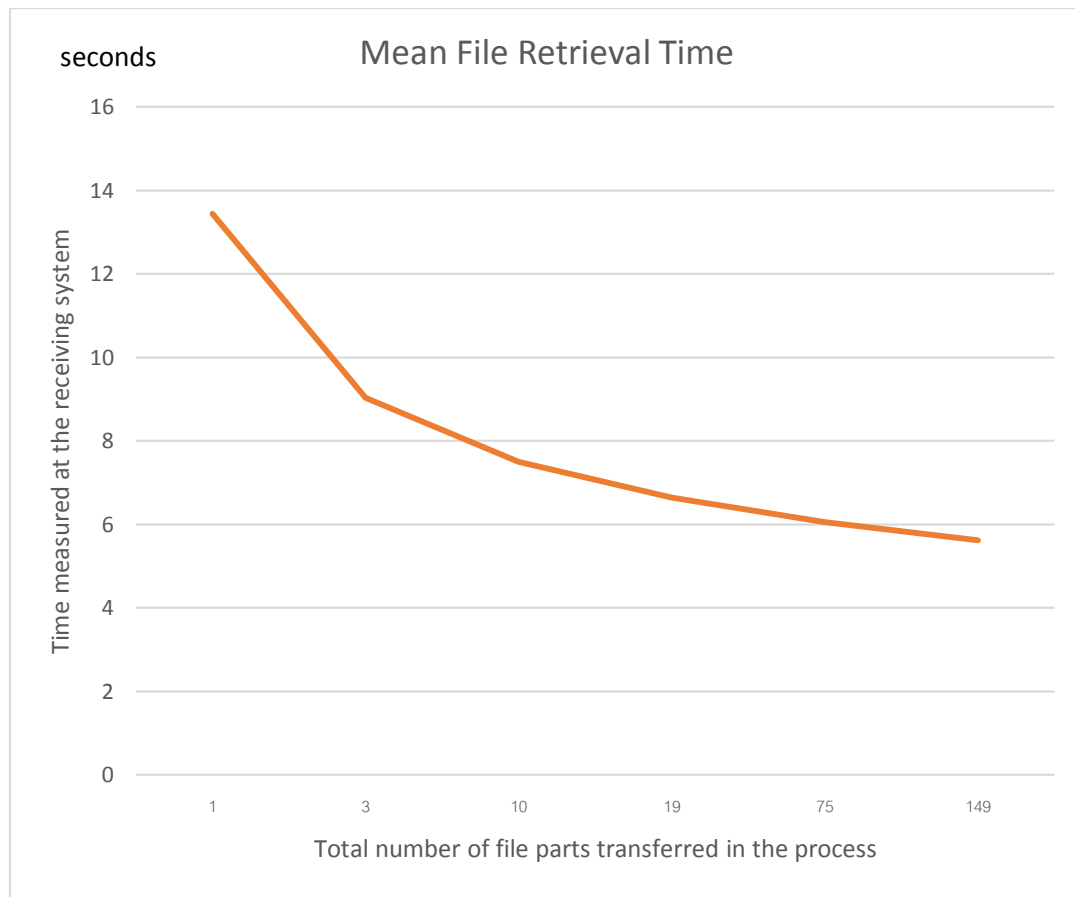


Figure 4.17 Mean File Retrieval Time graph

It is apparent that with the increase in the number of file parts, the time to retrieve file parts from three seeders has been reduced significantly.

4.5 User Satisfaction Evaluation

This experiment was evaluated by 15 volunteers; 8 males and 7 females. The main objective of this experiment is to obtain feedbacks from the users after experiencing the implementation of security enhancement on multimedia file. Within this group there are 12 volunteers below the age of 20. The rest are in the age of 20-30 years old. All the test users have a daily experience with multimedia file.

4.5.1 Experimental procedures and results

A sample video is split into 9 equal parts by third party program and code. All parts are distributed in 3 different devices of android platform; Samsung S1, S3 and Note 8. Firstly, test users are asked to judge the quality of the original video file by expressing their opinion after comparing original video file with the third-party-joined and code-joined one. By the result, 13 out of 15 had agreed that the code-joined video possesses quality same as that of original video shown in Figure 4.18.

