View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ScholarBank@NUS

BUYER-SELLER WATERMARKING PROTOCOL IN DIGITAL CINEMA

HADY GUNAWAN

(B.Comp. (Comp. Sci.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgement

I would like to express my gratitude to Prof. Mohan Kankanhalli for constantly guiding me and giving me good advice throughout the whole process of my research. His supervision has helped me a lot in completing this project. He has been kind and understanding, even when I failed to make any progress, which has caused this whole process to be more enjoyable and made me feel less pressurized. It is really an honor for me to work with such a great professor.

Table of Contents

Summary

Digital Rights Management (DRM) has been hailed as the solution to illegal copying and distribution of digital movies. It employs many different kinds of mechanisms, such as encryption, watermarking, and digital fingerprinting, to provide a protection system to these high-valued digital assets. Not only to managing content's access control and its usage rights, a DRM system also provides a forensics tracking device called *digital fingerprint*. However, digital fingerprinting always assumes the trustworthiness of content provider, and thus may cause customers to be subjects of framing and false implication. Complete control over the generation, insertion, and detection process enables the content provider to easily reproduce the content copy sent to a user, which can be then used to accuse a user of an unlawful act he did not do.

This customer's right problem was successfully tackled by the concept of *Buyer-Seller Watermarking Protocol*, which accommodates the rights of both seller and buyer. Besides the normal digital fingerprint, another special mark, which is hidden from both involved parties, is inserted into the content, so that seller is unable to reproduce a buyer's copy and, at the same time, buyer does not have the capability to remove the special mark.

Unfortunately, every existing buyer-seller watermarking protocol either fails or relies on the trustworthiness of Watermark Certification Authority (WCA) to solve the customer's right problem. The involvement of WCA is required to generate and ensure the validity of watermark used in every transaction. As these protocols were, in the first place, assembled to eliminate the assumption on seller's honesty, a requirement of a new trusted third party is undesirable.

We address this issue by proposing three buyer-seller watermarking protocols that do not require the participation of a WCA. The watermark generator role is shifted to either customer or content provider, while still ensuring the validity of watermark used. The first protocol, a variant of Memon and Wong's protocol, depends on permutation and privacy homomorphic cryptosystem to conceal the watermark inserted. The use of watermark invariant to permutation is avoided by a watermark-validity checking. In the second protocol, customer's right problem is tackled by employing homomorphic encryption system and two kinds of permutations. The validity of watermark is guaranteed as it is generated by content provider. In the third protocol, substitution, instead of permutation, is used along with homomorphic cryptosystem to achieve the secrecy of watermark inserted. The problem of invariant watermark does not exist since the protocol uses no permutation.

Consequently, the three buyer-seller watermarking protocols proposed guarantee that the content provider has no way to reproduce the content copy a customer receives and a customer is, by no means, able to remove the watermark without rendering the content useless.

List of Tables

List of Figures

1. INTRODUCTION

Piracy has always been an issue to resolve in film industry. Illegal reproduction and distribution following unauthorized interception while films are on distribution chain from movie studios to theaters, and then to viewers, have been robbing content providers of what actually belongs to them. When analog media was reigning, although illicit copying had been causing movie studios a big revenue loss, it used to be less threatening, due to the inferior quality of the result. The complex and expensive nature of the copying process limited the quantity of illicit copy available in the market, whereas poor quality of such copy hindered people from purchasing them, giving pirates relatively little benefit from their unlawful deed.

When the world switched from analog to digital technology, an opportunity was opened for film industry to grow as digital technology promises a more affordable and easier way to produce and distribute their commercial goods. *Digital Cinema*, referring to production and distribution of a motion picture in a digital format along with the use of a digital projector for exhibition purpose [1], promises both producers and cinemas a higher presentation quality and a significantly lower production and maintenance cost. Since digital movies can be duplicated very easily without loss, it is now very simple to produce high quality copies of a movie at a very low cost. Another problem in traditional cinema is that film medium deteriorates pretty quickly due to repeated use. These degenerated prints have to be replaced in order to maintain a good show quality. Digital projection eliminates this problem [26].

In addition, the advances in computing and networking technologies have enabled high-speed communication throughout the Internet. Alongside this communication technology, digital cinema provides a very convenient and fast way to

distribute video content, an easy and immediate access to film libraries, and a strong potential for developing new business models [26].

Nevertheless, digital technology and the widespread use of Internet have caused piracy to become a much more serious concern. Unlike in the past, once pirates have access to the video data, they can now duplicate and distribute it effortlessly. Perfect duplication of digital data not only guarantees the high quality of movies distributed to cinemas, but enhances the quality of a pirated copy as well. Considering the pervasive use of Internet, which provides a fast and convenient communication channel, and the availability of peer-to-peer file sharing systems, like Napster, Kazaa, Gnutella, Freenet, etc, it is well understood how easy an illicit copy can be distributed extensively to end-users. Internet is also an open insecure channel that enables pirates to easily intercept any data sent through it. The motion picture industry in the U.S. estimates its revenue loss due to unauthorized duplication and redistribution of movies via physical media, like video cassettes, VCDs, DVDs, etc, exceeds $3 billion annually [3]. It is also reported that there are 350,000 to 400,000 illegal movie-downloads done everyday. The revenue loss due to Internet redistribution of illicit copies is estimated to be up to $4 billion annually [3].

Despite all the advantages promised by digital technology, many movie studios are still reluctant to make use of these technologies because of this piracy threat and the lack of technology that can securely protects their rights upon their digital assets. Content creators and owners are concerned about the consequences of illegal copying and distribution on a massive scale. Therefore, there is a demand for a protection system that can enforce access control and, at the same time, manage the content usage rights, such that unauthorized access can be prevented. This protection

system should be able to ensure that a digital movie is played by authorized operators, on authorized equipments, and at authorized times only. Simultaneously, it must guarantee that only certain actions under certain conditions specified by content owner can be performed on the digital content.

Digital Rights Management (DRM) system has been proposed as the solution to the security problem in digital cinema. It is the core system that allows movie studios to disseminate their cinematic assets in a secure and restricted way. As content owners specify the operations and the conditions under which they can be performed on the content, a DRM system will ensure that a digital movie can only be accessed according to the rules specified by the producing studio.

Even though we try to protect digital content from unauthorized access and manage its usage rights, all these mechanisms will be ineffectual when the movie is converted into analog signal and displayed on a movie screen. No matter how secure the access control mechanism is, a digital movie eventually needs to be presented in the clear to the viewers. Once digital content is converted to analog signal, it is no longer protected and vulnerable to illegal copying. The analog output can be easily provided as an input to a camcorder or a DVD recorder. This problem, known as "the analog hole" problem, has been responsible for most of illicit copies available at large.

Knowing that any protection systems can never guarantee a perfect security at all times, we need another technology for forensic tracking purpose. A unique identification should be embedded into each copy of the films, if possible relating the content to the people having access to it, in order to enable the copyright owner to trace back the source of a piracy act. In a DRM system, this property is achieved by

inserting a *digital fingerprint*, a user-specific distinct watermark, into every content copy to sell. Digital fingerprints serve as a forensic analysis tool that enables studios to identify the pirates upon locating an illicit copy of their movies.

Unfortunately, digital fingerprinting only supplies right protection to content provider and does not protect the rights of customers at all. It always implicitly assumes the honesty of content provider and lets content provider completely control the fingerprinting process, causing all fingerprinting schemes to be biased and unfair to customers. Content provider always knows the exact fingerprint inserted to customer's copy, so he can easily reproduce copies of the content containing a user's fingerprint and illegally redistribute them. As the result, it enables content provider to falsely accuse and frame innocent customer. This unpleasant situation defines what *customer's right problem* is. It is clear that customer's right problem actually nullifies the objective and the purpose of fingerprinting itself. It can cause an irresolvable dispute by opening a chance for a malicious user to deny his unlawful act and claim that the unauthorized copy was originated from the content provider.

To solve this customer's right problem, the concept of *Buyer-Seller Watermarking Protocol* accommodating the rights of both the buyer and the seller was introduced. However, all existing solutions that successfully solve this problem rely on the trustworthiness of Watermark Certification Authority (WCA) as a party generating the watermark used in every transaction. Since buyer-seller watermarking protocol was, in the first place, introduced to eliminate the assumption on seller's honesty, a requirement of a new trusted third party is not desirable.

We address this issue by proposing three buyer-seller watermarking protocols that do not require the participation of other trusted third party, besides the arbiter and

certification authority (CA). We eliminate the involvement of WCA without ignoring the reasons why it was initially introduced. In the first protocol, we tackle the problem caused by watermark which is invariant to permutation by requiring content provider to check the validity of watermark proposed by customer. The second protocol solves the problem by shifting back the watermark generation process to content provider. Two kinds of permutation are employed to conceal the watermark from both parties. The problem of watermark invariant to permutation does not exist in the third protocol as no permutation is involved in this protocol. Instead, substitution and encryption are used to prevent both parties from knowing the exact watermark inserted.

The rest of the report is organized as follows. In section 2, we give an overview to the notion of digital cinema and its environment. It is followed by a glimpse of digital rights management concept adapted to the digital cinema setting in section 3. We describe customer's right problem and buyer-seller watermarking protocol in section 4. In section 5, we shall present our own buyer-seller watermarking protocols which do not require the presence of watermark certification authority. Construction details comprising encryption and watermarking schemes that can be used in our protocols are discussed in section 6, whereas security analysis of the protocols is given in section 7. Lastly, we conclude our thesis in section 8.

2. DIGITAL CINEMA

In general, digital rights management is an abstract concept that can be applied to any multimedia content. However, since each type of multimedia data, be it image, audio, or video data, has its own characteristics that are unique and distinctive, it is advantageous to understand the nature of the digital content to protect and the environment in which the system will operate in order to construct a protection system with a significant effect. Therefore, in this section we shall discuss key properties of a digital movie and a simple distribution model in digital cinema. Nevertheless, we might want to first be aware of what digital cinema refers to and what the objective of an attack in the context of digital cinema is.

Various definitions of digital cinema were presented in many different publications. In this thesis, digital cinema refers to a combination of production and distribution process of a motion picture in a digital format along with the use of a digital projector for exhibition purpose [1].

In digital cinema, a pirate is a person who illegally reproduces and distributes other's digital content without the content owner's consent. It is clear that the objective of a pirate is to get an access to (newly released) very high value entertainment content of a cinematic title, which can later be duplicated and redistributed without restriction [26]. A pirate can be either a participant of the production or distribution process (an insider) or a person who is totally not involved (an outsider). While most of researchers have been emphasizing their works on protection system against outsider attacks, it is reported that 77% of illegal movie samples are originally leaked out by industry insiders [3]. Thus, building a protection system against these insider attacks is equally important.

2.1 Digital Movie

There are actually many factors that distinguish digital movie from other multimedia data. Nonetheless, we are going to discuss only some of those characteristics which are deemed to be relevant in a process of constructing a digital right protection system.

The first distinctive characteristic that a digital movie has is its huge volume. Compared to audio and image, video data has much larger size and contains more redundancy. The redundancy is caused by the high degree of similarity between neighboring video frames and the overlapping information they share. Furthermore, for the purpose of providing a high quality show, we are dealing with video data which is of higher spatial resolution, causing it to need even larger storage. Knowing this fact, we can easily see why compression plays a vital role in digital cinema.

In order to get a clearer idea on how big the volume of a digital movie is, let us illustrate it with an example from [1]. Consider a movie stored at 24 frames per second, each frame consists of 1024 rows and 1280 columns, and each pixel is stored with 10 bits each of red, blue, and green. A two-hour movie would require almost 800 Gigabytes plus maybe 10% audio. After compression, the size is reduced to the range of 50-100 Gigabytes while still maintaining sufficient fidelity. In fact, this number does not well picture the real situation in digital cinema. In this example, those numbers represent 1K spatial resolution, whereas in practice a movie distributed to theaters should have spatial resolution of 2K to 4K.

The second feature differentiating a digital movie from other multimedia is its value curve. When it is first released, a movie has an extremely high value. This initial value can be up to hundreds million dollars. However, it never lasts long, it

declines very rapidly after few weeks from its release date. It is reported that the value can go down by millions of dollars in one day. For example, DreamWorks' Shrek 2 grossed about US$270 millions dollars within the first two week of its release in the U.S. [51]. However, it made only about US$100 millions dollars during the next two weeks, which indicates more than 60% decrement from that in the first two weeks. Overall, Shrek 2 managed to make 83.5% of its total revenue of US$436.722 millions within one month of its release in the U.S. Please refer to figure 1 for the value curve of movie Shrek 2 in its first ten weeks. The figures shown on the chart are taken from [51].

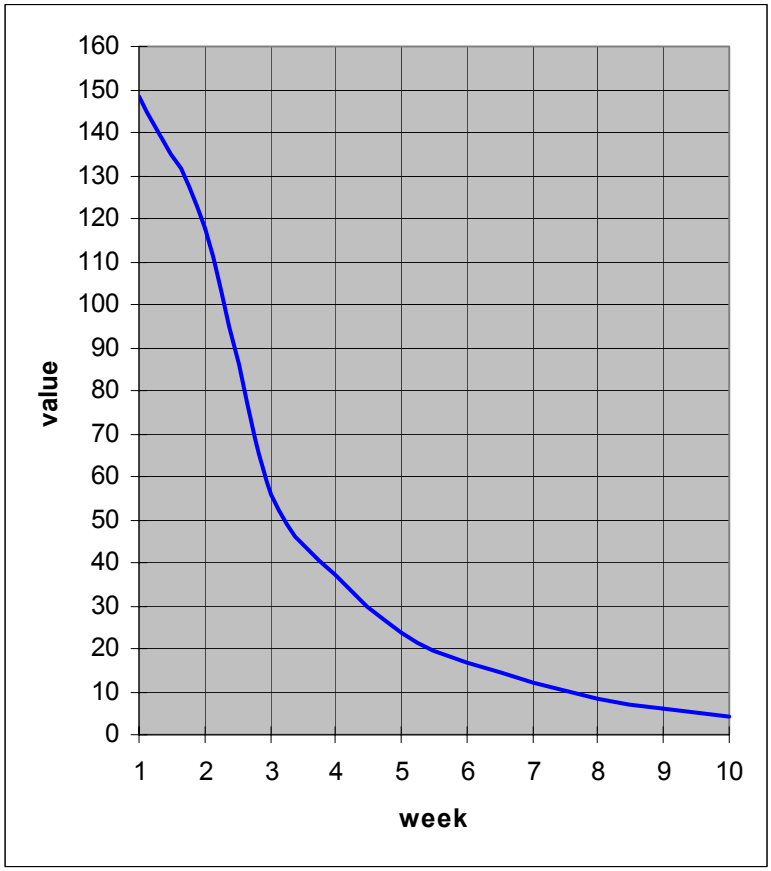

Figure 1. Value curve of the movie Shrek 2

From the graph shown above, it is clear that the biggest part of total exhibition revenue is made during the first few weeks after the movie is released. As the

consequence of this unique characteristic, we can deduce that the time span during which protection system is crucial is very limited. Piracy threat must be handled much more seriously during this critical range.

Another important aspect that should be taken into consideration when designing a digital assets protection system in digital cinema, although it is not unique to video data only, is the fact that digital content can be effortlessly copied, altered, and distributed in a relatively short time. The fact that a lossless, if not exactly the same, copy of digital content can be easily produced, not only benefits content providers, but assists pirates to produce illegal copies of good quality as well. Protection system must be designed in a way, such that the illegal copying will result in a drastically degraded quality video.

2.2 Distribution Model in Digital Cinema

From the studio, a movie must be distributed to the theaters to be able to be enjoyed by the viewers. The knowledge about the distribution process is important in deciding how the protection system should work. The distribution model we are going to present is adopted from Liu et al.'s work [34].

Usually there are four parties involved in a basic distribution process, they are content provider, distributor, consumer, and clearinghouse. In real life, there might be an e-commerce system integrated to the distribution system to handle the financial payment and to trigger the function of clearinghouse. This system normally involves another party. Nevertheless, it is outside the scope of the project and will not be explained further in this thesis.

- **Content Provider** is the digital rights owner of the digital content, who wants to protect these rights of theirs against the act of piracy. In the context of digital cinema, content providers will be movie studios who produce the films.
- **Distributor** is a party who provides the distribution channels for digital content to be delivered from content providers to consumers. Upon receiving the digital content, distributors create a catalogue presenting the content and the right metadata for the content promotion.
- **Consumer** is a party who accesses and uses the digital content. Consumers obtain the digital content from the distributors and buy licenses to access the content from clearinghouse. In the context of digital cinema, consumers correspond to movie theaters where digital movies are shown to the viewers.
- **Clearinghouse** is a party who handles digital licensing by issuing and controlling the rights to access the content. Clearinghouse issues a digital license in exchange with consumer's payment. Royalty fees and distribution fees will then be paid to the content provider and the distributor, respectively.

Clearinghouse is not necessarily a separated body; sometimes it can be combined with the distributor or the content provider itself. In that case, the responsibility of handling digital licensing will be shifted to the corresponding party. Please refer to figure 2 for a typical distribution model in digital cinema. The diagram of the distribution model is a modified version of diagram of DRM model presented in [34]. The diagram is adjusted to the context of digital cinema in order to increase its relevance.

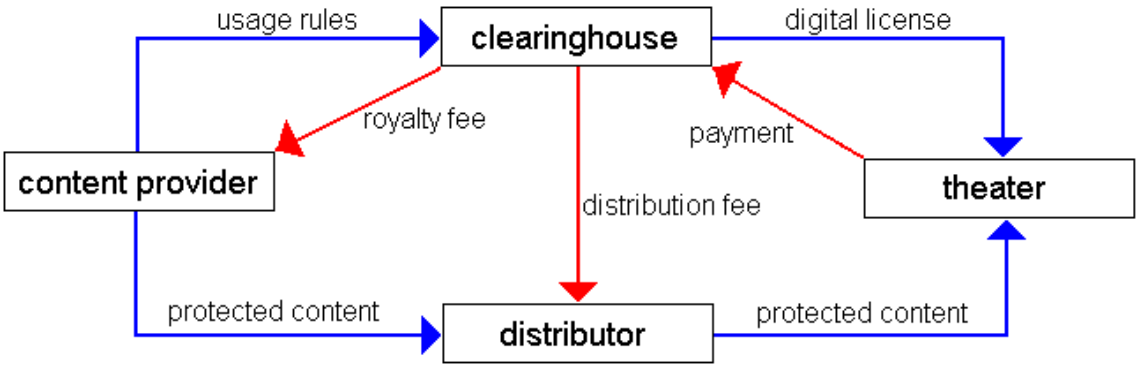

Figure 2. Distribution model in digital cinema

The distribution process usually flows in the following way:

First, the content provider encodes the digital content and then packs it for the preparation of distribution process. Subsequently, the digital content is transferred to the distributor, whereas the usage rules are sent to the clearinghouse. Consumer will then get the digital content from the distributor and request for a valid license from the clearinghouse. Upon receiving a license request, the clearinghouse will authenticate the consumer. Only after verifying consumer's identity and receiving consumer's payment, a digital license indicating the usage rules and the rights given to the corresponding consumer is sent to the requesting consumer. The consumer will now be able to access the digital content according to the usage rules specified by the content provider. As the digital content moves from the content provider to the consumer, the payment moves in the opposite direction, that is from the consumer to the content provider.

The distribution model explained above is a simplified form of the real world situation. In real life, as digital cinema involves a vast market, scattered all over the world, the distribution process is done in a multi-layered manner and the digital content must go through a chain of distributors before it can reach the consumer. As the result, distribution process can be pictured as a tree-like hierarchy. Figure 3

displays an example of this tree-like hierarchy. This figure is adapted from Kirovski et al.'s work [26].

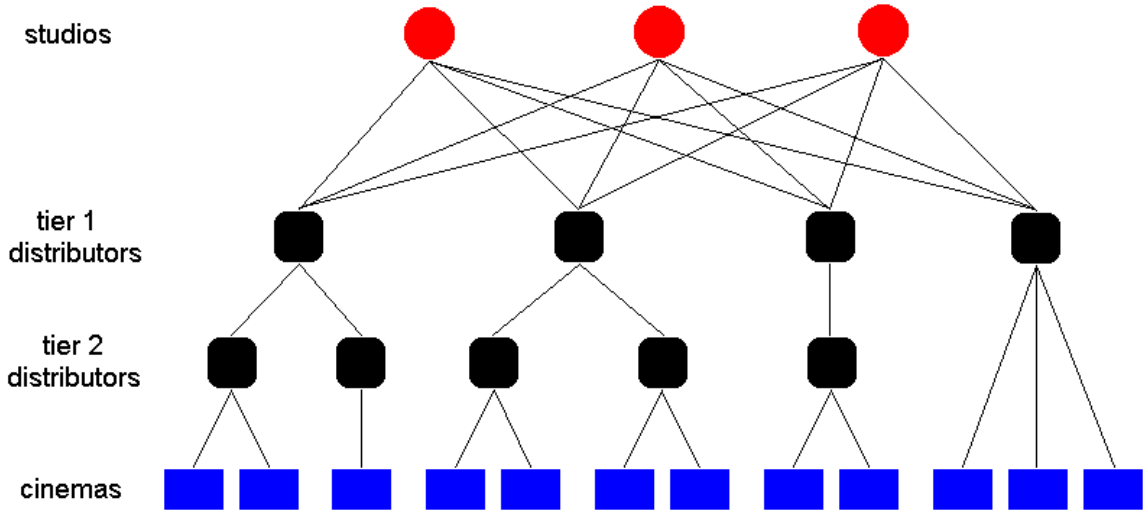

Figure 3. An example of distribution hierarchy

Besides that, unlike illustrated in our distribution model, in reality digital cinema involves a large number of content providers, distributors, and a huge number of movie theaters and their multiple projectors. However, compared to other applications, like video/audio broadcast, music-on-demand, and video-on-demand, the set of participants in digital cinema context is relatively smaller (several hundred thousand projectors worldwide versus tens, or even hundreds of millions of satellite TV receivers)[26].