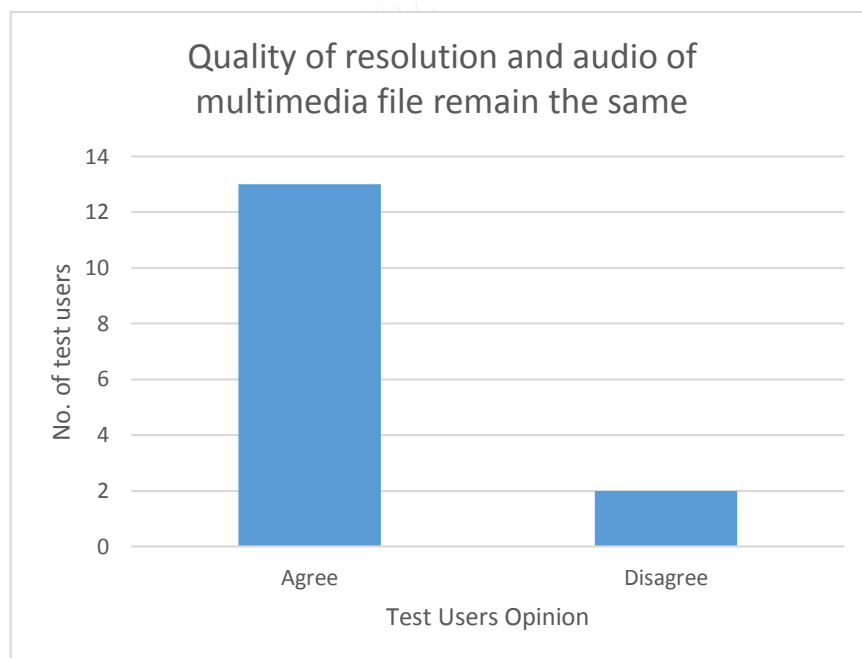


Figure 4.18 Test users' opinion on quality of sample video

Next, test users are asked to view file parts in the devices. They are asked to express their feeling regarding security issue. 11 out of 15 test users had agreed that security of file parts split by code increases shown in Figure 4.19.

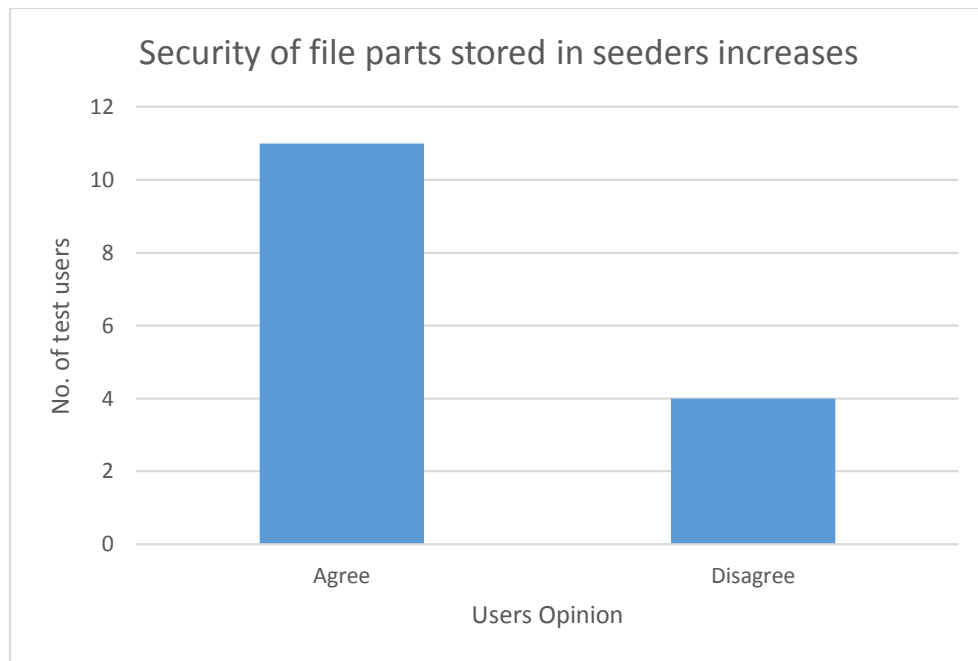


Figure 4.19 Test users' opinion on security enhancement

4.5.2 User evaluation discussion

The following table shows majority of test users' feedback on quality and security after the experiment.

Table 4.20: Users feedback on quality and security after the experiment

Testing Scopes	Majority of test users' feedback
Quality of resolution and audio of multimedia file after split-join process	Remain the same
Security of file parts stored in seeders	The implemented method proposes higher security

Although a video file undergoing split-join process remain the same, as discussed in content verification section, some test users had disagreed that the video lost its originality due to a personal bias.