Another aspect differentiating digital cinema to other applications is the playback device. Compared to those used in other applications, the projectors used by movie theaters are much more costly because they contain expensive optical equipments which are functional in guaranteeing a high quality show. Together with the relatively smaller set of participants, this fact allows content providers to implement a more sophisticated protection system without causing a significant increase to the total cost.

3. DIGITAL RIGHTS MANAGEMENT IN DIGITAL CINEMA

In this section, an introduction to the notion of Digital Rights Management (DRM) will be first given, followed by the requirements of a DRM system in digital cinema and some works that have been done in this area. A short description and the objectives of DRM are presented in the first part of this section. The second part of this section explains the eight properties that are demanded from a DRM system in digital cinema. In the last part of this section, we will give an overview of some ideas proposed by many different researchers to solve the movie piracy problem.

3.1 DRM: Definition and Objectives

To date, there has not been standardization of the definition of Digital Rights Management (DRM). DRM is defined in many different ways in the literatures; some of the definitions are listed below:

- The Association of American Publishers defines DRM as the technologies, tools, and processes that protect intellectual property during digital content commerce [20].
- According to Eindhorn, DRM entails the operation of a control system that can monitor, regulate, and price each subsequent use of a computer file that contains media content, such as video, audio, photo, or text [20].
- Gordon describes DRM as a system of information technology (IT) components and services that strive to distribute and control digital products [20].
- Emmanuel and Kankanhalli define DRM as a set of technologies and approaches that establish a trust relationship among the parties involved in a digital asset creation and transaction [21].

Although those definitions have various ways of phrasing in describing DRM, they basically share a common idea. In general, DRM refers to a system that protects high-value digital assets by controlling the distribution and usage rights of those digital assets.

From its definition, we can deduce that the objectives of a DRM system are as follows:

- To ensure secure distribution of the content and to avoid attackers from intercepting the content while being delivered from one point to another in the distribution chain.
- To enforce access control on the digital content and to prevent unauthorized access to the content.
- To protect the copyrights of the digital content and to avoid illegal copying and distribution of the content.
- To manage content usage rights and to ensure that access to digital content is allowed only under the conditions specified by the content owner.

The core concept used in DRM is the separation between the digital content and the rights ruling the content access. Instead of buying the digital content, the consumer purchases a digital license granting certain access rights to him. A digital license is a digital data file that specifies certain usage rules for the digital content [34]. The idea is to allow protected content to be distributed without restriction and to ensure that this protected content is nothing, but garbage without the presence of a valid digital license. As the consequence, the protection and distribution of the content can be separated from those of the rights.

3.2 DRM Requirements in Digital Cinema

As mentioned in Section 2, digital rights management generally can be applied to any multimedia content. Nevertheless, every application has different set of requirements to fulfill. Consequently, DRM must be adjusted specifically according to the requirements demanded by the application in order to achieve maximum result. In this section, we shall see the requirements that a DRM system should satisfy in the context of digital cinema. The list of requirements presented below is accustomed in line with the characteristics of digital movie and distribution model presented in the previous section.

Basically, all the requirements of DRM in digital cinema can be classified into eight major groups: concealment, access control, content usage rights management, forensic tracking, quality of service, efficiency, scalability, and renewability. Each of these eight requirements is explained elaborately below.

3.2.1 Concealment and Content Protection

Concealment is responsible for nullifying an attack in which a pirate tries to intercept the digital content while it is being distributed from the movie studios to the movie theaters. The content should be protected in such a way, so that attacker will not be able to access the content, even though he successfully intercepts the protected content. A DRM system must ensure that the protected content has no value and appears random without the appropriate secret key. In other words, it should be useless for user to steal protected content without stealing the secret key locking it.

As pirates may try to steal digital content at any stage of the distribution process, the content protection system must be persistent, i.e. it has to stay with the

content wherever it goes. The content must be protected not only while it is being transferred on an insecure channel from one party to another, but also when it is in transit from one distribution stage to the next. Thus, we also require each party involved in the distribution process to be a secure repository for protected content with capability of securely performing:

- Authentication: to ensure that the party interacting with them is indeed a legitimate party as well.
- Rights management (licensing): to prevent unauthorized user from accessing the content and to ensure that every user can only perform actions that are specified in their licenses.
- Content encryption and decryption: to prevent pirates from getting an access to the unprotected content, although he successfully steals the protected content from the repository.
- Fingerprint embedding and detection: to provide a pirate-tracking tool.
- Integrity checking: to prevent the protected content from being tampered with by an attacker.

In order to further tighten the security, each party involved should employ a tamper-resistance mechanism, either tamper-resistance hardware or software, in their systems, so that the cost of initial attack increases and pirates are deterred from stealing the protected content.

It is also important to ensure that the protection system is embedded into the content itself and not into its header. The fields in the file headers are often static, and therefore they can be guessed from information in the bit stream, or they can even be

ignored. Hence, a protection system applied to the content header can be easily broken by simply discarding the protected header.

It may seem that the content is safe once we can protect the content in accordance with our discussion above, but there is actually one more way for pirate to obtain the content without having to break the protection system, the analog hole. No matter how secure the protection system is, a digital movie eventually needs to be presented transparently to the viewers. As mentioned in the earlier part of the report, when a digital movie is converted into analog signal and displayed on a movie screen, it is vulnerable to illegal copying. Therefore, besides protecting the digital content, we need to protect the analog output as well. A DRM system should be able to tackle this problem by ensuring that capturing the analog signal using camcorder will result in a severely degraded copy of the content, or even result in a totally random signal.

3.2.2 Access Control

Access control is an important part of a DRM system that is used to prevent unauthorized access to the digital content. In digital cinema, a DRM system should help the movie studios to ensure that their movies can only be accessed by authorized operators on authorized equipments and at authorized times. Therefore, authentication process must take place before a DRM system decides whether or not to give access right to an individual. Every access request from an unauthorized user must be turned down by the DRM system. Moreover, a DRM system should guarantee that a digital movie can only be accessed under certain conditions as well. DRM should provide a kind of conditional access to digital content, such that access is only allowed when a set of rules has been satisfied.

As explained in the previous subsection, the digital content and the digital license granting users rights to access the digital content are managed and distributed separately. This separation concept is the backbone of the access control in a DRM system. Possession of a valid digital license can determine whether an individual has the right to access certain digital contents. Usually the protection system providing secrecy of the digital content is combined together with the concept of digital license in order to enforce access control mechanism. The secret key that can unlock the protection system is integrated into the digital license, such that only authorized users having valid licenses can access the content.

Since digital licenses plays such an important role in enforcing access control, a secure protection system must also be applied to them. Similar to the content protection, a protected license should appear random, such that attackers cannot extract any information about the digital license without the corresponding key. The protection has to stay with the license both while it is being distributed on an insecure channel and while it is being stored by any party involved. Again, it is done in order to avoid attackers from learning about the information stored in the digital license without first breaking the protection system.

As the content provider might give different set of rights to each user, a digital license received by one user might differ from that of another user. In order to prevent attackers from swapping their licenses with a more “powerful” license of others, a digital license should be linked to the identity of the owner and it should not be transferable to other parties. The clearinghouse, therefore, should perform secure authentication before issuing and verifying a digital license in order to get the identification of the user and at the same time validate that he is indeed a legitimate

user. Besides authentication, integrity checking must also be performed by the receiver of the license in order to avoid the license from being tampered with by attackers. Last but not least, non-repudiation in right issuing must be enforced to prevent illegal right issuing.

3.2.3 Content Usage Rights Management

Content usage rights need to be managed in order to prevent malicious theaters from illegally copying and editing the content. A DRM system must help the movie studios to ensure that only certain actions can be performed on their digital movies.

As the first step of content usage rights management, the content provider must specify the set of operations that can be performed on the content and the conditions on which they can be carried out before the content is distributed to the movie theaters. Unlike the digital license, these action-condition pairs should be embedded to the digital content, so that a DRM system can always refer to them before granting users a permission to execute the requested operation. Similar to the content protection system, the action-condition information should not be embedded into the content header. Otherwise, attackers can simply remove the header to break the content usage rights management system.

Once the content usage rights are embedded to the content, it is a DRM system's responsibility to ensure that an action can only be performed on the content if it is specified by the content provider and all the conditions have been fulfilled.

3.2.4 Forensic Tracking

As no protection system can ever guarantee a perfect security at all times, we need forensic tracking technology to trace back the source of a piracy act. A unique identification should be embedded into each copy of the films, relating the content to the people having access to it, in order to enable movie studios to identify the pirates.

A DRM system should embed this unique identification imperceptibly, such that it is impossible, except by guessing, for attackers to locate the positions where the unique identification is embedded without knowing the secret key used in the embedding process. The marked content must be visually indistinguishable from the original copy of the content. Robustness is another important property that a DRM system should guarantee. The unique mark should survive common signal processing operations, like scaling, cropping, translation, rotation, filtering, noise reduction, and change of brightness. In other words, it should be infeasible for attackers to alter or remove the unique identification without causing significant damage to the content. Therefore, a DRM system should never insert the fingerprint into the content header lest pirates discard the header to disable the tracking mechanism.

In order to guarantee the reliability of the identification code, DRM must ensure that the codes are collusion-resistant and frame proof. No coalition of users should be able to collude their marked copies in order to erase the identification code. Neither should users be able to fabricate the unique identification for the purpose of framing innocent users. The forensic tracking mechanism should be designed in a way, such that the code detected in an illicit copy always refers to at least one of the pirates and never points to an innocent user. Even though some users collaborate and

collude their marked copies, the remaining code should always enable the content provider to identify at least one of the pirates.

Besides preventing a group of malicious users from framing other users, it is also important to prevent the content owner from producing fake proof in order to accuse an innocent party of a piracy act.

3.2.5 Quality of Service

In spite of all the technologies employed in a DRM system, quality of service must not be affected. Any mechanisms used to provide content protection, access control, usage rights management, or pirate tracking should have an insignificant impact on the visual quality of the digital content. The distortion caused ought to be imperceptible, so that the high fidelity of the digital movie is sustained.

Hindering the viewing experience of the audience should never be an option in the movie industry. Therefore, a DRM system has to be constructed with quality degradation as the function to be minimized.

Moreover, a DRM system should ensure that any potential failure, for example clearinghouse server breakdown, would not interfere with the ability of the theaters to exhibit the movies and detract from the paying viewer's experience.

3.2.6 Efficiency

Efficiency measures the practicability of a DRM system. We do not want to use a system that takes million years to process a movie, uses all the storage available in this world, or costs us more than the value of the content itself. Hence, we should

limit the amount of space, time, and money used to implement a DRM system. The smaller amount of resources a DRM system needs, the more feasible it is.

As mentioned in the earlier part of this thesis, a digital movie has a huge volume, and thus compression has an important part to play in digital cinema. In order to achieve storage efficiency, any mechanism deployed in a DRM system should have a limited impact on the compression ratio. These technologies should not cause the compression to become ineffective by introducing more redundancy than the compression algorithm can eliminate.

Because of the security mechanisms, a digital movie must now be preprocessed before it can be played on the screen. In order to maintain the quality of the show and to stream the movie in a smooth continuous manner, we require those security mechanisms to have a real-time performance. The amount of time consumed to apply the security mechanisms on the content is also crucial in the distribution process. Since the content provider needs to send a great number of copies to a great number of movie theaters, a DRM system with a non-polynomial processing time is simply undesirable.

In terms of finances, the implementation of DRM should not cause a significant increase in the production, distribution, exhibition, and maintenance cost. It must be guaranteed that the total cost does not exceed the value of the digital content itself, because there is no one in this world who would spend $1 million to protect a $100K asset. So far, a high price to pay is one reason why movie studios are still hesitant to switch to digital cinema framework.

3.2.7 Scalability

Scalability of a DRM system is defined as the flexibility of the system's network to be expanded or shrunk upon changing the set of participants. In digital cinema, the set of parties involved in the distribution process of a cinematic title might be different from that of another title. Movies which are more popular have larger distribution network, whereas less popular movies have typically smaller distribution network. As the set of participants changes every time movie studios want to distribute a digital content, total reconstruction of the DRM system and key management for each change is definitely not desirable.

It should cost little effort, time, and money to adjust the DRM system to such changes. Movie theaters and distributors should be able to join and leave the system's network without messing up the whole rights protection system. At the same time, the content provider should not need to restructure the whole DRM system after expelling a party from the network. In other words, a DRM system should be flexible to the network resizing without compromising the security aspect of the system.

3.2.8 Renewability

Renewability indicates the ability of a DRM system to recover after a successful attack. Again, no system can provide perfect security. Eventually, attacker will succeed in finding a way to break the protection system. Thus, renewability does matter in designing a digital right protection system.

The protection system must be designed in a way, such that the impact of an attack is localized. The content provider should be able to isolate the part of the system that has been compromised, so that it will not affect the other parts of the

system. It is also vital to guarantee that by successfully breaking the protection system, an attacker can only obtain an access to a very limited number of cinematic titles (one is the best).

Furthermore, it is important to ensure that the system can be renewed within a very short period of time using very little resources in an effortless manner. The system should be able to resume immediately after a successful attack and the total cost the content provider needs to pay to recover the system from a compromise should be as small as possible. A thorough system restructuring should be avoided as well.

After discussing the ideal situation desired in digital cinema, it is easy to see that DRM is a very complex system. No single technology could stand alone to satisfy all the requirements. Instead, we need to combine several security concepts and many solutions together in order to make a maximum contribution. Some common technologies employed in DRM systems are encryption, watermarking, digital fingerprinting, message authentication code (MAC), and digital signature.

3.3 Related Works

In this subsection, we shall see some works that have been done in order to build a DRM system in digital cinema. Overview of the contribution made by each work will be presented together with its strengths and limitations.

3.3.1 DRM in Digital Cinema

Many research works [1][26][30][31][33][34] agreed that the combination of encryption and digital watermarking is the solution to the rights management problem. Encryption is used to provide the concealment property by protecting the digital content while being distributed to users. At the same time, encryption enforces access control on the content by allowing only users having the right decryption key to access the content. The distribution of decryption key to the users is done by implementing the concept of digital license. Digital license containing the decryption key is delivered to the users after their payment is received. In order to prevent malicious users from misusing the license, digital watermark stating the action-condition pairs allowed to be performed on the content is embedded to the content. Each time the playback device receives a user request to access the content, it will check the conditions stated in the watermark before deciding whether the access right will be granted to the requesting user. A unique user-specific watermark, also known as a digital fingerprint, is embedded to the content, so that the content provider can keep track every copy of the content distributed to the users. A digital fingerprint is also used as a forensic tracking tool whenever the content provider successfully locates an illicit copy. Unfortunately, even though these works proposed a set of technologies that can be employed in DRM, they did not specifically explain how each technology should be applied on the content.

Besides explaining how encryption and watermarking can be useful in DRM, Liu et al. [34] presented a DRM model involving four parties: the content provider, the distributor, the clearinghouse, and the consumer. They pointed out that digital license is the core concept of DRM and illustrated how digital license concept is

applied in a DRM system. Some cryptographic mechanisms mentioned in this work are symmetric/asymmetric encryption, digital signature, one-way hash function, and digital certificates. Tamper resistance technology is also mentioned as the supplementary security mechanism. They closed with a brief explanation on privacy, fair use, and usability concerns.

Bloom [1], not only discussed about encryption and watermarking, but also addressed the "analog hole" problem. He mentioned that embedding watermark to the content could not solve this problem unless all camcorder producers agree to integrate a watermark detector to their devices. Instead, he suggested *camcorder jamming*, a technology to interfere with the ability of camcorder to record a movie in a theater, as a better solution to this problem.

In order to protect the integrity of digital license, Kirovski et al. [26] suggested appending the hash value of the content and license, which is signed by the distributor, to the digital license, so that it can be verified before accessing the content. Moreover, they mentioned briefly about employing error-correcting code to construct a fingerprinting scheme that is collusion-resistant and frame proof. A special kind of error-correcting codes is used to provide a set of fingerprints to embed. These codes are designed in a specific way, so that by colluding a subset of codewords, it will result in neither another codeword (frame other user) nor a zero vector (erase the fingerprint). However, this approach is only effective for small number of users. As the number of users grows, this method becomes impractical.

In addition to explanation on general concept of encryption and watermarking in DRM, Linnartz et al. [33] proposed the use of physical mark on the media where an authorized copy is stored in order to prevent playback devices from playing an illicit

copy resulted from camcorder copying. Playback devices must match the watermark embedded in the content with the physical mark before granting user an access to the content. They also suggested a method to enable user to copy the content for limited number of times, which they called the *ticket* concept. Let *m* be the number of copy operations allowed to be performed on the content. The results of passing a random number through a cryptographic one-way function *F*, *n* and *n-m* times, denoted by *W* and *T* respectively, are embedded to the content. Every time a user requests for a right to copy the content, playback device checks if $F^p(T)$ is equal to *W* for some $p > 0$. If yes, copy operation can be carried out, and then *T* will be changed to $F(T)$. Otherwise, the request will be rejected. However, physical mark concept does not allow user to copy the content at all, and their copy generation control does not stop users from making unlimited number of copies using camcorder.

After giving a brief explanation on *Potato system* that convinces customers to pay for digital contents because of the advantages and provision promised for paying customers, Grimm and Aichroth [24] introduced the concept of Lightweight DRM (LWDRM) that relies on the responsible behavior of the customers. LWDRM involves two file formats: local media file (LMF) and signed media file (SMF). After making the payment, customer will receive LMF file from content provider, which consists of the content encrypted using AES and the key encrypted using customer's public key. Thus, this type of file cannot be transferred outside of the receiving device. A user can transfer the content by first producing its corresponding SMF file, which consists of encrypted and watermarked content and the key "signcrypted" using his private key. This deters users from transferring the content illegally as it contains his signature. To address privacy issue, Grim and Aichroth suggested the use of

pseudonyms as customer identifiers. Nonetheless, this method does not protect the content from camcorder recording.

Byers et al. [3] classified attacks into two groups: insider and outsider attacks. They studied 285 movie samples available on file sharing networks in order to find out the source of the leakage and the date of availability of those illegal copies. They suggested to define a procedure for tracking where the artifact is at all times, as well as who is responsible for it, as a short-term mitigation. They proposed a monitoring system done by human resources, allowing access to digital content only with the presence of an authorized party, to prevent insider attacks. As medium-term mitigation, they proposed the concept of trusted content player, which is tamper resistant and acts as a content storage device. A user must enter a one-time password to access the content on the trusted device. At playback, the player would project a tracking code on top of the content. Although short and medium term mitigations were discussed, they did not present any long-term mitigation. They presented their proposed solutions at a very abstract level and they did not explain the details of these solutions, making them too general to implement.

Chong et al. [10] proposed the idea of a second level of management and control in their Security Attribute Based Digital Rights Management (SABDRM). Instead of relating the identity of a user directly to his rights, they proposed the concept of security attributes that bridges the identity and the rights of a user. These security attributes, which may include role, group membership, time and location to access the content, etc, together with the identity of a user determines the contents that the user can access and the rights that the user may exercise on the contents. The way SABDRM works is highly similar to the standard DRM: the content is distributed in a

protected form and access is enabled only with the presence of a digital license containing the decryption key and the set of actions a user can perform on the content. Another unique feature of SABDRM is that each copy of content is encrypted using a user-specific key, so each user receives different copy of protected content. However, except determining the rights that a user has together with the identity of that user, security attributes are redundant and useless. They only complicate the system and make SABDRM not suitable for large number of participants. Moreover, user-specific encryption keys make key management even more complex. Although it can avoid collusion and framing problem, it cannot survive camcorder recording.