Majority of the test users agree that the MSSE effectively limits the accessibility of common multimedia users. While the rest of them disagreed that multimedia files are somehow still playable if they are in the hands on computer experts.

Surprisingly, all of the test users do not care about multimedia file security. Test users have an intention of using any free multimedia file whenever they get an opportunity without having to stop and think about an intellectual property issue.



CHAPTER 5

CONCLUSIONS AND DISCUSSIONS

In this chapter, the discussion will be discussed in Section 5.1, limitation of the experiment in Section 5.2 and finally conclusion in Section 5.3.

5.1 Discussion

People always find new methods somehow to violate any source of protection. Some people do it for a challenge; some do it for illegal distribution. Normally, human nature desires to acquire something that is protected. New protection mechanisms should always been invented to keep up with the upcoming illegal way of copyright violation.

Although smartphones provide high mobility, but they lack a lot of other features like, battery life, data storage, CPU performance. That is when a cloud computing technology has shined its utility. With the shifting from cloud computing to mobile cloud computing era, people involve more and more with mobile devices in their daily routines that is one of the reason the matter about mobile cloud system should be emphasized.

Like in conservation of environment, an ethical thinking in people's collective minds is the best approach, but extremely difficult to inculcate. In the Internet, people tend to do illegal activities more than the real world. Some people know that downloading files is wrong but still do it. Many people do not know that it is illegal to violate copyright.

Multimedia file retrieval in mobile cloud does provide a site to quickly obtain such files. It just lacks in security issue regarding an intellectual property violation. This Thesis can be supporting the other existing DRM techniques discussed in Chapter 2 and bring about a more secured file retrieval system.

While other researched on DRM techniques try to lower the quality of multimedia content, this paper proposed the method of making the files not-playable. Table 5.1 shows the overview comparison of the MSSE and other DRM techniques.

Table 5.1: Comparison amongst other DRM techniques

Technique	Type of Multimedia File mentioned in the paper	Result after the implementation of DRM technique
BenPro	MP3	Non-smooth audio content
DoRM Tag	MP3	Not-playable if not authenticated
Audio Watermarking	Audio file	Low quality audio
Proposed method (The MSSE)	All tested video file format (MP4, 3GP, AVI, WMV)	Not-playable by any media player

Hence, the result shows that with the methodology implemented, media file parts have been protected up to some extends and the Digital Right Management techniques have been strengthened.

5.2 Limitation of the experiment

A major limitation of this experiment is that, it protects the media files from mostly common end users, but not expert users in media or hacking field. Those experts may somehow be able to access file content while the other common users may give up when the content isn't playable by any of their video players.

5.3 Conclusion

The methodology, the MSSE that has been proposed in this paper may not be the ultimate solution for the security enhancement in multimedia file entity, but it can surely be an idea that may further get developed or implemented into the other modular mechanisms.

The results have shown that multimedia files can be protected, at least from common multimedia users, up to some extends. This statement has been confirmed using the results based on questionnaire discussed in Section 4.5.1. The implementation of splitting and joining does not alter any file content while providing more security to the aggregate parts. Fixed size block and parallel files streaming also promote quick file transfer. Consequently, the approach can be applied to another project without any affect to efficiency, or can even boost the system's performance up. The result of this research has led to an enhancement in security as well as hyping up the performance.

REFERENCES

- [1] Kaewmahingsa, K., Bhattarakosol, P., Mobile Cloud System: A Solution for Multimedia Retrieval via Mobile Phones, Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCCT2012), Seoul, Korea, 3-5 December 2012, pp. 36-40.
- [2] Ibrahim and I.K., "Mobile multimedia: Bringing users into focus," ELMAR, page 7, 2007
- [3] J.Emery and A.Karmouch, "A multimedia Document Architecture and Rendering Synchronization Schemc" , Procrrding, Second International Conference on Broadband Island.
- [4] C. Wu, B. Li and S. Zhao, "Characterizing Peer-to-Peer Streaming Flows," Selected Areas in Communications, IEEE Journal, Volume: 25, Issue: 9, pages 1612-1626, 200
- [5] Kullapat Theera-angkananon, "BENPro: A Blinding Protection Method in Digital Right Management", pages 533-538
- [6] Karmouch, A. "A Multimedia Information and Communications System: MEDIABASE." Proc. Multimedia Commun. '93 conference, Alberta.