Although it is a secure multicast protocol that is presented by Chu et al. [11], their work shares some common aspects with DRM. Similar to a DRM system, their protocol also relies on the concept of encryption and watermarking to provide access control and forensic tracking mechanisms. Each message sent is encrypted, and each authorized member will obtain the decryption key from the group leader. In order to get the ability to trace back the source of leakage, sender produces two different watermarked copies of each frame of the video, encrypt them with different keys, and multicast both copies. The group leader will generate unique random string for each member to indicate which sequence of watermarked copies that particular user can access. So, each user receives a different set of decryption keys. Unfortunately, their mechanism can only detect collusions with a small collusion group. Tolerating more detection error or generating more watermarked copies for each frame can help, but they can cause unreliability and inefficiency.

3.3.2 Video Encryption

Tosun and Feng [52] proposed a light-weight, multi-layered video encryption algorithm that encodes only some parts of the video while still providing reasonable degree of security. The video is first processed using 8×8 block discrete cosine transform (DCT) compression. Two breakpoints, loss-tolerant and security breakpoints, will be then set to partition the coefficients into 3 groups: base, middle, and enhancement layer. Base and middle layer are encrypted using VEA1, while enhancement layer is left unprotected. VEA1 divides data into two groups based on a secret key, and then XOR operation is carried out between the two groups. The result of DES encryption on the second group will be then appended to the result of XOR operation to form the ciphertext. This method allows user to adaptively set the breakpoints to balance the security and performance according to his need. Tosun and Feng also presented an algorithm to determine breakpoints adaptively when a target bandwidth rate is provided.

In 2001, Tosun and Feng [53] proposed another video encryption algorithm. This time, an error preserving encryption mechanism is specially designed for transmission of video over wireless network. Standard cryptosystem cannot be used to protect content sent over wireless network because of their error propagation property and the avalanche effect. A single bit error can cause the protected content to be decrypted to garbage since they do not preserve the transmission errors. In order to solve this problem, Tosun and Feng constructed an encryption system based on the concept of error preserving function. If plaintext *x* and *y* differ at *i* positions, then their encrypted form, $E(x)$ and $E(y)$, also differ at *i* positions. They explained that this kind of functions could be generated using permutation and complementation of a

subset of the bits. This very fast encryption method successfully solves the transmission error problem, but it is lack of security property and vulnerable to known plaintext attack.

By presenting a video restoration algorithm based on motion vectors only in the beginning of their work, Liu and Li [35] showed that encrypting only pixel data residing in I frames is not enough and motion vectors alone are sufficient to restore reasonable apprehensible video streaming data that are recognizable by humans. Thus, they proposed an algorithm to encrypt these motion vectors residing in P and B frames of a video as a complement to the I frame encryption. Their encryption method consists of two steps: concealing and distancing. In the first step, motion vectors are XOR-ed with a random number to wipe off their static features. Then, the resulting vectors are scrambled according to a set of mapping tables to hide their spatial relationship. The random number table and mapping tables are re-generated using some random number generator controlled by a secret key each time the algorithm is invoked. Therefore, the security of their method relies on that of the random number generator. As motion vectors consume over half of the video stream bandwidth and they encrypt all of them, this method causes a significant overhead to the overall encryption performance.

Based on Claude Shannon's work, Lookabaugh and Sicker [36] explained how selective encryption could even produce better security as it only encrypts important part of the data, and thus reduces the amount of material that can be used to attack the encryption algorithm. They presented two simple algorithms to illustrate the idea of selective encryption. The first algorithm uses a 3-bit scalar quantizer to convert continuous valued input to one of the eight possible 3-bit words. Selective encryption

involves scrambling a few most significant bits of those words. In-the-clear portion of the stream is statistically independent of the scrambled portion, so it does not help attackers to guess the scrambled portion. However, this kind of encryption cannot recover the original data perfectly due to some information lost during the quantization process. The second method suggested the encryption of a portion of bits in the headers of a video data. This method is very fast, but it has serious security problem. As the fields in the file headers are often static, they can be guessed from information in the bit stream, or they can even be ignored.

Chiaraluce et al. [7] proposed a video encryption algorithm that uses three chaotic functions to encrypt the most significant bit of the DC coefficient of DCT, the AC coefficients of the I frames, the sign bit of the AC coefficients of the P frames, and the sign bit of the motion vectors. The input and the parameters of the skew tent map CM_1 and the sawtooth likewise map CM_2 are generated using a secret key. The real numbers produced by CM_1 and CM_2 are summed up together, and then scaled to obtain a number between 1 and 256. This number will be used as the input of the logistic map CM_3. On the input number, CM_3 is applied 64 times to produce a sequence of 512 bits, which will be XOR-ed with the content to produce the ciphertext. The chaotic sequence produced by this sequence of operation is quite similar to white noise, making the ciphertext appear random as well. Nevertheless, this method involves a quite complex set of computations, causing its performance to be slightly inferior to other selective encryption schemes.

Shieh [48] introduced a video encryption algorithm called *Take*, *Skip*, and *Permute* (TSP), which is based on entropy coding. According to his method, the content will be first compressed using Huffman entropy coding and encryption starts

only after the compression process is completed. Once the entropy-coded stream is produced, starting from the beginning of the stream, a few bits are taken randomly, followed by selectively skipping a sequence of bits before the next taking process. These taking and skipping process are repeated until we reach the end of the stream. The permutation process will then take place to shuffle all those chosen bits. So, after the permutation process, the stream is partly scrambled. The positions of chosen bits, the number of bits to skip, and the permutation table are all controlled by a secret key. Although this method is very simple and fast, it is vulnerable to known plaintext attack. If both plaintext and ciphertext are known, attackers can try to observe the difference and guess the three parameters controlling the encryption.

Zeng and Lei [57] proposed a frequency domain video encryption system, in which video data are concealed by employing bit and block scrambling. The input video signal is first transformed into frequency domain and decomposed into subbands by performing 2D wavelet transform. The sign bit and refinement bits of each coefficient which are not highly compressible are selected for scrambling. Then, each subband is divided into a number of blocks of the same size. Within each subband, these blocks of coefficients are shuffled. In order to further improve the security, each block of coefficient can be replaced by one of its eight rotated versions. The result of this rotation process is the ciphertext of the corresponding input. The bit scrambling, block shuffling, and block rotation operations are all controlled by a secret key. Zeng and Lei also mentioned that an 8×8 block based DCT can be used instead. After dividing the coefficients into segments, DC and AC coefficients within each segment are scrambled. The sign bits are also encrypted by flipping the sign randomly or with respect to a threshold. These scrambling and sign flipping can be

applied only on the I frames and I blocks in the P/B frames to reduce the computation complexity. To avoid motion vectors from leaking some information about the video, their signs can be encrypted in the same way.

3.3.3 Digital Watermarking

Digital watermarking is a technique for embedding a message into a digital content by imperceptibly modifying the content. Readers might want to refer to [15] for an overview to digital watermarking concept. Some existing watermarking techniques are presented below.

Dittman et al. [16] presented a watermarking classification dividing watermarks into five groups based on their application area. Two types of watermarks mentioned, fingerprint and copy control watermarks, play a very important role in a DRM system. They later described the requirements of each class of watermarks with respect to six properties of digital watermarking and several types of possible attacks for each class. Both fingerprint and copy control watermarks require high robustness, high security, and imperceptibility. However, fingerprint watermarks have higher complexity and its detection uses non-blind method, whereas copy control watermarks should have low complexity and its detection should be done blindly. In a blind watermarking technique, watermark detection can be done in the absence of the original unwatermarked content, whereas a non-blind technique requires the presence of the original unwatermarked content in the detection process. They also mentioned about StirMark Benchmark, an automated evaluation architecture for multimedia watermarking. The idea is to put different watermarking methods to a series of tests

and attacks, followed by the detection process, to measure the reliability of each method.

Wessely et al. [56] proposed a video watermarking algorithm that uses a two-dimensional discrete wavelet transform (DWT) based on the simple Haar-wavelet. DWT approach is chosen as the result of an extensive benchmark showed that it achieved the highest robustness, whereas Haar-wavelet is selected because its low- and high-pass filters are computationally inexpensive to implement. According to their method, the watermark is embedded into the LH_3 horizontal high-pass subband of the blue color channel with a set of twelve Walsh-series as the carrier. The detection can be done blindly by estimating the watermark bit with respect to the correlation between the Walsh pattern and the LH_3 coefficients. To further improve the robustness against attack like deletion, duplication, or swapping of video images, they suggested an idea of embedding more than one copy of the watermark. The concept of content adaptive energies was also proposed to improve robustness without causing any perceptible visual artifacts.

The watermarking scheme proposed by Cheng and Huang [5] first applies the pyramid transform to preprocess the I frames of the video. Pyramid transform is adopted for its multiresolution, low complexity, good prediction, and easy control of embedding errors. The watermark is embedded in the pyramid transform domain with the modulation magnitude that is maximized under the fidelity constraints to achieve the best robustness and detectability. Optimum decision rule derived using the statistical model of the generalized Gaussian distribution is used to detect the embedded watermarks blindly. Experiments demonstrated that their watermarking scheme has low visual distortion, high robustness, and accurate detection.

In [38], Lubin et al. proposed a forensic digital watermarking system to enable content provider to trace back the source of piracy act. They first pointed out that unlike the other types of watermark, detection of forensic watermark could be done with the presence of the original video and detection need not be performed in real-time as detection is only done occasionally by the content provider. They achieved the robustness and imperceptibility properties by restricting the watermark pattern to be very low frequency in both space and time. The high degree of information in the low frequency components makes them difficult to distort without degrading the fidelity. At the same time, human beings are insensitive to low frequency distortions, guaranteeing imperceptibility of the watermarks. They chose the carriers based on the concept of sub-threshold summation, such that inserting one of them would not cause any visual artifacts, but inserting many of them would produce visible distortions. They mentioned that the concept of error-correcting codes could further improve the security of their method.

Lu et al. [37] introduced the concept of video frame dependent watermark (VFDW) in order to achieve robustness against two kinds of watermark estimation attacks (WEAs), collusion and copy attacks. Collusion attack tries to remove watermark by colluding video frames with the same watermark, whereas copy attack tries to embed a watermark to unmarked video. In digital cinema, copy attack can be performed to attack fingerprint watermarks by embedding a watermark that frames innocent user. Accurate watermark estimation, in terms of both polarity and energy, is an indispensable component to achieve effective WEAs, so they proposed the use of video frame dependent hash, called *frame hash*, as part of the embedded watermark. The original watermark is merged with the frame hash using a shuffling function

working based on a secret key to obtain the VFDW, which is then embedded to the content. Because of the frame hash, averaging method to estimate embedded watermark does not work. Collusion attack now results in degraded video and copy attack causes a distortion without successfully forging a watermark.

3.3.4 Digital Fingerprinting

Kundur and Karthik [27] proposed a method that combined the process of video fingerprinting and video encryption in order to construct an effective and efficient protection system. The idea is to encrypt the video with a key, which is the same for all users, and then send a set of slightly different decryption keys to users. The decryption process using many different keys would result in decrypted copies that are slightly different for each user. The difference between those copies would act as a forensic tracking mean. They used DCT to first process the raw data, and then the video is partially encrypted by sign-scrambling only a chosen subset of the resulting coefficients. Each user will receive the same encrypted content, but will be given a unique subset of keys for decrypting only a fraction of the encrypted coefficients. The locations and the sign bits of the remaining concealed subset are hidden from the receiving user and constitute the digital fingerprint in his copy. In order to achieve the robustness against collusion attack, they design a different set of common hidden encrypted coefficients for each combination of users, so that it can uniquely determine the exact colluding members when collusion attack happens. Using this method, they successfully cut down the amount of computation and the bandwidth requirement as the content needs to be encrypted once only and only one version of the content needs to be transmitted to all users. Nevertheless, their method is still susceptible to key

collusion attack and requires the video features being decrypted made known to users, making the encryption less secure.

Schonberg and Kirovski [46] proposed a phase-shifted spread-spectrum fingerprinting as a solution to the analog hole problem. They embed the fingerprint, defined as a spread-spectrum sequence of independent identically and uniformly distributed random samples, in the DCT domain of the video frames. For each coefficient, they consider the DCT coefficients with the same index from the neighboring DCT blocks within frames as well as within some preceding and succeeding frames, and compute the standard deviation of those coefficients to determine the magnitude of the fingerprint. The fingerprint will be then smoothly transitioned across those frames. In order to improve imperceptiveness, low frequency/high energy DCT coefficients are not marked. They also introduced the concept of *pilot fingerprints* for fast detection. Schonberg and Kirovski pointed out that the collusion resistance of their methods is constant, invariant to the content size. Nonetheless, their methods are only effective for collusion of very small size (1 or 2) and require a fingerprint that is sufficiently long. Additionally, they cannot resist the gradient attack.

Under the *Marking Assumption*, which says that by colluding users can only detect a mark if it differs in their copies and users cannot change the undetected marks without rendering the content useless, Boneh and Shaw [2] showed how to construct a *c-frameproof* code, a code that prevents the colluding users from framing an innocent user, and a *c-secure* code, a code that enables content provider to trace back an illegal copy to the source of piracy act in the presence of c users colluding, using error-correcting codes. For both kinds of codes, they first show a simple code satisfying the

desired property with length that is linear to the number of users, then together with an error-correcting code, it is used as an alphabet for the construction of new codes with shorter length. Boneh and Shaw also showed how to identify the colluding users when their codes are employed. Despite the effectiveness of their codes to deal with collusion attack, the length of those codes is still too large, which is polymonial to the maximum number of colluding users and logarithmic to the total number of users.

Trappe et al. [54] introduced the concept of *balanced incomplete block design* (BIBD) to construct an anti-collusion code with length equal to the square root of the number of users. The basic idea is to design a set of codewords such that each combination of codewords with certain size shares a unique subset of ones. They also proposed subgroup-based construction to decrease the computation requirement needed to identify colluders by grouping together users that are likely to collude into one group and assigning to each group a different anti-collusion code. As the result, it reduces the amount of computation and increases the detection statistics when colluders come from the same subgroup. However, this method decreases the ability to detect colluders from different subgroups. Since it is difficult to predict the correct way of grouping, this construction is not very useful. In spite of its shorter length, the code proposed by Trappe et al. only works in CDMA signaling and not in orthogonal signaling. They also assume that when fingerprints are averaged, the resulting message is the logical AND of those codewords, which is not true.

Another fingerprinting scheme which is based on error-correcting code was proposed by Ferrer and Joancomarti in [19]. Their embedding process starts with the compression of the content using JPEG algorithm. Every pixel in the compressed form will be compared to that of the uncompressed one in order to determine the

positions where marks will be embedded. The fingerprints will be encoded using an error-correcting code before being embedded to the content. The special marks are embedded only into pixels where the compressed and uncompressed contents differ. Detection process can be done easily by reversing the embedding process with the presence of the original content. Ferrer and Joancomarti showed that dual Hamming code can be used for encoding in the embedding process in order to deal with collusion attack involving two users. Although their method is relatively simpler, their fingerprinting scheme is not robust against random geometric distortions and combinations of basic image processing operations. Beside that, their method can only resist collusions of size two using a code of which length is linear to the number of users.

The other codes that have been used to deal with collusion attack are binary sorted code [32] and Reed Solomon code [55]. Lindkvist [32] showed that binary linear code and coset of binary linear code can only be used to resist collusions consisting of at most two users. She explained that for collusions of size larger than two, colluders can choose randomly an odd number of their codewords and then perform Modulo Two strategy to form another codeword which is not in the set of colluders' codewords. Modulo Two strategy is carried out by choosing the bit that appears an odd number of times at every position. She then proved that binary sorted code can be used as an alternative for handling collusion attack. Veerubhotla et al. [55] demonstrated how Reed Solomon code can be used to provide certain form of traceability by showing that given a word that is a linear combination of some codewords, we can determine the unique set of codewords used to construct the word efficiently. However, they also pointed out that if colluding members create an illicit

copy by making erasure in every detectable mark, it may be impossible to trace the colluders. Consequently, for tracing to be successful with high probability, the strategy chosen by colluders must be controlled, which is almost impossible to do in real life.

3.3.5 Other Related Works

Senoh et al. [47] addressed the inconvenience caused by many different DRM system employed by many different providers. User must install many different players to support many different file formats because those protection systems have no capability to inter-operate with each other. They proposed a new Intellectual Property Management and Protection (IPMP) method which supports inter-operability between those protection systems, while maintaining each of them individually. This method was proposed at ISO/IEC JTC1/SC29/WG11 (MPEG) in 2000 and the specification has been standardized as ISO/IEC 14496-1 Amendment 3 (MPEG-4 IPMP Extension), ISO/IEC 14496-13 (MPEG-4 IPMP), and ISO/IEC 13818-11 (MPEG-2 IPMP). This method requires content provider to send the protected content together with the IPMP information which tells users how the content is protected, what tools are needed to decode the protected content, and how to configure these tools to access and decode the content. If any of these tools are unavailable, IPMP information tells users the URLs where they can be downloaded or the necessary decoders can be delivered together with the content itself. This approach solves the inter-operability problem and makes it easier to renew a protection system. However, by telling users how the content is protected and how to decode it, it also tells pirates how to attack

the protection system more effectively. It also adds some overhead for the terminal to read and digest this IPMP information before it can access the content.

Embedding user-specific watermarks to the contents and appending user identities to the digital licenses, to certain extent, have affected user privacy. Conrado et al. [13] and Feigenbaum et al. [18] pointed out this privacy issue and explained how users can be annoyed by the rights purchase and content usage tracking done by the content provider. They suggested that rights issuing must be done anonymously. Conrado et al. proposed the use of secret security identifier (SSI), instead of user's public key, in license issuing process to conceal the real user identity. This SSI can be changed regularly to make tracking difficult. However, it results in a need to keep track all the SSI changes for all users, and therefore makes forensic tracking more difficult as well. Feigenbaum et al. suggested that in the process of content usage tracking, the content providers should collect only information that they really need and they should disclose how this information would be used. User privacy might seem to be irrelevant in the context of digital cinema, but we should not overlook the possibility of tracking done by pirates to obtain information about all contents a theater has access to and to create over time a pattern of theater's content usage.

Skraparlis [50] explained the use of message authentication codes (MAC) and digital signatures to protect the integrity of digital content. He explained a few ways to apply the hash function on the data blocks. Besides that, he also mentioned that labeling is more preferable than watermarking to be used as the medium of the authentication codes. Watermarking techniques are not chosen because its efficacy is unproven, it has relatively higher complexity, and it causes quality degradation. At the

same time, MAC does not have to be hidden imperceptibly as it is already protected by a cryptographic hash function.

Summary

The summary of all related works presented in this section is shown on the table below.

Table 1. Comparison among some existing protection systems used for digital video.

	Related Works	CP	AC	UR	FT	QS	E	S	R	UP	CS
DRM system	Liu et al. [34]	✓	✓	✓	✓	✓					
	Bloom [1]	✓	✓	✓	✓						
	Kirovski et al. [26]	✓	✓	✓	✓		✓		✓		
	Lin et al. [30]	✓	✓	✓	✓	✓	✓				
	Lin et al. [31]	✓	✓	✓	✓	✓	✓		✓		
	Linnartz et al. [33]	✓	✓	✓	✓	✓					
	Grimm & Aichroth [24]	✓	✓	✓	✓	✓	✓				
	Byers et al. [3]	✓	✓	✓	✓						
	Chong et al. [10]	✓	✓	✓	✓	✓					
	Chu et al. [11]	✓	✓	✓	✓	✓	✓	✓			
Encryption	Tosun & Feng [52]	✓	✓								
	Tosun & Feng [53]	✓	✓				✓				
	Liu & Li [35]	✓	✓								
	Lookabaugh & Sicker [36]	✓	✓				✓				
	Chiaraluce et al. [7]	✓	✓								
	Shieh [48]	✓	✓				✓				
	Zeng & Lei [57]	✓	✓								
Watermarking	Dittman et al. [16]		✓	✓	✓	✓					
	Wessely et al. [56]		✓	✓	✓						
	Cheng & Huang [5]		✓	✓	✓	✓	✓				
	Lubin et al. [38]		✓	✓	✓	✓					
	Lu et al. [37]		✓	✓	✓						
Fingerprinting	Kundur & Karthik [27]				✓		✓				
	Schonberg & Kirovski [46]				✓	✓					
	Boneh & Shaw [2]				✓						
	Trappe et al. [54]				✓						
	Ferrer & Joancomarti [19]				✓						
	Lindkvist [32]				✓						
	Veerubhotla et al. [55]				✓						
Misc.	Senoh et al. [47]							✓			
	Conrado et al. [13]		✓							✓	
	Feigenbaum et al. [18]			✓						✓	
	Skraparlis [50]			✓							

Note: CP - Concealment and Content Protection
AC - Access Control
UR - Content Usage Rights Management
FT - Forensic Tracking
QS - Quality of Service
E - Efficiency
S - Scalability
R - Renewability
UP - User Privacy
CS - Customer's Security

Observe that despite all different protection mechanisms they provide, all of them protect only the rights of content provider and none of them addresses the rights of the customers. We shall see in the next section how a failure in protecting customer's rights causes these protection schemes to be totally unfair to customers. In section 5, we shall present three solutions to this problem.

4. BUYER-SELLER WATERMARKING PROTOCOL

Encryption and access control scheme of a Digital Rights Management (DRM) system only protect the content from being illegally accessed by unauthorized users. They do not prevent an authorized user from illicitly reproducing the content. Moreover, no matter how robust and reliable the cryptosystem and the access control scheme are, all these mechanisms will be ineffectual when the movie is converted into analog signal and displayed on a movie screen. Regardless of all different kinds of protection systems being used, a digital movie eventually needs to be presented to the viewers in the clear, causing it to be unprotected and vulnerable to illegal copying. This problem, known as "the analog hole" problem, has been responsible for most of illicit copies available at large.

In order to fight against illegal copying, both copy protection and copy deterrence systems can be used as complimentary protection systems. Although copy

protection system, like a special hardware used for viewing and copying or an invisible watermark inserted to indicate number of copies allowed to be made, successfully prevents users from digitally copying the content files, it does not solve the analog hole problem and it is unable to help in identifying the copyright violator. Copy deterrence system, on the other hand, is achieved by a mechanism that chains the identity of each user to the copy of content he owns. A user-specific distinct watermark, called digital fingerprint, is embedded into each copy of the films that content provider distributes. This mechanism discourages users from performing unauthorized duplication and distribution. Simultaneously, it provides a forensic-tracking mean for content provider. Whenever an illicit copy is found, the origin of the copy can be determined by extracting the unique watermark embedded in the copy. Knowing that any protection systems can never guarantee a perfect security at all times, it is very important to include this tracking mechanism in the system.

Nevertheless, digital fingerprinting only supplies right protection to content provider and does not protect the rights of customers at all. The consequences of this unfairness are elaborated in the following subsection, followed by the buyer-seller watermarking concept to solve the problem and overview to works having been done in the two subsequent subsections.

4.1 Customer's Right Problem

A digital fingerprinting scheme is, in the first place, designed to protect the copyright of a content provider, and not to protect that of the customers. In all fingerprinting schemes, it is always assumed implicitly that the content provider is honest and trustworthy [44], whereas customers are always deemed as highly potential source of

piracy acts. As the result, every scheme gives the content provider a full control over the fingerprinting process. Fingerprint generation, insertion, and detection are solely done by content provider; no other party is involved in any of those processes.

Unfortunately, the assumption on seller's reliability and honesty may not always hold in real life, causing all fingerprinting schemes to be biased and unfair to customers. The following situations show what harm this assumption can do to a lawful customer:

- **False implication**

  Suppose after sending a fingerprinted copy of a digital content to user U, the content provider unintentionally inserts the fingerprint generated specifically for user U into the copy sent to another user, let's say user V. Assume that V is malicious user and illegally reproduces and redistributes the content. Later, when content provider finds an illicit copy distributed by V, instead of admitting his mistake that he used the same fingerprint for two different users, he can choose to accuse user U of a piracy act since the fingerprint found in the illegal copy matches the one in the copy user U has. User U has no way to prove his innocence as the evidence does not side him and he does not know about the mistake done by content provider.

- **Framing by content distributor**

  Assume that content provider hires an agent A to distribute the digital content he produces and agent A will pay the royalty fee on per-copy basis. Legally, agent A must sell different copies to different users. Nonetheless, in order to maximize his profit, agent A can choose to sell the same copy to many different buyers, let's say user U is one of the buyers. Later, agent A will report to content provider that he

only sold one copy to user U. It does not really matter whether the other buyers illegally distribute their copies or not. Once content provider discovers the existence of their copies, user U will be implicated and sued for illegal redistribution, even though he did not do it. Again, he cannot deny the accusation since the evidence spells his name as the culprit and he has no idea about the unlawful act of agent A.

- **Framing by content provider**

  Because the fingerprint generation process is completely controlled by content provider, he knows the exact fingerprint inserted to the copy that each customer receives. Therefore, he has no difficulty in reproducing the exact fingerprinted copy that a particular user receives. Assume that content provider is malicious and he has sold a copy of certain digital content to user U. In order to get a good amount of money in a very easy way, content provider can reproduce copies of the same content containing fingerprint of user U and distribute them. Consequently, he can charge user U for illegal distribution and ask for compensation from him. The same as the two previous cases, user U has no way to refute the accusation for his unique fingerprint is found in an illegal copy.

It is very clear from the three cases that due to the assumption on seller's honesty, the rights and interests of customers are left unprotected, which potentially causes a legitimate customer to bear the punishment of a deed that he did not do. This condition where customer's rights are unprotected and vulnerable to framing attack defines the customer's right problem.

Beside false implication and framing, the worst consequence of customer's right problem is that it nullifies the objective and the purpose of fingerprinting itself. Once customers learn about this specific problem, it can cause an irresolvable dispute. Imagine a situation where content provider performs every transaction legally, but there is a malicious customer who redistributes the digital content he has. Content provider can actually bring the matter to the court and sue this particular user for an act of piracy. However, now this malicious user can deny his unlawful act and point his finger at content provider by claiming that the illicit copy was produced by the content provider. He can argue that content provider knows the exact fingerprint inserted into his copy, and therefore content provider can reproduce the copy he owns effortlessly. When it happens, content provider will have no proof to establish the truth and the guilty user is able to escape from the consequence of his act. In other word, the forensic tracking mechanism is made void.

4.2 Description and Requirements

Customer's right problem in the traditional fingerprinting schemes was first brought up to the surface by Qiao and Nahrstedt [44] in 1998. However, their protocols did not effectively solve the problem. It was the protocol proposed by Memon and Wong [39] later in the same year that first successfully solved the customer's right problem. From that moment on, every protocol designed to address customer's right problem is named after the name of Memon and Wong's protocol, *Buyer-Seller Watermarking Protocol*. The overview of those two works are presented in the next subsection, whereas the details can be found in [44] and [39][40].

A Buyer-Seller Watermarking Protocol is a protocol that incorporates techniques of watermarking and fingerprinting to protect the rights of both the buyer (customer) and the seller (content provider) [23].

The underlying idea of a buyer-seller watermarking protocol is to insert into the digital content to be distributed another special mark, besides the normal digital fingerprint, that both content provider and customer have no full knowledge of. Instead of letting content provider completely control the generation of this mark, both content provider and customer take part in the process and each contributes a part of the mark produced. However, content provider knows nothing about the part created by customer, and vice versa. Therefore, none of them knows the exact mark being inserted into the content.

Content provider not knowing the exact watermarked copy that a customer receives implies that he cannot reproduce copies of the original content containing the customer's watermark, and thus he cannot falsely accuse an innocent customer of a piracy act. On the other hand, content provider is still able to identify the source of an unlawful act from the fingerprint and watermark found in unauthorized copy, and then prove it to a third party without having to worry about customer claiming that the illicit copy may be originated from him. At the same time, the fact that customer does not know the exact watermark inserted guarantees that he cannot remove it from the content he receives. It is clear that in a buyer-seller watermarking protocol, neither content provider nor customer is assumed to be honest and trustworthy.

Besides providing a robust forensic tracking mean and preventing framing, there are some other requirements that a buyer-seller watermarking protocol should satisfy. These requirements often measure the performance of a protocol, so satisfying

all of them will be the ideal situation. However, satisfying one requirement often means refutation of some other requirements, making it difficult to provide them all. The requirements of a buyer-seller watermarking protocol are listed below. The list of requirements is compiled from [8][9][23][25][29].

- **Traceability**

  A watermarking protocol should enable content provider to trace a piracy act to its source. In other words, content provider should be able to identify customers who duplicate and redistribute their contents illegitimately.

- **No Repudiation**

  A watermarking protocol should prevent guilty customers from denying their unlawful act. A buyer accused of illegal copying should not be able to claim that the unauthorized copy may be produced by content provider or a security breach of his system. This requirement provides content provider's security.

- **No Framing**

  A watermarking protocol should eliminate the possibility of accusing an innocent customer. Neither malicious content provider nor other customers should be able to run away from the consequence of their violations by pushing the blame to an honest customer. Customer's security is assured by this requirement.

- **Collusion Resistance**

  A watermarking protocol should not enable a coalition of customers to locate, delete, or fabricate the special mark embedded by comparing their copies. Even though they have access to certain number of watermarked copies, they should not be able to find the mark and recover the original content.

- **Anonymity**

  A watermarking protocol should allow customers to purchase a digital content without having to expose their identity to the content provider.

- **Unlinkability**

  A watermark protocol should prevent content provider from recording the purchase history of a customer. Given two different watermarked contents, it should be infeasible to deduce if they are purchased by the same customer.

- **No Additional Trusted Third Party**

  Besides an arbiter and certification authority (CA), a watermark protocol should not require the involvement of a trusted third party (TTP) in any stage of the process. Buyer-seller watermarking protocol was first introduced to eliminate the assumption on seller's honesty, therefore it is unreasonable to introduce another participating party, other than arbiter and CA, whose honesty is assumed. The assumption on arbiter's and CA's honesty is acceptable since it also exists in the original situation, i.e. in the traditional fingerprinting and watermarking schemes. Hence, having this assumption does not make a buyer-seller watermarking protocol inferior to traditional fingerprinting and watermarking schemes.

- **No Unbinding Problem**

  A watermark protocol should provide a mechanism to bind a generated watermark to the specific digital content it is inserted, and thus prevent content provider from transplanting a watermark detected in a pirated copy into other copies of (possibly higher-priced) digital contents in order to get more compensation. This unbinding problem was first discovered by Lei et al. [29].

- **Customer's Convenience**

  A watermark protocol should not hinder customers from purchasing a digital content by the inconvenience it causes. It is important to minimize the amount of computation required to purchase a digital content. Customers should not be burdened by a heavy computation. Neither should they be required to communicate with many parties in a single transaction. In some cases, it is also good to exempt customers from participating in dispute resolution process. Moreover, due to the number of contents a buyer could purchase, a watermark protocol should enable customers to decrypt many different contents using a single key. Thus, customers do not have to maintain a list of keys needed to decrypt all contents they purchased.

4.3 Existing Solutions

In order to address the customer's right problem, Qiao and Nahrstedt [44] proposed two watermarking protocols which are based on non-invertible watermarking scheme. The first protocol, called TTP watermarking protocol, depends heavily on a trusted third party to perform watermark generation and embedding. Content provider and customer do not directly communicate to each other. Every transaction is done with TTP as their middleman. Content provider sends the original content to TTP for watermarking. TTP encrypts the original content using DES and uses the ciphertext as the watermark. This ciphertext is embedded into the content and the watermarked content is sent to the customer. Realizing the heavy burden a TTP has, Qiao and Nahrstedt proposed the second protocol, called Owner-Customer watermarking protocol. In this method, customer generates a random sequence by encrypting a bit

sequence mutually agreed between customer and owner, and then sends it to the owner. Content provider encrypts this sequence using DES and embeds the ciphertext into the content as a watermark and sends the watermarked content to the customer encrypted using the random bits he generated earlier. As only customers know the key used to generate the random bits, all legal customers now have evidence to prove their rights on the content. However, these two methods do not solve the customer's right problem since the content provider knows exactly each watermark embedded to the customer's copy, and therefore he can reproduce the same watermarked copy and redistribute it. As the result, content provider can frame innocent users by accusing them of a piracy act.

The Buyer-Seller watermarking protocol proposed by Memon and Wong [39][40] is the first method that solved the customer's right problem. They successfully designed a protocol that prevents both content provider and customer from knowing the exact watermark being embedded to the content. Their protocol requires a trusted third party, called Watermark Certification Authority (WCA), to generate the watermarks on customer's behalf. In their protocol, transaction starts with a request for a watermark from buyer to WCA. Memoryless WCA generates a random watermark, encrypts it using customer's public key, and transmits it to customer. Customer will then send this encrypted watermark to content provider. Content provider first produces a fingerprint, unique to each customer, and inserts it into the content in order to enable him to identify each copy sold. He will then generate a random permutation function to permute the encrypted watermark received from customer. This encrypted and permuted watermark will be inserted to the encrypted content as a second watermark. This can be done due to the use of public

key cryptosystem that is privacy homomorphic with respect to watermark insertion operation. The encrypted watermarked content will be then transmitted to the requesting buyer. By inserting the watermark in encrypted form, seller does not know the exact watermarked copy that buyer receives, thus he cannot create copies of the original content containing the buyer's watermark. On the other side, content provider still can identify the buyer of an unauthorized copy from the fingerprint found in it. The most undesirable feature of this protocol is the requirement of a trusted and reliable WCA. WCA is required in order to ensure that the watermark used in each transaction is not approximately invariant to permutation. However, without an assumption on its honesty, it is possible that WCA colludes with either seller or buyer to frame the other party.

Due to the success Memon and Wong achieved in solving customer's right problem, their protocol became the foundation of many other protocols proposed after theirs. Some variants of Memon and Wong's protocol can be found in [6][9][17][23][25].

Cheung and Curreem [6] modified Memon and Wong's protocol by introducing the concept of *watermark certificate* and accommodating ownership transfer of sold contents. A watermark certificate produced by WCA consists of encrypted watermark, the encryption key, and digital signature of them signed by WCA. They claimed that it is used in order to prevent the encrypted watermark of a user to be used by another user, who had sold a digital content to the user, in some other transaction with content provider. In Cheung and Currem's protocol, when a customer wants to buy a digital content from other customer, the buying customer sends his watermark certificate to the selling customer. The selling customer will then

forward his watermarked content and the watermark certificate to the content provider. Content provider will produce a new watermarked content carrying buying customer's watermark and send it to the selling customer, followed by selling customer forwarding it to the buying customer. Even though it is claimed to be useful, the concept of watermark certificate is actually redundant. In Memon and Wong's protocol itself, the encrypted watermark of a user cannot be used by another user because only that particular user knows the corresponding secret key, another user will not be able to decrypt the encrypted content without this secret key. Additionally, the transfer of ownership is not a desirable feature for content provider. Therefore, assuming the willingness of content provider to be involved in the process is not realistic.

Ju et al. [25] introduced the use of a pair of one-time anonymous public and private keys in order to provide buyer's anonymity and transaction unlinkability. The identity of a customer will only be revealed by WCA when he is involved in an illegal redistribution. Moreover, they do not require customers to be involved in the dispute resolution process. Instead, customers need to send their private key encrypted using a judge's public key to WCA, so that the judge can access it whenever dispute resolution is considered necessary. However, it means that the judge that will be act as an arbiter must be decided before any transaction and take part in the watermark generation protocol. No other judge will later be able to help to resolve the dispute. It also implies that the honesty of judge is assumed and the possibility of WCA colluding with the judge to betray either seller or buyer is ignored. Beside that, trusting WCA to keep customer's identity and their private keys is not a very good

idea. It is a single point failure that once it is compromised, the security system will be torn down.

Choi et al. [9] addressed the issue of possible collusion among content provider, WCA, and judge in Ju et al.'s protocol. They modified Memon and Wong's protocol by changing its watermark generation protocol with theirs. In their method, WCA must generate a number of watermarks for a customer to choose. The concept of commutative cryptosystem is applied in order to conceal the watermark chosen by customer from WCA. They also use anonymous pair of public and private keys to provide user's anonymity and unlinkability. Choi et al. undo the changes made by Ju et al. in dispute resolution protocol and restore it to that of Memon and Wong's protocol, so that arbiter can be appointed only when it is necessary and no judge is involved in watermark generation protocol. Even though they successfully eliminate the possibility of collusion between judge and the other parties, but honesty of WCA is still assumed. WCA knows the true identity of customers and by colluding with seller the chosen watermark can be recovered. It is done by comparing the encrypted form of every watermark offered to customer to the one that seller keeps for that particular customer. So, other than anonymity and unlinkability, this protocol has the same properties as those of Memon and Wong's.

Goi et al. [23] provided the security analysis for Ju et al.'s and Choi et al's protocols, followed by presenting their remedy to those problems in their work. They eliminate the possible involvement of WCA in a collusion by letting the customer to generate his own watermark. However, they forgot that it may threaten seller's security as customer may produce watermark which is invariant to permutation. Therefore, it defeats the main purpose why the concept of WCA is introduced in the

first place. Goi et al. also suggested that customers certify their anonymous key pairs to certificate authority (CA), which is definitely trustable, instead of WCA.

Emmanuel and Kankanhalli [17] explained the use of Memon and Wong's buyer-seller protocol in the context of video broadcast. First, broadcaster will produce a masked video by blending an opaque mask frame onto the original video. The same masked video will be sent to all subscribers. The buyer-seller protocol will be then applied to obtain subscriber's watermark, so that the unmasking frame can be tailored uniquely for each subscriber. The unmasking frame received by each subscriber is actually the masking frame subtracted by the broadcaster-generated fingerprint and the subscriber's watermark. Thus, when unmasking process is done, the content will be automatically fingerprinted and watermarked. Again, the major weakness of this method is the requirement of trusted WCA. Besides that, they suggested to use Niederreiter public-key cryptosystem that is privacy homomorphic with respect to addition in order to enable unmasking-frame production without broadcaster knowing the exact watermark being embedded. This cryptosystem adds too much redundancy to the ciphertext and causes a severe blow up in the size of the ciphertext. They mentioned that for plaintext of size 32 bits, it will result in a ciphertext of length 370 bits, which means more than ten times of the length of the plaintext. In their protocol, the unmasking frame, which is as big as the video to broadcast, must be sent in encrypted form. As the result, the bandwidth required for sending the unmasking frame is simply too large.

Chang and Chung [4] claimed that Memon and Wong's protocol cannot withstand man-in-the-middle attack because content provider never provides his private information to convince customer that he is the genuine content provider.

Hence, they proposed a protocol where content provider uses a pair of private and public keys similar to those in El Gamal cryptosystem to control the generation and verification of the embedded watermark. In their protocol, customer generates his own watermark and then permutes it using a one-way permutation function before sending it to the provider. This permuted watermark will be combined with fingerprint generated specifically for the customer using content provider's private key to produce a new watermark. The resulting watermark will be then inserted to the content and the watermarked content will be transmitted to the customer. However, their effort and idea are not very useful because their claim about the Memon and Wong's protocol is not true in the first place. Memon and Wong assumed secure authentication before the protocol starts, and thus the two parties can identify themselves to each other. In addition, the permutation function used in the watermark embedding process is only known by the content provider. So, it is clear that we do not need another kind of private key to control the watermark generation. The worst thing about Chang and Chung's protocol is the fact that their modification makes void the protection against false implication as content provider has now full knowledge about the exact watermark inserted, and therefore defeats the main objective of the interactive protocol.

Another variant of Memon and Wong's work is Lei et al.'s work [29] that spotted *unbinding problem* in all protocols proposed earlier, including Memon and Wong's. Unbinding problem is caused by failure to provide proper mechanism to bind a generated watermark to the specific digital content it is inserted. This problem enables content provider to transplant a watermark detected in a pirated copy into other copies of (possibly higher-priced) digital contents and get more compensation.

They tackle this problem by requiring seller and buyer to set up a common agreement specific for a particular content that will be involved in the transaction. Once agreed, it is now content provider, not customer, who will request for a watermark to WCA. WCA will send back the generated watermark encrypted using customer public key to keep seller in the dark about the inserted watermark. WCA is also asked to produce the signature of the watermark and the agreement in order to explicitly bind these two data. As buyer has no knowledge about the watermark, seller does not need to permute it and he can directly embed it together with a fingerprint into the content in encrypted domain. Consequently, the watermarking employed need not be *linear*. Buyer will receive the watermarked content in encrypted form. In this protocol, customer only needs to communicate with seller and nobody else during the transaction. Moreover, he is not involved in dispute resolution protocol as judge asks WCA, instead of buyer, to reveal the watermark. Nonetheless, the assumption on the honesty of WCA is still a must to prevent a conspiracy between WCA and seller. Moreover, in this protocol, content provider can cheat by sending a random key, instead of customer's public key, to WCA. WCA will use the key to encrypt the watermark. By using the corresponding decryption key, content provider will have no problem in recovering the watermark generated. In other word, customer's right problem is unsolved.

Choi and Park [8] showed how the idea of buyer-seller protocol can be applied in multiple-purchase environment and how it can be adjusted to accommodate mobile communications with limited computing resources. They used a concept similar to El Gamal cryptosystem to achieve a protocol which needs only one decryption key for deciphering multiple contents encrypted using many different keys. However, their

protocol requires customer to do all purchases at one time, making it a bit unrealistic. The assumption on the honesty and reliability of WCA is still needed as well. To enable buyer-seller protocol on mobile communications, Choi and Park introduced the use of mobile agent, which will perform most of the computation steps on behalf of customers. They shift the work from customers to this mobile agent. Unfortunately, as the side effect of this addition, we now have one more party that is assumed to be trustworthy.

Summary

Please refer to the following table for the comparison among all existing solutions discussed in this section.

Table 2. Comparison among all existing buyer-seller watermarking protocols.