- [7] Jürgen Nützel “Potato System and Signed Media Format - an Alternative Approach to Online Music Business” published in Web Delivering of Music, 2003. 2003 WEDELMUSIC. Proceedings. Third International Conference, Page(s) 23-26.
- [8] Carlos Serrão, António Serra, Miguel Dias, Jaime Delgado, “Protection of MP3 Music Files Using Digital Rights Management and Symmetric Ciphering”, published in Automated Production of Cross Media Content for Multi-Channel Distribution, 2006. AXMEDIS '06. Second International Conference, Page(s) 128 – 135.
- [9] Wang, R. Karmouch, A., “A multimedia file structure for continuous and discrete media”, Published in: Electrical and Computer Engineering, 1993. Canadian Conference, Page(s): 644 - 647 vol.2.
- [10] Willem Jonker and Jean-Paul Linnartz, “Digital rights management in consumer electronics products”, published in Signal Processing Magazine, IEEE (Volume:21 , Issue: 2), pages 82 - 91
- [11] JEFFREY LOTSPIECH, MEMBER, IEEE, STEFAN NUSSER, AND FLORIAN PESTONI, “Anonymous Trust: Digital Rights Management Using Broadcast Encryption”, published in Proceedings of the IEEE (Volume: 92, Issue: 6), pages 898 – 909.
- [12] Jung-Soo, Lee, Ki-Song Yoon, “The System Integration of DRM and Fingerprinting”, published in Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference (Volume: 3), pages 4 pp. – 2183.

[13] Piyali Mandal, Ashish Thakral , Shekhar Verma, “Watermark based digital rights management”, published in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference (Volume: 1), pages74 - 78 Vol. 1

[14] Mikko Löytynoja, Nedeljko Cvejic, Tapio Seppänen, “Audio Protection with removable watermarking”, published in Information, Communications & Signal Processing, 2007 6th International Conference, pages 1 – 4.

[15] Liangbin Zheng, Ruqi Chen, Xiaojin Cheng, “Research and Implementation of Digital Rights Management Model for Vector Graphics”, published in Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on (Volume:2), pages 17 – 20.

[16] Malay Kishore Dutta, Phalguni Gupta, Vinay K. Pathak, “Perceptible Audio Watermarking for Digital Right Management Control and Research”, published in Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference, pages 1 – 5.

[17] Hu Feng ; Zhejiang Gongshang Univ., Hangzhou, China, “Study on the Measures and Basic Safeguard for Chinese Enterprises to Implement the Intellectual Property Strategy”, Published in: Management and Service Science (MASS), 2011 International Conference, Page(s): 1 – 5.

[18] Yao-Jen Liu, Shang-Jyh Li, “The intellectual property policy of Taiwan: a strategic viewpoint”, Published in: Engineering Management Conference, 2004. Proceedings. 2004 IEEE International (Volume:1), Page(s): 42 - 46 Vol.1.

[19] Fiallo, E.H., “The lack of ethics in cyber space: a case for cyber ethics”, Published in: Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference.

[20] Poonia, A.S., Bhardwaj, A. ; Dangayach, G.S., “Ethical values and practices for Cyber Society”, Published in: Current Trends in Information Technology (CTIT), 2009 International Conference, Page(s): 1 – 5.

[21] Jiaming He, Hongbin Zhang, “Digital Right Management Model Based on Cryptography and Digital Watermarking”

[22] Gregory J. Kirsch and Yannis Skulikaris “Intellectual Property Protection for Software in the United States and Europe: The Changing Roles of Patents and Copyrights”, Published in: Software Engineering, 2000. Proceedings of the 2000 International Conference, Page(s): 827.

[23] B. Vukeli, K. Škaron, “Cyber-crime and violation of copyright”, Published in: Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention, Page(s): 1127 – 1130.

[24] Stini, M., Dusseldorf ; Mauve, M. ; Fitzek, F.H.P., “Digital Ownership: From Content Consumers to Owners and Traders”, Published in: MultiMedia, IEEE (Volume:13 , Issue: 4), Page(s): 1 – 6.

[25] Farzad Sabahi, “Cloud Computing Security Threats and Responses”, Published in: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference, Page(s): 245 – 249.

[26] Mohamed Almorisy, John Grundy and Amani S. Ibrahim, “Collaboration-Based Cloud Computing Security Management Framework”, Published in: Cloud Computing (CLOUD), 2011 IEEE International Conference, Page(s): 364 – 371.

[27] Kotwal, P.A., Singh, A.R., “Evolution and effects of mobile cloud computing, middleware services on cloud, future prospects: A peek into the mobile cloud operating systems”, Published in: Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference, Page(s): 1 – 5

[28] Sharma, R., Kumar, S. ; Trivedi, M.C., “Mobile Cloud Computing: A Needed Shift from Cloud to Mobile Cloud”, Published in: Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference, Page(s): 536 – 539.

[29] Farhan Bashir Shaikh, Sajjad Haider “Security Threats in Cloud Computing”, Published in Internet Technology and Secured Transactions (ICITST), 2011 International Conference on 11-14 Dec. 2011, Page(s): 214 - 219

[30] Klemm, A, Waldhorst, O.P., “A special-purpose peer-to-peer file sharing system for mobile ad hoc networks”, Published in: Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th (Volume:4), Page(s): 2758 - 2763 Vol.4.

[31] M. Eric Johnson, Dan McGuire, Nicholas D. Willey, “The Evolution of the Peer-to-Peer File Sharing Industry and the Security Risks for Users”, Published in: Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, Page(s): 383.



APPENDIX



SECURITY ENHANCEMENT IN MULTIMEDIA RETRIEVAL VIA MOBILE PHONES Questionnaire Form

Explanation

In order for the system to make sure that it creates more security on the accessibility of multimedia files, we will do some questionnaire on users in different background of knowledge. Users will be categorized into mainly 3 types according to their frequency of computer usage and the data gathered will be analyzed independently. The video will be very brief and the video format is 3gp.

The questionnaire is categorized into 3 parts

1. General information of the users
2. Users feedback on the methodology
3. Some more suggestions from users

Activities

Phase 1: No media security applied

- Users will be explained about the brief concept of DRM and purpose of the thesis.
- Users will be given a short video clip to watch.
- The video clip will be split using third party program. Users will be asked to play each of them again.
- Users rate their feedback on this matter regarding to Digital Right Management.
- File parts of the video clip will be joined again using the same program. Users watch the same clip again after joining it.
- Users rate their feedback regarding the Quality of video clip before and after the join.

Phase 2: Media security applied

- The video clip will be split again using codes. Users will be asked to try to play each of them again. Users may use any available media player.
- Users rate their feedback on this matter regarding to Digital Right Management.
- File parts of the video clip will be joined again using codes. Users watch the same clip again after joining it.
- Users rate their feedback regarding the Quality of video clip before and after the join.

-



Please turn over

General Information of Users

(Please make ✓ sign into)

1. Gender

1.) Male

2.) Female

2. Age

1.) Below 20 years old

2.) 20 – 30 years old

3.) 31 – 40 years old

4.) Above 40 years old

3. Your relationship with multimedia

1.) Working in a field of multimedia/law/creative Design

2.) Working in a field of education

3.) A general multimedia user

4.) Hardly using multimedia

4. Exposure with multimedia files such as video clips, songs, movies.

1.) Everyday

2.) Once or twice a week

3.) Once or twice a month

4.) Few times a year

5.) None

5. Are you concerned about media files' security

1.) Yes, I do.

2.) No, I don't.

Users' feedback on the methodology

Content	Feedback		
	Agree	Disagree	Not interested
Phase 1: Third Party program			
1.1 Quality of the original clip is acceptable			
1.2 Quality of each part of a video clip after the split is acceptable			
1.3 Quality of the video clip after the join is acceptable			
1.4 Quality of the video clip after the join is equally the same as the original clip			
1.5 You notice the change in video clip after the join			
1.6 After watching the clip, you notice that the multimedia file is unprotected			
Phase 2: Codes			
2.1 After the split, video file parts are not playable			
2.2 Quality of the video clip after the join is acceptable			
2.3 Quality of the video clip after the join is equally the same as the original clip			
2.4 You notice the change in video clip after the join			
2.5 After watching the clip, you notice that the multimedia file is protected and the protective methodology works			

Suggestions by users

- 1. The strength of the system you would like to mention

.....
.....

- 2. The weakness of the system you would like to mention

.....
.....

- 3. Other suggestions

.....
.....

Thank you for your co-operation 😊

VITA



Mr. Pongsakorn Kingsuwankul

Date of Birth: 9th November 1988

Place of Birth: Bangkok, Thailand

Email address: pongsakorn.aui@gmail.com

Department: Msc. Computer Science

Science Department Chulalongkorn University



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