Requirements	**Existing Solutions**									
	[44]	[40]	[6]	[25]	[9]	[23]	[17]	[4]	[29]	[8]
Traceability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
No Repudiation	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
No Framing	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Collusion Resistance	No	No	No	No	Yes	No	Yes	No	No	Yes
Anonymity	No	No	No	Yes	Yes	Yes	No	No	Yes	Yes
Unlinkability	No	No	No	Yes	Yes	Yes	No	No	Yes	Yes
No Additional TTP (WCA)	Yes	No	No	No	No	Yes	No	Yes	No	No
No Unbounding Problem	No	No	No	No	No	No	No	No	Yes	No
Customer's Convenience										
• Not watermark generator	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
• Number of parties to communicate with	1	2	2	2	2	1	2	2	1	2
• No participation in dispute resolution	No	No	Yes	Yes	No	No	No	Yes	Yes	Yes
• Single decryption key in multiple purchases	No	No	No	No	No	No	No	No	No	Yes

Note: [44] refers to Qiao and Nahrstedt's Owner-Customer Watermarking Protocol, which is better than their TTP Watermarking Protocol.

It is shown on the above table that all existing solutions truly depend on an additional trusted third party to solve the customer's right problem. The existing protocols that do not require the participation of a WCA fail to solve the problem, which is indicated in their failure to satisfy either no repudiation or no framing requirements. In the next section, we shall see how customer's right problem can be successfully solved without having to involve any additional trusted third party.

5. PROPOSED SOLUTIONS

All existing solutions to customer's right problem rely on the trustworthiness of Watermark Certification Authority (WCA) as a party who generates a valid watermark for every transaction. WCA is required in those solutions to ensure that the watermark used in each transaction is not approximately invariant to permutation. Otherwise, it will be possible for customer to perform a brute-force attack in order to figure out the permuted watermark, and thus remove it from the copy he received from content provider. Although those protocols assume that WCA is memoryless, it is almost impossible for us to assume that WCA does not have the full knowledge of the watermark used in each transaction. As the result, there is a possibility that WCA colludes with either content provider or customer to betray the other party. In order to avoid this situation, they assume that WCA is honest.

However, as we have seen earlier, introducing a new trusted third party is not the best option because buyer-seller watermarking protocol was, in the first place, invented to eliminate an assumption on seller's honesty.

In order to address this issue, we propose three buyer-seller watermarking protocols that do not require the participation of other trusted third party besides the

arbiter and certification authority (CA). We shall see in this section how we can actually remove the requirement of a watermark certification authority without ignoring the reasons it was introduced. On the other hand, it is totally acceptable to assume that arbiter and CA are honest since this assumption does exist in the traditional fingerprinting and watermarking schemes. Moreover, CA is the issuer of public key certificates in public-key cryptosystem infrastructure, so it is definitely trustable. Otherwise, no public-key cryptosystem would be secure and no public and private key pair would be binding or confidential [23].

Before we start elaborating our protocols, let us first introduce the notations that will be used in the explanation of those protocols.

5.1 Notations and Assumptions

In the model of the proposed protocols, four different roles involved are as follows:

1. **S** : the seller, content provider who wishes to make a profit on the sales of digital contents he produces.
2. **B** : the buyer, customer who purchases copies of the digital contents from S.
3. **CA** : a trusted certification authority who is responsible for issuing public-key certificates to all parties involved in the protocols.
4. **J** : the judge, an arbiter who adjudicates lawsuits against the infringement of copyright and intellectual property.

The notations are defined as follows:

X	The original unwatermarked copy of a digital content.
V	A digital fingerprint generated by seller specifically for each buyer.
W	The watermark to be inserted to the content.

X'	The fingerprinted copy of the content.
X''	The fingerprinted and watermarked copy of the content, which is delivered to the buyer.
$\oplus$	The watermark insertion operation.
(pk_I, sk_I)	A public-private key pair of individual *I*. The public key is denoted by pk_I, whereas sk_I denotes the private key.
$E_{pk_I}(M)$	The ciphertext of message *M* encrypted using *I*'s public key.
$D_{sk_I}(C)$	The plaintext of ciphertext *C* decrypted using *I*'s private key.
$Sign_{sk_I}(M)$	The signature of message *M* signed by *I* using his private key.
(pk_H, sk_H)	A public-private key pair of a homomorphic public-key cryptosystem.
$E_{pk_H}(M)$	The ciphertext of message *M* encrypted using a homomorphic public-key cryptosystem.
$D_{sk_H}(C)$	The plaintext of ciphertext *C* decrypted using a homomorphic public-key cryptosystem.

In our protocols, we assume that public-key infrastructure has been established and each party involved has already had his own public-private key pair as well as a digital certificate issued by **CA**. Therefore, before each transaction, all parties involved are able to authenticate each other and communication between any two parties can be done in a secure manner.

We also assume the existence of a public key cryptosystem that is *privacy homomorphic* with respect to the watermark insertion operation $\oplus$. A cryptosystem is

a *privacy homomorphism* with respect to operation $\oplus$ if and only if it has the property that

$$E_k(m_1 \oplus m_2) = E_k(m_1) \oplus E_k(m_2)$$

for any m_1 and m_2 in the message space and for any k in the key space [40]. So, by interchanging the encryption and insertion operation, the result will still be the same. This property enables us to insert a watermark in the encrypted domain. Please refer to Section 6.1 for some instances of such cryptosystem.

Another assumption we make is that every message exchanged between any two parties includes a timestamp and nonce, like in Emmanuel and Kankanhalli's protocol [17]. A timestamp indicates the generation and expiration time of the message, whereas nonce is a random number that has to be unique within the time span indicated by the timestamp. Nonce is used in order to prevent replay attack. However, they will not be written explicitly for the sake of clarity.

5.2 Memon and Wong's Buyer-Seller Watermarking Protocol without Watermark Certification Authority

The first protocol that we propose is a variant of Memon and Wong's buyer-seller watermarking protocol [39][40]. We modify Memon and Wong's protocol by removing the Watermark Certification Authority (WCA) role and shifting the task of generating watermark to the buyer. Hence, a customer must generate his own watermark for each purchase he makes. In order to prevent customers from generating a watermark which is invariant to permutation, content provider needs to check the validity of the watermark sent by customer and he can reject it if it is invalid.

This protocol consists of three subprotocols, they are *content-watermarking protocol*, *copyright violator identification protocol*, and *dispute resolution protocol*. The detail of each subprotocol is presented below.

5.2.1 Content-Watermarking Protocol

Let **B** be the customer wanting to purchase a copy of content *X* from **S**.

1. Buyer **B** generates a watermark $W = (w_1, w_2, \ldots, w_n)$ specifically for this transaction.
2. Buyer **B** chooses a public-private key pair (pk_H, sk_H) for the homomorphic cryptosystem, and then computes $Sign_{sk_B}(pk_H)$.
3. Buyer **B** encrypts W with pk_H to obtain

$$E_{pk_H}(W) = \left(E_{pk_H}(w_1), E_{pk_H}(w_2), \ldots, E_{pk_H}(w_n)\right),$$

   and then signs it using his private key sk_B to get $Sign_{sk_B}\left(E_{pk_H}(W)\right)$.
4. Buyer **B** sends pk_H, $Sign_{sk_B}(pk_H)$, $E_{pk_H}(W)$, and $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ to **S**.
5. Seller **S** verifies the signature of encrypted watermark $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ by checking if $E_{pk_B}\left(Sign_{sk_B}\left(E_{pk_H}(W)\right)\right)$ is equal to $E_{pk_H}(W)$. If they are equal, **S** continues with the next step, otherwise the transaction is cancelled. In the same way, **S** also verifies the encryption key pk_H and its signature $Sign_{sk_B}(pk_H)$.
6. Let $b_1, b_2, \ldots, b_p \in \{0,1\}^k$ be all the different blocks in a string $U \in \{0,1\}^{qk}$, $k > 0$, $0 < p \le q$, and each b_i occurs c_i times, $1 \le c_i \le q$, in U. Define a function *perm* as follows:

$$perm(U) = \frac{\left(\sum_{i=1}^{p} c_i\right)!}{\prod_{i=1}^{p} c_i!} = \frac{q!}{\prod_{i=1}^{p} c_i!}$$

Seller **S** computes $perm\left(E_{pk_H}(W)\right)$ to get the number of different permutations to which $E_{pk_H}(W)$ is not invariant, i.e. the number of permutations σ such that $\sigma\left(E_{pk_H}(W)\right) \neq E_{pk_H}(W)$. Observe that $perm\left(E_{pk_H}(W)\right)$ also indicates the number of permutations to which W is not invariant. It is because every encryption function is injective, i.e. for all messages x and y, $x = y \Leftrightarrow E_{pk_H}(x) = E_{pk_H}(y)$.

7. Seller **S** checks the validity of watermark W by comparing the number of different permutations to which $E_{pk_H}(W)$ is not invariant, to a threshold δ_{perm}. This threshold is used by **S** to ensure that the watermark W presented by **B** is not approximately invariant to permutation, i.e. the number of permutations σ such that $\sigma(W) \neq W$ is large enough, so that it is infeasible for **B** to perform a brute force attack to guess the permutation that will be used by **S** in step 9. If $perm\left(E_{pk_H}(W)\right) \geq \delta_{perm}$, then **S** continues with the next step. Otherwise, **S** rejects watermark W.
8. Seller **S** generates a fingerprint V, which is unique for each customer, and then inserts it into the original copy of the digital content X to get a fingerprinted copy $X' = X \oplus V$.
9. Seller **S** chooses a random permutation σ, and uses it to permute the elements of the encrypted watermark $E_{pk_H}(W)$. In other words, **S** computes

$$\begin{aligned}\sigma\left(E_{pk_H}(W)\right) &= \sigma\left(E_{pk_H}(w_1), E_{pk_H}(w_2), \ldots, E_{pk_H}(w_n)\right) \\ &= \left(E_{pk_H}\left(w_{\sigma(1)}\right), E_{pk_H}\left(w_{\sigma(2)}\right), \ldots, E_{pk_H}\left(w_{\sigma(n)}\right)\right) \\ &= E_{pk_H}\left(w_{\sigma(1)}, w_{\sigma(2)}, \ldots, w_{\sigma(n)}\right) \\ &= E_{pk_H}\left(\sigma\left(w_1, w_2, \ldots, w_n\right)\right) \\ &= E_{pk_H}\left(\sigma(W)\right).\end{aligned}$$

This equation $\sigma\left(E_{pk_H}(W)\right) = E_{pk_H}\left(\sigma(W)\right)$ is true as $E_{pk_H}(W)$ is of the form $\left(E_{pk_H}(w_1), E_{pk_H}(w_2), \ldots, E_{pk_H}(w_n)\right)$, and thus interchanging encryption and permutation operations will give us the same result.

10. Seller **S** inserts the permuted watermark into the fingerprinted content X' in encrypted domain. In other words, **S** first computes $E_{pk_H}(X')$, and then inserts the permuted encrypted watermark to it, to obtain the encrypted and watermarked content X''.

$$\begin{aligned}E_{pk_H}(X'') &= E_{pk_H}(X') \oplus E_{pk_H}\left(\sigma(W)\right) \\ &= E_{pk_H}\left(X' \oplus \sigma(W)\right).\end{aligned}$$

11. Seller **S** sends $E_{pk_H}(X'')$ to buyer **B**.
12. Seller **S** stores identity of buyer **B**, ID_B, pk_H, $Sign_{sk_B}(pk_H)$, $E_{pk_H}(W)$, $Sign_{sk_B}\left(E_{pk_H}(W)\right)$, V, and σ as one entry in Table$_X$. Table$_X$ contains one entry for each copy of X that **S** sells.
13. Buyer **B** decrypts the encrypted content he receives from seller **S** using the corresponding decryption key sk_H to obtain the watermarked content X''. That is **B** computes

$$D_{sk_H}\left(E_{pk_H}(X'')\right) = X'' = X \oplus V \oplus \sigma(W).$$

Please refer to figure 4 for the idea underlying this content-watermarking protocol.

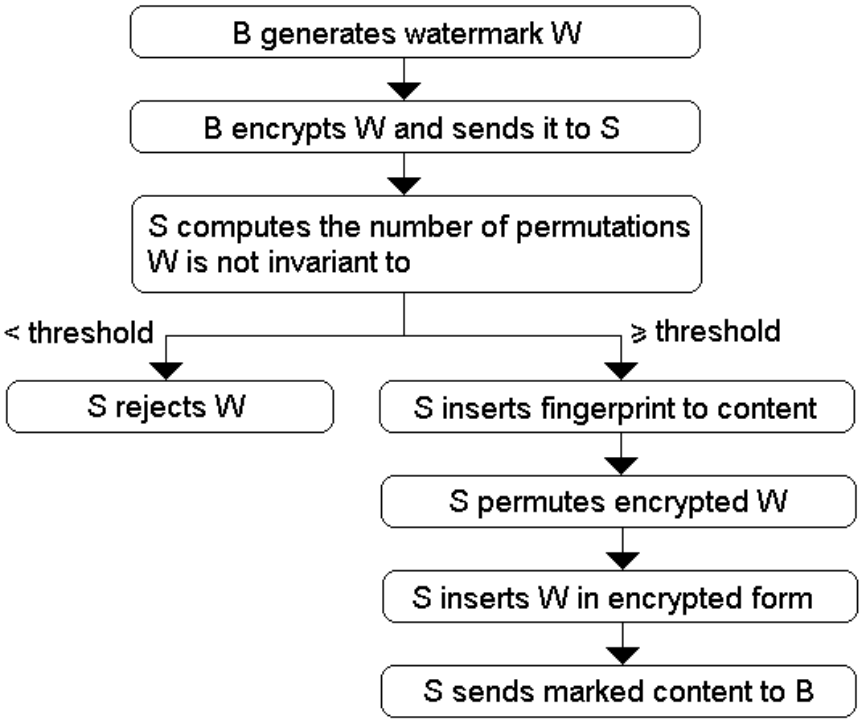

Figure 4. Content-watermarking protocol of the first protocol.

5.2.2 Copyright Violator Identification Protocol

1. When seller **S** discovers an authorized copy of content *X*, say *Y*, he extracts the unique fingerprint embedded in *Y* using the watermark extraction function *D*, which takes both *X* and *Y* as its input. Let $V_{FOUND} = D(X, Y)$ be the fingerprint detected in *Y*.
2. Seller **S** correlates V_{FOUND} with every fingerprint stored in Table_X in order to find the one with the highest correlation beyond a confidence threshold. Let V_{MAX} be the fingerprint that has the highest correlation with V_{FOUND}. If fingerprint V_{FOUND} cannot be matched to any fingerprint in Table_X, then the protocol fails.
3. Seller **S** retrieves all the information that corresponds to fingerprint V_{MAX} from Table_X. The information includes the identity of buyer, say ID_B, his encrypted

watermark and its signature, $E_{pk_H}(W)$ and $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ respectively, the encryption key of the homomorphic cryptosystem and its signature, pk_H and $Sign_{sk_B}(pk_H)$ respectively, and permutation σ.

Once seller **S** has the identity of buyer from whom the unauthorized copy was originated, **S** can appoint a judge **J** and proceed with dispute resolution protocol.

5.2.3 Dispute Resolution Protocol

Let **J** be the judge appointed by **S** to resolve the dispute between him and buyer **B**.

1. Seller **S** sends Y, ID_B, $E_{pk_H}(W)$, $Sign_{sk_B}\left(E_{pk_H}(W)\right)$, pk_H, $Sign_{sk_B}(pk_H)$, and σ to judge **J**.
2. Judge **J** verifies the signature of encrypted watermark $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ by checking if $E_{pk_B}\left(Sign_{sk_B}\left(E_{pk_H}(W)\right)\right)$ is equal to $E_{pk_H}(W)$. If they are equal, **J** continues with the next step, otherwise the case is dropped. In the same manner, **J** also verifies the encryption key pk_H and its signature $Sign_{sk_B}(pk_H)$.
3. Judge **J** sends $E_{pk_H}(W)$ to buyer **B**.
4. Buyer **B** decrypts $E_{pk_H}(W)$ using the corresponding private key sk_H to obtain $W = D_{sk_H}\left(E_{pk_H}(W)\right)$.
5. Buyer **B** sends W to judge **J**.
6. Judge **J** verifies W by encrypting it using key pk_H, and then comparing the result to $E_{pk_H}(W)$ he received from **S**. If they are equal, **J** goes on with the next step. Otherwise, **B** is found guilty.

7. Judge **J** computes the permuted watermark $\sigma(W)$ and checks its existence in *Y*. If $\sigma(W)$ is detected in *Y*, **B** is declared guilty. Otherwise, **B** is deemed innocent.

5.3 Bi-permutation Buyer-Seller Watermarking Protocol

The first protocol requires customers to generate the watermark used in every transaction, whereas content provider only needs to permute the generated watermark. Considering the limited resources that customers have and the inconvenience caused, this protocol may hinder costumers from purchasing the digital content. In order to address this issue, we swap the tasks that content provider and customer must perform in our second protocol. As content providers, in general, have more computing resources and power than customers, it is more reasonable to have content providers do more work than customers.

In this protocol, the watermark to be inserted is created by the content provider. The watermark will be then permuted twice, once by each party, in order to prevent both parties from acquiring the full knowledge of the watermark inserted. First, customer performs bit permutation on each element of the generated watermark to conceal it from the content provider. Consecutively, content provider will perform block permutation on the bit-permuted watermark to prevent customer from knowing the exact watermark inserted. The use two kinds of permutations explains why this protocol carries the term *bi-permutation*.

Bi-permutation buyer-seller watermarking protocol also consists of the same three subprotocols: content-watermarking protocol, copyright violator identification protocol, and dispute resolution protocol. The detail of each subprotocol is presented below.

5.3.1 Content-Watermarking Protocol

Let **B** be the customer wanting to purchase a copy of content *X* from **S**.

1. After receiving a request from buyer **B**, seller **S** generates a watermark $W = (w_1, w_2, \ldots, w_n)$ specifically for this transaction. Then, **S** computes the signature of this watermark, $Sign_{sk_S}(W)$, using his private key sk_S.
2. Seller **S** sends both watermark *W* and its signature, $Sign_{sk_S}(W)$, to buyer **B**.
3. Buyer **B** verifies the signature of the watermark by checking whether $E_{pk_S}\left(Sign_{sk_S}(W)\right)$ is equal to *W*. If they are identical, **B** carries on with the next step. Otherwise, **B** can either request for a retransmission or cancel the transaction.
4. Buyer **B** chooses a random permutation σ_B, and uses it to perform bit permutation on each element of the watermark *W*. In other words, **B** computes

   $$W' = \left(\sigma_B(w_1), \sigma_B(w_2), \ldots, \sigma_B(w_n)\right).$$

   B also encrypts σ_B with his public key pk_B to compute $E_{pk_B}(\sigma_B)$.
5. Buyer **B** generates a public-private key pair (pk_H, sk_H) for the homomorphic cryptosystem, and then signs the public key to get $Sign_{sk_B}(pk_H)$.
6. Buyer **B** encrypts *W* with pk_H to obtain

   $$E_{pk_H}(W) = \left(E_{pk_H}(w_1), E_{pk_H}(w_2), \ldots, E_{pk_H}(w_n)\right),$$

   and then signs it using his private key sk_B to get $Sign_{sk_B}\left(E_{pk_H}(W)\right)$.
7. Buyer **B** encrypts W' with pk_H to obtain

   $$E_{pk_H}(W') = \left(E_{pk_H}\left(\sigma_B(w_1)\right), E_{pk_H}\left(\sigma_B(w_2)\right), \ldots, E_{pk_H}\left(\sigma_B(w_n)\right)\right),$$

and then signs it using his private key sk_B to get $Sign_{sk_B}\left(E_{pk_H}(W')\right)$.

8. Buyer **B** sends pk_H, $Sign_{sk_B}(pk_H)$, $E_{pk_H}(W)$, $Sign_{sk_B}\left(E_{pk_H}(W)\right)$, $E_{pk_H}(W')$, $Sign_{sk_B}\left(E_{pk_H}(W')\right)$, and $E_{pk_B}(\sigma_B)$ to seller **S**.
9. Seller **S** verifies the signature of encrypted watermark $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ by checking if $E_{pk_B}\left(Sign_{sk_B}\left(E_{pk_H}(W)\right)\right)$ is equal to $E_{pk_H}(W)$. If they are equal, **S** continues with the next step, otherwise the transaction is cancelled. In the same way, **S** also verifies the encryption key pk_H against its signature $Sign_{sk_B}(pk_H)$, and the ciphertext of permuted watermark $E_{pk_H}(W')$ against its signature $Sign_{sk_B}\left(E_{pk_H}(W')\right)$. After the encryption key pk_H is verified, **S** encrypts W with pk_H and compares the result to $E_{pk_H}(W)$ in order to ensure that **B** did not change the watermark.
10. Seller **S** finds all distinct elements of W and groups the indexes of elements that are identical into one set. **S** collects all these sets of indexes together and names it $part(W)$. For example, let $W=(a,b,c,b,a)$, then its corresponding $part(W)$ is equal to the set $\{\{1,5\},\{2,4\},\{3\}\}$. **S** then performs the same operation to $E_{pk_H}(W')$ in order to obtain the set $part\left(E_{pk_H}(W')\right)$. Observe that $part\left(E_{pk_H}(W')\right)$ is actually equal to $part(W')$ because every encryption function is injective, i.e. for all messages x and y, $x=y \Leftrightarrow E_{pk_H}(x)=E_{pk_H}(y)$.
11. Seller **S** compares the set $part(W)$ to the set $part\left(E_{pk_H}(W')\right)$. Since **B** performs the same permutation σ_B to every element of W to get W', the two

sets should be identical. Therefore, **S** only continues with the transaction if the two sets are identical. Otherwise, it is terminated as **B** has possibly changed the watermark.

12. Seller **S** generates a fingerprint V, which is unique for each customer, and then inserts it into the original copy of the digital content X to get a fingerprinted copy $X' = X \oplus V$.

13. Seller **S** chooses a random permutation σ_S, and uses it to permute the elements of the encrypted watermark $E_{pk_H}(W')$. In other words, **S** computes

$$\begin{aligned}
\sigma_S\left(E_{pk_H}(W')\right) &= \sigma_S\left(E_{pk_H}\left(\sigma_B(w_1)\right), E_{pk_H}\left(\sigma_B(w_2)\right), \ldots, E_{pk_H}\left(\sigma_B(w_n)\right)\right) \\
&= \left(E_{pk_H}\left(\sigma_B\left(w_{\sigma_S(1)}\right)\right), E_{pk_H}\left(\sigma_B\left(w_{\sigma_S(2)}\right)\right), \ldots, E_{pk_H}\left(\sigma_B\left(w_{\sigma_S(n)}\right)\right)\right) \\
&= E_{pk_H}\left(\sigma_B\left(w_{\sigma_S(1)}\right), \sigma_B\left(w_{\sigma_S(2)}\right), \ldots, \sigma_B\left(w_{\sigma_S(n)}\right)\right) \\
&= E_{pk_H}\left(\sigma_S\left(\sigma_B(w_1), \sigma_B(w_2), \ldots, \sigma_B(w_n)\right)\right) \\
&= E_{pk_H}\left(\sigma_S(W')\right).
\end{aligned}$$

The equation $\sigma_S\left(E_{pk_H}(W')\right) = E_{pk_H}\left(\sigma_S(W')\right)$ is true as $E_{pk_H}(W')$ is of the form $\left(E_{pk_H}\left(\sigma_B(w_1)\right), E_{pk_H}\left(\sigma_B(w_2)\right), \ldots, E_{pk_H}\left(\sigma_B(w_n)\right)\right)$, so that interchanging encryption and permutation operations will give us the same result.

14. Seller **S** inserts the double-permuted watermark into the fingerprinted content X' in encrypted domain. In other words, **S** first computes $E_{pk_H}(X')$, and then inserts the encrypted double-permuted watermark to it, to obtain the encrypted and watermarked content X''.

$$\begin{aligned}
E_{pk_H}(X'') &= E_{pk_H}(X') \oplus E_{pk_H}\left(\sigma_S(W')\right) \\
&= E_{pk_H}\left(X' \oplus \sigma_S(W')\right).
\end{aligned}$$

15. Seller **S** sends $E_{pk_H}\left(X''\right)$ to buyer **B**.

16. Seller **S** stores identity of buyer **B**, ID_B, pk_H, $Sign_{sk_B}\left(pk_H\right)$, $E_{pk_H}\left(W\right)$, $Sign_{sk_B}\left(E_{pk_H}\left(W\right)\right)$, $E_{pk_H}\left(W'\right)$, $Sign_{sk_B}\left(E_{pk_H}\left(W'\right)\right)$, V, $E_{pk_B}\left(\sigma_B\right)$, and σ_S as one entry in Table$_X$. Table$_X$ contains one entry for each copy of X that **S** sells.

17. Buyer **B** decrypts the encrypted content he receives from seller **S** using the corresponding decryption key sk_H to obtain the watermarked content X''. That is **B** computes

$$D_{sk_H}\left(E_{pk_H}\left(X''\right)\right) = X'' = X \oplus V \oplus \sigma_S\left(\sigma_B\left(w_1\right), \sigma_B\left(w_2\right), \ldots, \sigma_B\left(w_n\right)\right).$$

Please refer to figure 5 for the idea underlying this content-watermarking protocol.

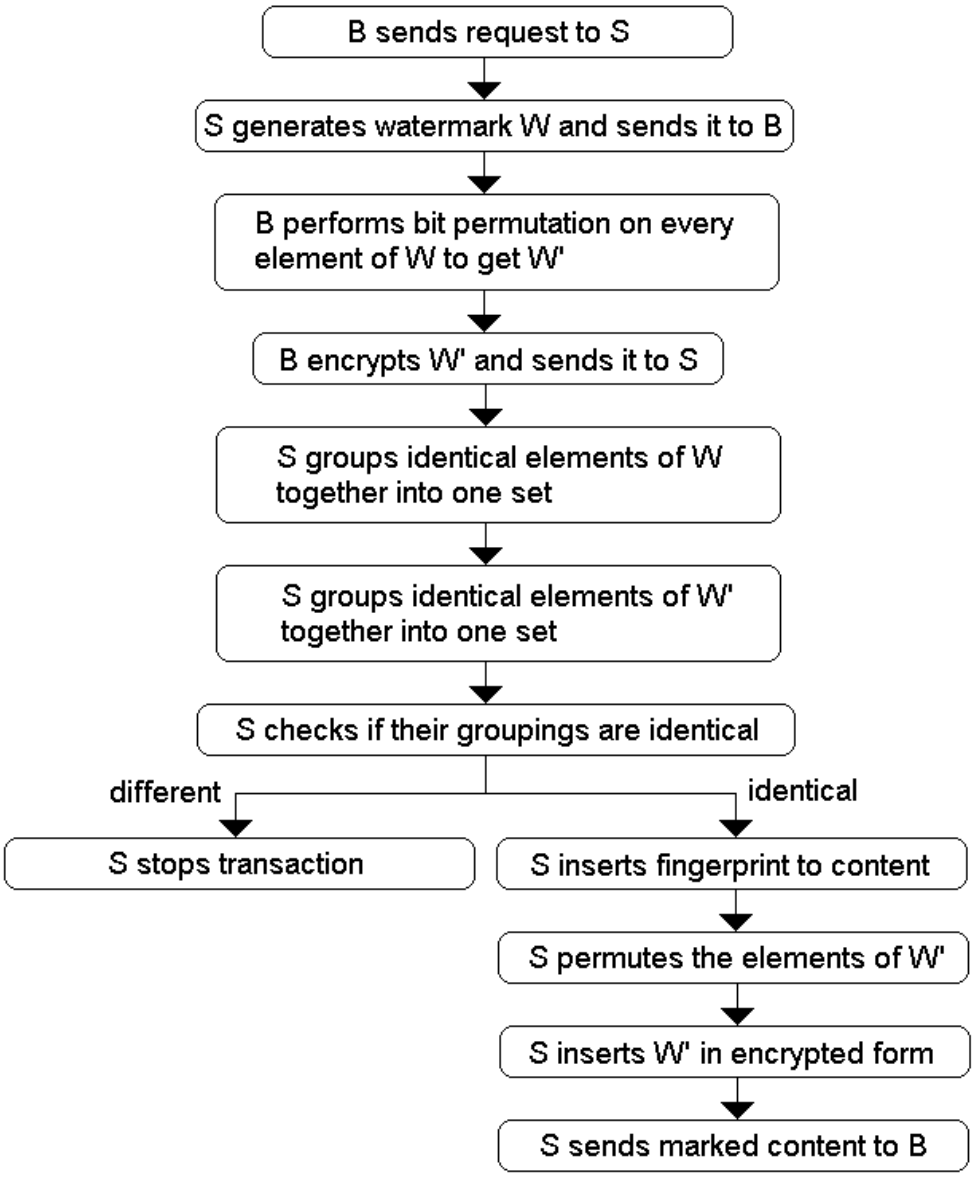

Figure 5. Content-watermarking protocol of the second protocol.

5.3.2 Copyright Violator Identification Protocol

1. When seller **S** discovers an authorized copy of content *X*, say *Y*, he extracts the unique fingerprint embedded in *Y* using the watermark extraction function *D*, which takes both *X* and *Y* as its input. Let $V_{FOUND} = D(X, Y)$ be the fingerprint detected in *Y*.

2. Seller **S** correlates V_{FOUND} with every fingerprint stored in Table$_X$ in order to find the one with the highest correlation beyond a confidence threshold. Let V_{MAX} be the fingerprint that has the highest correlation with V_{FOUND}. If fingerprint V_{FOUND} cannot be matched to any fingerprint in Table$_X$, then the protocol fails.
3. Seller **S** retrieves all the information that corresponds to fingerprint V_{MAX} from Table$_X$. The information includes the identity of buyer, say ID_B, pk_H, $Sign_{sk_B}(pk_H)$, $E_{pk_H}(W)$, $Sign_{sk_B}\left(E_{pk_H}(W)\right)$, $E_{pk_H}(W')$, $Sign_{sk_B}\left(E_{pk_H}(W')\right)$, $E_{pk_B}(\sigma_B)$, and σ_S.

Once seller **S** has the identity of buyer from whom the unauthorized copy was originated, **S** can appoint a judge **J** and proceed with dispute resolution protocol.

5.3.3 Dispute Resolution Protocol

Let **J** be the judge appointed by **S** to resolve the dispute between him and buyer **B**.

1. Seller **S** sends Y, ID_B, pk_H, $Sign_{sk_B}(pk_H)$, $E_{pk_H}(W)$, $Sign_{sk_B}\left(E_{pk_H}(W)\right)$, $E_{pk_H}(W')$, $Sign_{sk_B}\left(E_{pk_H}(W')\right)$, $E_{pk_B}(\sigma_B)$, and σ_S to judge **J**.
2. Judge **J** verifies the signature of encrypted watermark $Sign_{sk_B}\left(E_{pk_H}(W)\right)$ by checking if $E_{pk_B}\left(Sign_{sk_B}\left(E_{pk_H}(W)\right)\right)$ is equal to $E_{pk_H}(W)$. If they are equal, **J** continues with the next step, otherwise the case is dropped. In the same manner, **J** also verifies the encryption key pk_H against its signature $Sign_{sk_B}(pk_H)$, and the ciphertext of permuted watermark $E_{pk_H}(W')$ against its signature $Sign_{sk_B}\left(E_{pk_H}(W')\right)$.

3. Judge **J** sends $E_{pk_H}(W)$, $E_{pk_H}(W')$, and $E_{pk_B}(\sigma_B)$ to buyer **B**.
4. Buyer **B** decrypts $E_{pk_H}(W)$, $E_{pk_H}(W')$, and $E_{pk_B}(\sigma_B)$ using the corresponding private key sk_H to obtain $W = D_{sk_H}\left(E_{pk_H}(W)\right)$, $W' = D_{sk_H}\left(E_{pk_H}(W')\right)$, and $\sigma_B = D_{sk_B}\left(E_{pk_B}(\sigma_B)\right)$, respectively.
5. Buyer **B** sends W, W', and σ_B back to judge **J**.
6. Judge **J** verifies W, W', and σ_B by encrypting them using key pk_H, and then comparing the results to $E_{pk_H}(W)$, $E_{pk_H}(W')$, and $E_{pk_B}(\sigma_B)$ he received from **S**. If they are equal, **J** goes on with the next step. Otherwise, **B** is found guilty.
7. Judge **J** performs bit permutation σ_B on every element of watermark W and compares the resulting data to W'. **J** proceeds to the next step only if they are identical. Otherwise, **B** is deemed guilty.
8. Judge **J** computes the permuted watermark $\sigma_S(W')$ and check its existence in Y. If $\sigma_S(W')$ is detected in Y, **B** is declared guilty. Otherwise, **B** is deemed innocent.

5.4 Encryption-Based Buyer-Seller Watermarking Protocol

Although we successfully shifted certain amount of works to content provider, customer, in the second protocol, is still required to perform bit permutation on every element of the generated watermark. In the context of digital movie, due to the huge volume of the content, this operation might still be significant to some theaters with very limited resources. Moreover, allowing customers to modify the generated watermark opens an opportunity for customers to swap it with some other watermarks

which are more advantageous to them. In order to tackle this problem, we require content provider to perform a validity check after receiving the modified watermark from customers. However, since it is required that content provider does not know the exact operation done by customer, it is impossible for content provider to ensure that the watermark he receives from customer is indeed the permuted version of the one he originally generated. The customer is still able to swap the watermark with another watermark with a certain characteristic, although the swap does not make it any easier for him to break the system (please refer to Section 7.2 for details). In order to address these two problems, we propose the third protocol in which all watermarking operations are done on the seller side. It further minimizes the amount of work done by customer and at the same time eliminates the possibility of customer swapping the watermark. Nonetheless, this protocol still prevents content provider from knowing exactly the watermarked copy a customer receives.

In this protocol, upon receiving a request from a customer, content provider first generates the information sequence to be carried by the watermark. The only action that a customer has to do is to sign this sequence to prevent content provider from swapping it. In general, this sequence is much shorter than the watermark frames, causing the amount of work done by customer in this protocol to be significantly smaller than that in the previous protocol. The watermark will be then produced by content provider using this sequence of information. To conceal the watermark from customer, content provider will substitute a number of its bits. The resulting data will be then inserted to the original content, which is encrypted using customer's public key. As the result, it is the generated watermark, encrypted with

customer's private key, which will be inserted into the content, justifying the naming of our *Encryption-Based Buyer-Seller Watermarking Protocol*.

The same as the previous two protocols, our third buyer-seller watermarking protocol consists of the same three subprotocols: content-watermarking protocol, copyright violator identification protocol, and dispute resolution protocol. The detail of each subprotocol is presented below.

5.4.1 Content-Watermarking Protocol

Let **B** be the customer wanting to purchase a copy of content *X* from **S**.

1. Upon receiving a request from buyer **B**, seller **S** generates a sequence $u = (u_1, u_2, \ldots, u_p)$ containing the information to be carried by the watermark. This sequence is created specifically for this transaction only. Then, **S** computes the signature of this bit sequence, $Sign_{sk_S}(u)$, using his private key sk_S.
2. Seller **S** sends both bit sequence *u* and its signature, $Sign_{sk_S}(u)$, to buyer **B**.
3. Buyer **B** verifies the signature by checking whether $E_{pk_S}\left(Sign_{sk_S}(u)\right)$ is equal to *u*. If they are identical, **B** carries on with the next step. Otherwise, **B** can either request for a retransmission or cancel the transaction.
4. Buyer **B** signs this information sequence *u* using his private key sk_B to get the signature $Sign_{sk_B}(u)$.
5. Buyer **B** sends his signature of sequence *u*, $Sign_{sk_B}(u)$, back to seller **S**.

6. Seller **S** verifies the signature by checking whether $E_{pk_B}\left(Sign_{sk_B}\left(u\right)\right)$ is equal to u. If they are identical, **S** continues with the next step. Otherwise, **S** cancels the transaction.

7. Seller **S** selects a strictly increasing sequence of numbers $s=\left(s_1,s_2,\ldots,s_q\right)$, where $q<p$ and $1\le s_i<s_{i+1}\le p$ for all $i\in\left\{1,2,\ldots,q\right\}$. Then, **S** projects sequence u on every index contained in s, i.e. **S** extracts from u the bit sequence $u_{|s}=\left(u_{s_1},u_{s_2},\ldots,u_{s_q}\right)$.

8. Seller **S** substitutes the bit sequence $u_{|s}$ with another q-bit sequence $\hat{u}_{|s}=\left(\hat{u}_{s_1},\hat{u}_{s_2},\ldots,\hat{u}_{s_q}\right)$. This can be done using the same concept as that of *S-box* used in Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The idea is to split the sequence $u_{|s}$ into two parts, then take the decimal interpretation of these two binary sequences. Let the two numbers be r_u and c_u. After that, retrieve the q-bit binary sequence stored in row r_u and column c_u of a pre-generated table. The dimension of the *S-box* table depends on the value of q and how we split the sequence $u_{|s}$. The same *S-box* table can be used in every iteration of the protocol, i.e. the *S-box* table is fixed.

   For example, assume $u_{|s}=\left(10011110\right)$ and we split it right in the middle, i.e. the two parts are $\left(1001\right)$ and $\left(1110\right)$, then $r_u=\left(1001\right)_{10}=9$ and $c_u=\left(1110\right)_{10}=14$. After that, do a table look-up to retrieve the binary string stored in row 9 and column 14 of the *S-box* table, and then use it as $\hat{u}_{|s}$.

9. For all $i \in \{1,2,\ldots,q\}$, seller **S** puts back every $\hat{u}_{s_i}$ to position s_i of sequence u, i.e. **S** puts every $\hat{u}_{s_i}$ back to the position where u_{s_i} is taken, to get a new information sequence $\hat{u} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_p)$, where for all $i \in \{1,2,\ldots,p\}$,

$$\hat{u}_i = \begin{cases} \hat{u}_{s_j} \text{ if } i = s_j \text{ for some } j \in \{1,2,\ldots,q\} \\ u_i \text{ , otherwise} \end{cases}.$$

10. Seller **S** generates watermark $W = (w_1, w_2, \ldots, w_n)$ from the information sequence $\hat{u} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_p)$. This generation step is elaborated in Section 6.2.1.
11. Seller **S** generates a fingerprint V, which is unique for each customer, and then inserts it into the original copy of the digital content X to get a fingerprinted copy $X' = X \oplus V$.
12. Seller **S** sends a request for a pair of public-private key to certification authority **CA**. This key pair will be used in the homomorphic cryptosystem.
13. Upon receiving a request from **S**, **CA** generates a public-private key pair (pk_H, sk_H) for the specified homomorphic cryptosystem. **CA** encrypts the public key pk_H using seller's public key pk_S to get $E_{pk_S}(pk_H)$, and then signs the ciphertext using his private key sk_{CA} to obtain $Sign_{sk_{CA}}(E_{pk_S}(pk_H))$. Different from the public key, the private key sk_H is encrypted using buyer's public key pk_B to get $E_{pk_B}(sk_H)$, and then the ciphertext is signed by **CA** using his private key sk_{CA} to get $Sign_{sk_{CA}}(E_{pk_B}(sk_H))$.
14. Certification Authority CA sends $E_{pk_S}(pk_H)$, $Sign_{sk_{CA}}(E_{pk_S}(pk_H))$, $E_{pk_B}(sk_H)$, and $Sign_{sk_{CA}}(E_{pk_B}(sk_H))$ to seller **S**.

15. Seller **S** verifies the signature of the encrypted public key $Sign_{sk_{CA}}\left(E_{pk_S}\left(pk_H\right)\right)$ by checking whether $E_{pk_{CA}}\left(Sign_{sk_{CA}}\left(E_{pk_S}\left(pk_H\right)\right)\right)$ is equal to $E_{pk_S}\left(pk_H\right)$. In the same manner, **S** verifies $E_{pk_B}\left(sk_H\right)$ against it signature $Sign_{sk_{CA}}\left(E_{pk_B}\left(sk_H\right)\right)$.

16. Seller **S** decrypts $E_{pk_S}\left(pk_H\right)$ using his private key sk_S to retrieve the public key $pk_H = D_{sk_S}\left(E_{pk_S}\left(pk_H\right)\right)$. **S** then uses pk_H to encrypt the fingerprinted content X' and get $E_{pk_H}\left(X'\right)$.

17. Seller **S** inserts the watermark W generated earlier to the ciphertext of fingerprinted content $E_{pk_H}\left(X'\right)$ to get the ciphertext of watermarked content $E_{pk_H}\left(X''\right)$. It is assumed that the homomorphic cryptosystem is length-preserving, i.e. plaintext has the same length as its corresponding ciphertext. In other word, the domain of its encryption function is the same as that of its decryption function.

$$\begin{aligned} E_{pk_H}\left(X''\right) &= E_{pk_H}\left(X'\right) \oplus W \\ &= E_{pk_H}\left(X'\right) \oplus D_{sk_H}\left(E_{pk_H}\left(W\right)\right) \\ &= E_{pk_H}\left(X'\right) \oplus E_{pk_H}\left(D_{sk_H}\left(W\right)\right) \\ &= E_{pk_H}\left(X' \oplus D_{sk_H}\left(W\right)\right) \end{aligned}$$

18. Seller **S** sends $E_{pk_H}\left(X''\right)$, $E_{pk_B}\left(sk_H\right)$, and $Sign_{sk_{CA}}\left(E_{pk_B}\left(sk_H\right)\right)$ to buyer **B**.

19. Seller **S** stores identity of buyer **B**, ID_B, u, $Sign_{sk_B}\left(u\right)$, s, *S-box*, $E_{pk_S}\left(pk_H\right)$, $Sign_{sk_{CA}}\left(E_{pk_S}\left(pk_H\right)\right)$, $E_{pk_B}\left(sk_H\right)$, $Sign_{sk_{CA}}\left(E_{pk_B}\left(sk_H\right)\right)$, and V as one entry in Table_X. Table_X contains one entry for each copy of X that **S** sells.

20. Buyer **B** verifies the encrypted private key $E_{pk_B}(sk_H)$ against its signature $Sign_{sk_{CA}}\left(E_{pk_B}(sk_H)\right)$ by comparing $E_{pk_{CA}}\left(Sign_{sk_{CA}}\left(E_{pk_B}(sk_H)\right)\right)$ to $E_{pk_B}(sk_H)$. If they are identical, **B** continues with the next step. Otherwise, **B** may return the content to **S** and ask for a refund.

21. Buyer **B** decrypts $E_{pk_B}(sk_H)$ using his private key sk_B to recover the private key $sk_H = D_{sk_B}\left(E_{pk_B}(sk_H)\right)$. **B** then uses this private key to decrypt the encrypted content $E_{pk_H}(X'')$ he received from seller **S** and obtain the watermarked content X''. That is **B** computes

$$D_{sk_H}\left(E_{pk_H}(X'')\right) = X'' = X \oplus V \oplus D_{sk_H}(W).$$

Please refer to figure 6 for the idea underlying this content-watermarking protocol.

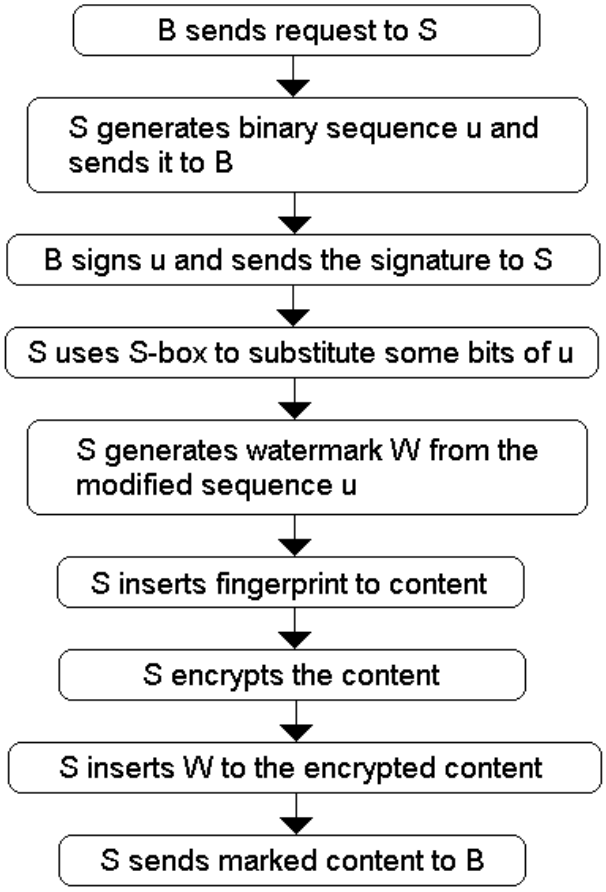

Figure 6. Content-watermarking protocol of the third protocol.

5.4.2 Copyright Violator Identification Protocol

1. When seller **S** discovers an authorized copy of content *X*, say *Y*, he extracts the unique fingerprint embedded in *Y* using the watermark extraction function *D*, which takes both *X* and *Y* as its input. Let $V_{FOUND} = D(X, Y)$ be the fingerprint detected in *Y*.
2. Seller **S** correlates V_{FOUND} with every fingerprint stored in Table_X in order to find the one with the highest correlation beyond a confidence threshold. Let V_{MAX} be the fingerprint that has the highest correlation with V_{FOUND}. If fingerprint V_{FOUND} cannot be matched to any fingerprint in Table_X, then the protocol fails.
3. Seller **S** retrieves all the information that corresponds to fingerprint V_{MAX} from Table_X. The information includes the identity of buyer, say ID_B, u, $Sign_{sk_B}(u)$, $E_{pk_S}(pk_H)$, $Sign_{sk_{CA}}\left(E_{pk_S}(pk_H)\right)$, $E_{pk_B}(sk_H)$, $Sign_{sk_{CA}}\left(E_{pk_B}(sk_H)\right)$, *S-box*, and *s*.

Once seller **S** has the identity of buyer from whom the unauthorized copy was originated, **S** can appoint a judge **J** and proceed with dispute resolution protocol.

5.4.3 Dispute Resolution Protocol

Let **J** be the judge appointed by **S** to resolve the dispute between him and buyer **B**.

1. Seller **S** sends *Y*, ID_B, *u*, $Sign_{sk_B}(u)$, *s*, *S-box*, $E_{pk_B}(sk_H)$, and $Sign_{sk_{CA}}\left(E_{pk_B}(sk_H)\right)$ to judge **J**.
2. Judge **J** verifies the signature of sequence *u*, $Sign_{sk_B}(u)$ by checking if $E_{pk_B}\left(Sign_{sk_B}(u)\right)$ is equal to *u*. If they are equal, **J** continues with the next step,

otherwise the case is dropped. **J** also verifies the signature $Sign_{sk_{CA}}\left(E_{pk_B}\left(sk_H\right)\right)$ by encrypting it using **CA**'s public key pk_{CA}, followed by comparing the result to $E_{pk_B}\left(sk_H\right)$. Similarly, **J** only continues if they are the same.

3. Judge **J** derives the sequence $\hat{u}$ from the sequence *u* using set of indexes *s* and the substitution table *S-box* in the same way as seller **S** did. Please refer to Section 5.4.1 step 7-9 for details.
4. Judge **J** generates the watermark *W* from the sequence $\hat{u}$ by following the same procedure as seller **S** did. The watermark construction process is explained in Section 6.2.1.
5. Judge **J** sends $E_{pk_B}\left(sk_H\right)$ to buyer **B**.
6. Buyer **B** decrypts $E_{pk_B}\left(sk_H\right)$ using his private key sk_B to recover the secret key $sk_H = D_{sk_B}\left(E_{pk_B}\left(sk_H\right)\right)$.
7. Buyer **B** sends sk_H back to judge **J**.
8. Judge **J** verifies the key sk_H he received from **B** by encrypting it using **B**'s public key pk_B, and then comparing the result to $E_{pk_B}\left(sk_H\right)$ he received from **S**. If they are equal, **J** goes on with the next step. Otherwise, **B** is found guilty.
9. Judge **J** decrypts the watermark *W* using the key sk_H to compute $D_{sk_H}\left(W\right)$.
10. Judge **J** checks the existence of $D_{sk_H}\left(W\right)$ in the unauthorized copy *Y*. If it is detected in *Y*, **B** is declared guilty. Otherwise, **B** is deemed innocent.

6. CONSTRUCTION DETAILS

For clarity and simplicity reasons, the details of cryptosystems and watermarking techniques were not included in the previous section. We assumed the existence of a cryptosystem that is privacy homomorphic with respect to the watermark insertion operation without mentioning any specific cryptosystems satisfying the desired property and explaining how the encryption and decryption are done. Neither did the explanation of each protocol contain any information about how a watermark is generated, embedded, and detected.

In this section, all this information will be provided in order to complete the explanation of our protocols. We will first introduce four cryptosystems that are privacy homomorphic with respect to either addition or multiplication, and then we explain briefly how encryption and decryption are done in each of the cryptosystems. In the second part of this section, we will present a spread-spectrum watermarking technique that can possibly be used in our protocols. The explanation will include watermark construction, insertion, and detection methods.

6.1 Privacy Homomorphic Cryptosystem

A cryptosystem is a *privacy homomorphism* with respect to operation *op* if and only if it has the property that

$$E_k\left(m_1 \; op \; m_2\right) = E_k\left(m_1\right) \; op \; E_k\left(m_2\right)$$

for any m_1 and m_2 in the message space and for any k in the key space [40]. So, encrypting two messages first, followed by applying operation *op* on the ciphertexts will result in the same value as applying the operation *op* first, followed by encrypting the output. This property enables us to insert a watermark in the encrypted domain, so

content provider is able to insert the watermark into the content without knowing what is exactly being inserted.

RSA [45] and El Gamal [22] cryptosystems are two examples of cryptosystems that are homomorphic with respect to multiplication, whereas Niederreiter cryptosystem [17][42] is an example of a homomorphism with respect to addition. Combining the two operations, multiplication and addition, Paillier cryptosystem [43] is homomorphic from multiplication to addition. We explain briefly the encryption and decryption functions of each of these four cryptosystems below.

6.1.1 RSA Cryptosystem

RSA cryptosystem [45] is designed based on the factoring problem. As opposed to multiplication, which is easy, finding the factors of a given number is difficult, particularly when the number is a multiplication of two large prime numbers. The security of RSA cryptosystem relies on the difficulty of factoring such large integers.

- Public key: a large integer $n = pq$, where p and q are two large prime numbers, and an integer b, where $2 \le b \le \phi(n) = (p-1)(q-1)$ and $\gcd(b, \phi(n)) = 1$.
- Private key: two prime factors of n, p and q, the Euler function of n, $\phi(n) = (p-1)(q-1)$, and the multiplicative inverse of b, $a \equiv b^{-1} \pmod{\phi(n)}$.
- Encryption: for any plaintext $x \in \mathbb{Z}_n$, the corresponding ciphertext is

$$E(x) = x^b \pmod{n}.$$

- Decryption: for any ciphertext $y \in \mathbb{Z}_n$, the corresponding plaintext is

$$D(y) = y^a \pmod{n}.$$

- RSA is a privacy homomorphism with respect to multiplication.

  For any two plaintexts x_1 and x_2,

$$\begin{aligned} E(x_1 \cdot x_2) &= (x_1 \cdot x_2)^b \pmod{n} \\ &= x_1^b \cdot x_2^b \pmod{n} \\ &= \left(x_1^b \pmod{n}\right) \cdot \left(x_2^b \pmod{n}\right) \\ &= E(x_1) \cdot E(x_2). \end{aligned}$$

6.1.2 El Gamal Cryptosystem

El Gamal cryptosystem [22] is constructed with discrete logarithm problem as the underlying idea. It is easy to raise a number to certain power, but finding the logarithm of a number is much more difficult. The security of the El Gamal cryptosystem is provided by the difficulty of finding the unique discrete logarithm of a number modulo a prime number.

- Public key: a prime number *p*, a primitive element modulo *p*, *g*, and a number

$$\alpha = g^a \pmod{p}.$$

- Private key: the discrete logarithm of α modulo p, $a = \log_g \alpha \pmod{p}$,

  where $2 \le a \le p-2$.

- Encryption: for any plaintext $x \in \mathbb{Z}_p$ and a random *k*, the corresponding

  ciphertext is $E(x) = (y_1, y_2)$, where

$$y_1 = g^k \pmod{p}$$

$$y_2 = x \cdot \alpha^k \pmod{p}.$$

- Decryption: for any ciphertext (y_1, y_2), the corresponding plaintext is

$$D(y_1, y_2) = y_2 \cdot (y_1^a)^{-1} \pmod{p}.$$

- El Gamal cryptosystem is a privacy homomorphism with respect to multiplication.

  For any two ciphertexts (y_1, y_2) and (z_1, z_2), where

$$y_1 = g^k \pmod{p} \qquad z_1 = g^m \pmod{p}$$

$$y_2 = x_1 \cdot \alpha^k \pmod{p} \qquad z_2 = x_2 \cdot \alpha^m \pmod{p}$$

$$\begin{aligned} y_1 \cdot z_1 &= \left(g^k \pmod{p}\right) \cdot \left(g^m \pmod{p}\right) \\ &= g^k \cdot g^m \pmod{p} \\ &= g^{k+m} \pmod{p} \end{aligned}$$

$$\begin{aligned} y_2 \cdot z_2 &= \left(x_1 \cdot \alpha^k \pmod{p}\right) \cdot \left(x_2 \cdot \alpha^m \pmod{p}\right) \\ &= x_1 \cdot \alpha^k \cdot x_2 \cdot \alpha^m \pmod{p} \\ &= (x_1 \cdot x_2) \cdot \alpha^{k+m} \pmod{p} \end{aligned}$$

$$\Rightarrow E(x_1) \cdot E(x_2) = E(x_1 \cdot x_2)$$

6.1.3 Niederreiter Cryptosystem

Niederreiter cryptosystem [17][42] is designed based on the concept of coding theory. The security of this cryptosystem lies on the difficulty of decoding process of a linear code. Niederreiter's system uses a linear $[n, k, d]$ code *C* over finite field F_q, where *n* is the length of each codeword in *C*, *k* is the dimension of *C*, and *d* is the minimum Hamming distance of *C* [17]. The information in this section is compiled from [17] and [42].

- Private key: three matrices H, M, and P, where H is an $(n-k)\times n$ parity-check matrix of C, M is an arbitrary $(n-k)\times(n-k)$ invertible matrix, and P is an arbitrary $n\times n$ permutation matrix.
- Public key: an $(n-k)\times n$ matrix $H' = MHP$.
- Encryption: the admissible plaintexts are column vectors with hamming weight of at most $t = \lfloor (d-1)/2 \rfloor$. The hamming weight of a vector x, $w(x)$, is defined as the number of non-zero entries in x. For any plaintext x, the corresponding ciphertext is $E(x) = H' \cdot x$.
- Decryption: given any ciphertext y, a column vector, first compute $y' = M^{-1} \cdot y = H \cdot P \cdot x$. Let $x' = P \cdot x$, then x' can be viewed as an error vector. The decoding algorithm of C is applied to the syndrome $y' = H \cdot x'$ to yield the error vector x'. The plaintext x is recovered by multiplying x' to P^{-1}, i.e. $x = P^{-1} \cdot x'$.
- Niederreiter cryptosystem is a privacy homomorphism with respect to addition.

  For any two plaintexts x_1 and x_2,

$$\begin{aligned} E(x_1 + x_2) &= H' \cdot (x_1 + x_2) \\ &= (H' \cdot x_1) + (H' \cdot x_2) \\ &= E(x_1) + E(x_2). \end{aligned}$$

6.1.4 Paillier Cryptosystem

Paillier cryptosystems [43] are constructed based on the *Composite Residuosity Class Problem*. Due to the complex nature of the problem, we are not going to discuss it any further. Interested readers may refer to [43] for further details about Composite Residuosity Class Problem. The encryption process of Paillier systems is very similar to the vote encryption process of Cohen and Fischer's Cryptographically Secure Election Scheme [12]. However, Cohen and Fischer did not explain the corresponding decryption process, making Paillier's systems a better choice for us to present in this report. We present an overview to each of the two cryptosystems proposed by Paillier below.

6.1.4.1 First Cryptosystem

- Private key: two large prime numbers p and q, *Carmichael*'s function of $n = pq$, $\lambda = \text{lcm}(p-1, q-1)$.
- Public key: a number $n = pq$, a base $g \in B \subseteq \mathbb{Z}^*_{n^2}$, where $\gcd\left(L\left(g^\lambda \bmod n^2\right), n\right) = 1$ and B is the set of elements of order $n\alpha$ for $\alpha = 1, 2, \ldots, \lambda$. For each $u \in \left\{v < n^2 \mid v \equiv 1 \bmod n\right\}$, the function L is defined as $L(u) = (u-1)/n$.
- Encryption: for any plaintext $x < n$ and a random $r < n$, the corresponding ciphertext is

$$E(x) = g^x \cdot r^n \left(\bmod n^2\right)$$

- Decryption: for any ciphertext $y < n^2$, the corresponding plaintext is

$$D(y) = \frac{L\left(y^{\lambda} \left(\text{mod } n^2\right)\right)}{L\left(g^{\lambda} \left(\text{mod } n^2\right)\right)} \text{ mod } n.$$

- The first Paillier cryptosystem is privacy homomorphic from multiplication to addition.

  For any two ciphertexts $E(x_1) = g^{x_1} \cdot r_1^n \left(\text{mod } n^2\right)$ and $E(x_2) = g^{x_2} \cdot r_2^n \left(\text{mod } n^2\right)$,

$$\begin{aligned} E(x_1) \cdot E(x_2) &= \left(g^{x_1} \cdot r_1^n \left(\text{mod } n^2\right)\right) \cdot \left(g^{x_2} \cdot r_2^n \left(\text{mod } n^2\right)\right) \\ &= g^{x_1} \cdot r_1^n \cdot g^{x_2} \cdot r_2^n \left(\text{mod } n^2\right) \\ &= g^{x_1 + x_2} \cdot \left(r_1 \cdot r_2\right)^n \left(\text{mod } n^2\right) \\ &= E(x_1 + x_2). \end{aligned}$$

6.1.4.2 Second Cryptosystem

- Private key: two large prime numbers p and q, *Carmichael*'s function of $n = pq$, $\lambda = \text{lcm}(p-1, q-1)$, and a number α, where $1 \le \alpha \le \lambda$.
- Public key: a number $n = pq$, a base $g \in B_\alpha \subseteq \mathbb{Z}^*_{n^2}$, where B_α is the set of elements of order $n\alpha$ for some $1 \le \alpha \le \lambda$, and a function L defined on every $u \in \left\{v < n^2 \mid v \equiv 1 \text{ mod } n\right\}$ as $L(u) = (u-1)/n$.
- Encryption: for any plaintext $x < n$ and a random $r < n$, the corresponding ciphertext is

$$E(x) = g^{x+nr} \text{ mod } n^2$$

- Decryption: for any ciphertext $y < n^2$, the corresponding plaintext is

$$D(y) = \frac{L\left(y^{\alpha} \left(\text{mod } n^2\right)\right)}{L\left(g^{\alpha} \left(\text{mod } n^2\right)\right)} \text{ mod } n .$$

- The second Paillier cryptosystem is also privacy homomorphic from multiplication to addition.

  For any two ciphertexts $E(x_1) = g^{x_1+nr_1} \text{ mod } n^2$ and $E(x_2) = g^{x_2+nr_2} \text{ mod } n^2$,

$$\begin{aligned} E(x_1) \cdot E(x_2) &= \left(g^{x_1+nr_1} \text{ mod } n^2\right) \cdot \left(g^{x_2+nr_2} \text{ mod } n^2\right) \\ &= g^{x_1+nr_1} \cdot g^{x_2+nr_2} \left(\text{mod } n^2\right) \\ &= g^{x_1+nr_1+x_2+nr_2} \text{ mod } n^2 \\ &= g^{(x_1+x_2)+n(r_1+r_2)} \text{ mod } n^2 \\ &= E(x_1 + x_2) . \end{aligned}$$

6.1.5 Discussion

The four cryptosystems mentioned above can be split into two groups according to the operations with respect to which they are homomorphic, addition and multiplication. Thus, the choice of cryptosystem to use determines the operation to perform in the watermark insertion process. If the cryptosystem is homomorphic to addition, then the watermark is inserted using addition operation. Similarly, multiplication operation is performed to embed the watermark if the cryptosystem is homomorphic to multiplication.

In each of the two groups, we have two cryptosystems to choose. When addition is preferred, we can use either Niederreiter's system or Paillier's system, whereas RSA and El Gamal are applicable when multiplication operation is more desirable.

Niederreiter cryptosystem, which is based on the concept of coding theory, is faster than Paillier's system with comparable security levels. Niederreiter's system is reported to be 48 times faster than RSA cryptosystem, which simpler than Paillier's system. However, Niederreiter's system adds too much redundancy to the ciphertext and causes a severe expansion in the size of the ciphertext. Emmanuel and Kankanhalli [17] mentioned that expansion factor of Niederreiter's system is at least ten. In terms of length expansion, Paillier's system is much better as it only expands the length of ciphertext to at most twice the length of the plaintext. Nonetheless, it has higher time complexity compared to Niederreiter's system. Either cryptosystems can be used according to needs and the availability of resources. When time is an important constraint, Neiderreiter's system makes a better choice. Similarly, when space efficiency is more prioritized, Paillier's system is definitely a wiser choice.

RSA and El Gamal cryptosystems perform similar set of operations in their encryption and decryption process. Both cryptosystems requires exponentiation and modulo operations. Nevertheless, for a comparable security measure, El Gamal requires larger number of operations than RSA, and therefore requires more intensive computation than RSA [41][49]. As the consequence, El Gamal is slower and less efficient than RSA, although the difference is not significant on modern processors. In terms of length expansion, RSA is also superior to El Gamal cryptosystem. RSA does not cause any expansion as both plaintext and ciphertext are of the same size, whereas El Gamal produces ciphertext that is twice longer than its corresponding plaintext. Moreover, El Gamal requires the use of a random number in its encryption process. Therefore, it has a need for "good" randomness to generate a unique and unpredictable value for this parameter. Otherwise, it may open a chance for adversary

to obtain the private key [49]. Therefore, RSA is a better choice than El Gamal when multiplication operation is preferred in the watermark embedding process.

When it does not really matter whether addition or multiplication is used in the watermark embedding process, RSA cryptosystem is the system we suggest. It is better established and more maturely studied than both Niederreiter's and Paillier's systems. Thus, its security is more guaranteed compared to that of the other two systems. RSA also eliminates the message expansion problem, which both Niederreiter's and Paillier's systems have. Unfortunately, RSA is much slower than Niederreiter's system.

6.2 Watermarking Scheme

In our first two protocols, content provider performs permutation on the generated watermark in order to prevent customer from knowing the exact watermark being inserted into the content. It implies that we need a watermarking scheme that is *linear*. A watermarking scheme is linear if the watermark can be inserted element-wise, that is the insertion of a watermark element is independent of the insertion of other watermark elements. Let $X = (x_1, x_2, \ldots, x_m)$ denote the content to be watermarked, $W = (w_1, w_2, \ldots, w_n)$ be the watermark to insert with $m \geq n$, and $\oplus$ be the watermark insertion operation. A watermark scheme is linear if the watermark insertion step can be represented as

$$X' = X \oplus W = (x_1 \oplus w_1, x_2 \oplus w_2, \ldots, x_n \oplus w_n, x_{n+1}, \ldots, x_m).$$

Although the watermarking scheme used in the third protocol need not be linear, the watermarking scheme presented in this section is linear to accommodate the other two protocols.

As we can consider a video as a sequence of images, each called a frame, video watermarking process can be viewed as watermarking a large number of images. Therefore, in this section, we shall only explain how the watermarking scheme is applied to a single frame. The whole process can be repeated to many other frames according to content provider's need. Content provider can choose to watermark either all frames or only a certain subset of those frames.

6.2.1 Watermark Construction

The watermarking construction technique presented in this section is taken from Emmanuel and Kankanhalli's work [17].

The watermark construction process starts with a process that maps the information sequence $u = (u_1, u_2, \ldots, u_p)$, $u_i \in \{0,1\}$ to a sequence $a = (a_1, a_2, \ldots, a_p)$, where for all $i \in \{1, 2, \ldots, p\}$

$$a_i = \begin{cases} -1 & \text{if } u_i = 0 \\ 1 & \text{, otherwise} \end{cases}.$$

The resulting sequence a is then spread using the chip rate C_r to obtain the spread sequence b of length $C_r \times p$. The chip rate C_r and the length of information sequence p are selected in such a way that $C_r \times p = n$. The spread sequence b is constructed as follows:

$$\forall j: \; b_i = a_j, \; jC_r \le i < (j+1)C_r$$

The spreading provides redundancy and improves the robustness to geometrical attacks such as cropping. After spreading the information sequence, we multiply the spread sequence with a pseudorandom noise sequence z, where

$z_i \in \{-1,1\}$. The multiplication will be followed by amplification of the result by a scaling factor $\gamma > 0$ to obtain the watermark $W = (w_1, w_2, \ldots, w_n)$, where

$$\forall i \in \{1,2,\ldots,n\}: \quad w_i = \gamma b_i z_i$$

The scaling factor γ is chosen in such a way that the watermark still remains detectable and, at the same time, invisible in the watermarked frames.

6.2.2 Watermark Embedding

We use the same watermarking technique as the one used by Memon and Wong [40], which is the spread-spectrum watermarking technique proposed by Cox et al. [14].

Let *X* be the video to watermark, *I* be the set of indexes indicating the subset of the video frames to watermark, and X_i be the *i*-th frame of the content *X*. We apply the watermarking scheme proposed by Cox et al. [14] to insert the watermark generated into each frame X_i, $i \in I$.

In Cox.et al.'s scheme, the content frame X_i is first compressed by performing two-dimensional *Discrete Cosine Transform (DCT)*. The *n* largest DCT AC coefficients are then extracted for watermarking. Results reported using 1000 DCT AC coefficients show the technique to be remarkably robust against various image processing operations, and also after printing and rescanning [40]. Let $\{x_1, x_2, \ldots, x_n\}$ denote the *n* largest DCT AC coefficients. Each watermark element w_i is embedded to coefficient x_i using the suitable insertion formula to yield the modified coefficients x_i'. The choice of insertion formula depends on the type of cryptosystem used. If the cryptosystem is a homomorphism with respect to addition, we can simply add the watermark to the coefficients, that is to compute

$$x_i' = x_i + w_i .$$

However, if the cryptosystem used is homomorphic with respect to multiplication, we need to first add 1 to the watermark elements before multiplying it to the coefficients, that is to use the following formula:

$$x_i' = x_i \times (1 + w_i) .$$

Observe that we do not multiply the watermark element by a scaling factor in both formulas. It is because the scaling of watermark element is carried out during the watermark construction process. Please refer to the previous subsection for details of this process.

After the modified coefficients $\{x_1', x_2', \ldots, x_n'\}$ are computed, the inverse of two-dimensional DCT is performed on these coefficients in order to obtain the watermarked frame X_i'. The whole embedding process is repeated to insert the watermark to other video frames.

6.2.3 Watermark Detection

In this section, we shall see how we can determine whether a video frame contains a watermark *W*. In other words, we shall discuss about the inverse of watermark embedding operation explained in Section 6.2.2. The watermark detection is done in a non-blind manner, i.e. it is performed with the existence of the original copy of the content. The information presented below is taken from [40].

Suppose we want to check the existence of watermark W in a video frame Y_i. First, the same two-dimensional DCT as explained in the previous subsection is applied to the frame Y_i. Then, we need to extract the n largest DCT AC coefficients,

let's denote it by $\{y_1, y_2, \ldots, y_n\}$. We then subtract these values from the n largest DCT AC coefficients of the corresponding frame X_i of the original content, $\{x_1, x_2, \ldots, x_n\}$, i.e. to compute $T = (t_1, t_2, \ldots, t_n)$ where

$$\forall i \in \{1, 2, \ldots, n\}: \quad t_i = x_i - y_i .$$

After T is computed, we compute the correlation between W and T. This correlation value indicates the confidence measure on the existence of watermark W in Y_i.

7. ANALYSIS

In the proposed protocols, we combine several different concepts together in order to achieve our objectives. Therefore, the properties of the protocols highly depend on those of the building blocks used to construct them. In this section, we shall discuss how the properties of the underlying concepts are utilized in order to fulfill the requirements mentioned in the earlier part of this report. We shall first see some characteristics which are common to those three protocols, and then we shall examine how each of these three protocols solves the customer's right problem in its own way.

Security

The security of the three proposed protocols relies on the security of the underlying cryptosystem, watermarking scheme, and the permutation.

The cryptosystem that we recommend, RSA cryptosystem, is very well-established and maturely studied, causing its security to be more reliable compared to the other homomorphic cryptosystems. RSA is believed to be secure if the proper parameters are used and it is employed properly. The choice of the two prime

numbers is highly important in RSA. It is reported that the length of each prime should be at least 1024 bits in order to achieve a guaranteed level of security [49]. RSA also eliminates the message expansion problem, which the other alternatives have.

Although people are still questioning the ability of many watermarking schemes to withstand many different known attacks due to the inexistence of standard performance measure, Cox et al.'s watermarking technique used in our three protocols is one of the best known and has been shown to be remarkably robust against common image processing attacks and even several cycles of analog to digital conversions. The robustness of the scheme critically depends on the availability of the original content which can be used to undo operations like scaling, cropping, rotations, and some other operations prior to watermark detection step [40].

The choice of permutations used in the first two protocols also plays an important role in ensuring the security of the protocols. The permutations must be chosen in such a way that the permuted watermark appears random and it does not expose any information about the original watermark. The number of watermark elements and the size of each element should be designed to be large enough in order to prevent attackers from performing brute-force attack and guessing the permutation used.

Traceability, Collusion Resistance, and No Framing by Malicious Users

Traceability is achieved in the three proposed protocols by inserting a unique fingerprint, denoted by V, to each copy of the content. It is the responsibility of content provider to ensure that each fingerprint inserted is unique for each customer

and to maintain a list of fingerprints used and their respective owners, so that it enables him to trace the source of an unauthorized distribution act from the fingerprint detected in an illegal copy of the content. It does not do any good for content provider not to perform the fingerprinting properly. Thus, it can be assumed that content provider inserts the proper fingerprint in a proper manner in order to guarantee the traceability.

In order to prevent a coalition of users from colluding their copies to remove the fingerprint or to frame another user, we can encode the fingerprints using collusion-resistant codes. Boneh and Shaw [2] have shown a way to construct a code that can satisfy these requirements. Their *c-secure* and *c-frameproof* code can be employed in order to ensure that content provider is able to identify at least one of the *c* colluders without falsely accusing an innocent user. The large size of those codes is not a problem in the context of video fingerprinting. The huge volume of the content provides a space for embedding a lengthy fingerprint.

Anonymity and Unlinkability

In order to provide anonymity, we can require each customer to use an anonymous certificate instead of the standard public-key certificate in every transaction he makes. Anonymous certificate is basically a public-key certificate which does not reveal the identity of the owner. Instead, a pseudonym is used to identify the owner. Each customer who does not wish their identity to be disclosed is able to request for an anonymous certificate to certification authority (CA), and then use it during authentication process preceding a transaction. In this case, content provider will not know the true identity of the customer. The true identity of customer is only known by

CA. The true identity of a customer is only exposed when he is suspected of an illegal copying and distribution in order to facilitate the dispute resolution protocol. The possibility of coalition between CA and content provider can be ruled out as CA is assumed honest and trustworthy. Otherwise, there is even no public-key infrastructure that is secure to be used in the protocols.

Nonetheless, anonymous certificate and pseudonym do not prevent people from relating two different copies of digital content purchased under the same pseudonym. To solve this problem, we need to require the anonymous certificate and pseudonym to be used for a limited number of transactions only. Customers need to request for a new anonymous certificate and a new pseudonym on a regular basis in order to securely hide their identity.

Binding mechanism

Unbinding problem, caused by failure to provide proper mechanism to bind a generated watermark to a specific digital content it is inserted, can be avoided by inserting to each copy of the content a watermark that contains the identification of each content copy. It can be done by including a time stamp indicating the time of transaction, a nonce, the title of the content, and the identity of parties involved in the transaction into the watermark to be inserted. This information is used to differentiate each pair of copies purchased by the same customer. This way, content provider will not be able to transplant a watermark detected in a pirated copy into other copies of (possibly higher-priced) digital contents in order to get more compensation from a guilty customer.

No Additional Trusted Third Party

The most distinctive feature of our protocols that differentiates our protocols from other existing solutions is the absence of watermark certification authority (WCA). None of our protocols requires the involvement of an additional trusted third party, other than CA and the arbiter, in any stage of a transaction. As mentioned earlier, the assumption on arbiter's and CA's honesty is acceptable since it also exists in the traditional fingerprinting and watermarking schemes. Moreover, CA is a party guaranteeing the secrecy of private keys in any public-key infrastructure, thus it is definitely trustworthy and reliable. In our protocols, the watermark is generated by either customer or content provider. Therefore, we can now rule out the possibility of coalition between seller and WCA existing in other protocols.

Despite the removal of WCA role in our protocols, we take into consideration the underlying reason why WCA was, in the first place, introduced. In the first protocol, we solve the problem of watermarks that are approximately invariant to permutation by requiring content provider to check the validity of watermark generated by customer. In the second protocol, watermark generation is performed by content provider. So, it is clear that he will not produce a watermark which is approximately invariant to permutation as it means helping customer to remove the watermark. In the third protocol, this problem does not even exist as no permutation is used.

7.1 Memon and Wong's Buyer-Seller Watermarking Protocol without Watermark Certification Authority

Being a variant of Memon and Wong's protocol [39][40], our first protocol solves the customer's right problem in the same way as their protocol does. By removing the watermark certification authority role and shifting its task to customer, we reduce the number of parties knowing the watermark being generated to the minimum, which is one. So, only customer knows the watermark generated. Since the generated watermark is sent to content provider in encrypted form and content provider does not know the corresponding private key, content provider does not have any knowledge about this watermark.

Upon receiving the encrypted watermark, content provider checks the validity of watermark by counting the number of different permutations to which it is not invariant. It is done in order to avoid the use of watermarks which enable customer to easily estimate. So, it is clear that content provider will not be benefited if he skips this step. Only if the watermark is acceptable, content provider will continue with the transaction by permuting the encrypted watermark, followed by embedding the permuted watermark into the content in encrypted domain. It is against content provider's interest not to perform the permutation in an appropriate manner as it might facilitate customer to estimate the embedded watermark more easily. Swapping the watermark with some other watermark will not be advantageous to content provider, either. A swap will only result in his inability to prove an illegal act of a customer. So, it is content provider's responsibility to choose a good random permutation and to insert the permuted watermark in the right manner. Content provider should also keep this permutation secret, lest it be known to customer.

In this protocol, it is impossible for content provider to reproduce copies of content containing a user's watermark since he has no knowledge about the user-generated watermark. He has his secret permutation and the encrypted watermark, but he does not have the private decryption key. Assuming the public-key cryptosystem and its infrastructure are secure, there is no way for content provider to decrypt it to obtain the watermark. Thus, content provider cannot frame a customer by distributing illicit copies of content containing the customer's watermark. For the same reason, a guilty customer cannot deny his unlawful deed by claiming that the unauthorized copy is created by content provider. On the other hand, customer will not be able to remove the watermark inserted without rendering the content useless for he does not know the permutation function applied to the generated watermark before embedding process. Neither content provider nor customer knows the exact watermark being embedded to the content. It is also against his own interest for customer to present a random watermark to the arbiter during dispute resolution process because it only causes himself to be considered guilty. Thus, it is guaranteed that content provider can prove a piracy act of a customer to a third party with no possibility of the accused denying his act. In other words, no framing and no repudiation requirements are satisfied.

Unfortunately, in this protocol, customers need to generate the watermark used in every transaction, which, up to certain degree, causes inconvenience to them. Moreover, they might need to repeat the process for few times if content provider rejects their watermarks. Although customers only need to communicate with seller in a transaction, they have to take part in dispute resolution process. If customers use the same public-private key pair in every transaction, they only need to keep one decryption key. However, the large amount of data encrypted using the same key

might help content provider to discover the private key. Therefore, customers need to store the list of decryption keys, each is needed to decrypt a content copy he purchased. In conclusion, customer's convenience is not provided by this protocol.

7.2 Bi-permutation Buyer-Seller Watermarking Protocol

In our first protocol, customers are required to generate the watermark used in every transaction, whereas content provider only needs to permute the generated watermark. Considering the limited resources that customers have and the inconvenience caused, we swap the tasks that content provider and customer must perform in our second protocol.

In this protocol, content provider creates the watermark to be inserted upon receiving a transaction request from a customer. The generated watermark will be then transferred to the customer for modification. The requesting customer only needs to perform bit permutation on every element of the watermark. In order to prevent content provider from guessing the permutation correctly, the length of watermark element should be designed to be long enough. Each element of the watermark should at least have 128 bits of precision to rule out the possibility of brute force attacks. It is against his own interest to skip this step or not to perform it in the right way. Therefore, it is customer's responsibility to choose a good permutation and hide the permutation safely.

The permuted watermark will be encrypted using the public key of the homomorphic cryptosystem and sent to content provider. Now, content provider has to group the indexes of all identical elements together. The grouping of the encrypted and permuted watermark is compared to that of the original watermark. This step is

done in order to prevent customer from swapping the watermark and presenting a random watermark. Since the same bit permutation is performed on all elements and encryption function is injective, these two groupings should be identical. If they are different, content provider can conclude that the customer has changed the watermark. So, by swapping the watermark with a random watermark, customer will not be able to cheat content provider for it will cause the transaction to be terminated. However, content provider will not be able to tell if customer swap the watermark with another watermark having the same grouping. It will only be discovered by an arbiter in a dispute resolution process as arbiter will repeat the permutation process and compare the result to what content provider has kept. It is, nonetheless, a useless effort done by the customer. It will not benefit him in any way. Watermarks with the same groupings also have the same set of permutations to which they are not invariant. Thus, changing the watermark with another one having the same grouping will help customer to estimate neither the permutation performed by content provider nor the watermark inserted to the content. We can therefore rule out this kind of swapping.

Once content provider validated the permuted watermark, he will permute the order of the watermark elements and insert it in encrypted form. In order to prevent customer from guessing this permutation correctly, we require the number of elements to be large enough. It is against content provider's interest not to perform the permutation in an appropriate manner as it might facilitate customer to estimate the embedded watermark more easily. Swapping the watermark with some other watermark will not be advantageous to content provider, either. It will only result in his inability to prove an illegal act of a customer. So, it is content provider's responsibility to choose a good random permutation and to insert the permuted

watermark in the right manner. Content provider should also keep this permutation secret, so that it is not known to the customer.

It is clear that content provider is only able to reproduce copies of content containing a user's watermark if he knows the bit permutation performed by the customer. However, this permutation is kept secret. Content provider has his secret permutation, the original watermark, and the encrypted bi-permuted watermark, but he has no knowledge about customer's permutation function. Assuming the public-key cryptosystem and its infrastructure are secure, there is no way for content provider to recover the bi-permuted watermark. Thus, content provider cannot frame a customer by distributing illicit copies of content containing his watermark. For the same reason, a guilty customer cannot deny his unlawful act by claiming that the unauthorized copy is originated by content provider. On the other hand, customer will not be able to remove the watermark inserted without rendering the content useless for he knows only the original watermark and his secret permutation, but not the seller's permutation function. Neither content provider nor customer knows the exact watermark being embedded to the content. Again, it is not advantageous for customer to present a random watermark or a different permutation function to the arbiter during dispute resolution process because it only causes himself to be considered guilty. Thus, it is guaranteed that content provider can prove a piracy act of a customer to a third party with no possibility of the accused denying his act. In other words, no framing and no repudiation requirements are satisfied.

In terms of customer's convenience, our second protocol is better than the previous protocol as customers only need to perform bit permutation on watermark elements, instead of generating the watermark itself. Additionally, they will never be

required to repeat the permutation process. The same as before, customers only need to communicate with seller in a transaction, but they have to take part in dispute resolution process. A single decryption key will only work if customers use the same public-private key pair in every transaction. However, the large amount of data encrypted using the same key might help content provider to discover the private key. Therefore, we can say that this protocol only satisfies the customer's convenience requirement partially.

7.3 Encryption-Based Buyer-Seller Watermarking Protocol

In order to further minimize the amount of work done by customer and to eliminate the possibility of customer swapping the watermark, we propose the third protocol in which all watermarking operations are done on the seller side.

In this protocol, upon receiving a request from a customer, content provider first generates the information sequence to be carried by the watermark. The only action that a customer has to do is to sign this sequence to prevent content provider from swapping it. If it is not signed, content provider can cheat by reversing the watermarking process. He can choose a random watermark to insert and then encrypt it. The ciphertext can be then used to find the corresponding information sequence. The random watermark is inserted to the copy of content sent to customer. This way, he knows what is exactly being embedded to the customer's copy and he can illegally distribute copies of content containing this random watermark. During a dispute resolution process, he can claim that this random watermark is the encryption of its ciphertext using customer's private key, and thus he successfully frames a customer. Therefore, it is very important to have customer verify and sign the information

sequence. It is disadvantageous for customer to skip this step or not to perform this step in the right way.

After receiving the signature of the sequence, content provider will substitute a number of bits of the information sequence to conceal it from customer. The number of bits substituted should be large enough to prevent customer from performing brute force attack to find the substitution. On the other hand, it should not be larger than the number of preserved bits. Otherwise, content provider can reverse the watermarking process as shown above to break the system. We can ask arbiter to check this number to avoid such attack. If the number of bits substituted is too large, arbiter must drop the charges on the accused customer. It is also very important to keep secret the substitution table and the positions of bits changed. Otherwise, customer will get full knowledge of the exact watermark inserted and this step is useless. Hence, content provider should ensure this step is carried out in the right way.

The substitution process will be then followed by content provider producing the corresponding watermark using this sequence of information. The generated watermark will be then inserted to the content that has been encrypted using the public key of the homomorphic cryptosystem. As the result, it is the generated watermark, encrypted with the private key, which will be inserted into the content. Content provider might want to encrypt the substituted watermark before embedding it into the content. However, it will cause him not to be able to prove a piracy act of a customer to a third party. Exchanging the watermark to insert with another watermark will also result in the arbiter's failure in detecting the legitimate watermark. Thus, content provider has no better choice than performing this step according to the convention.

In this protocol, the watermark is magically encrypted with the private key of the homomorphic cryptosystem by inserting it to an encrypted content. It is done without having to expose the key to content provider, who performs the insertion. Assuming the public-key cryptosystem and its infrastructure are secure, content provider has no way to obtain the private key, and therefore is unable to replicate the watermark inserted to the customer's copy. Although he is in charge of all watermarking process and knows the originally generated watermark, it is impossible for him to reproduce copies of content containing a user's watermark, which implies that he cannot frame an innocent customer. For the same reason, a guilty customer cannot deny his unlawful act by claiming that the unauthorized copy is originated by content provider. In other words, no framing and no repudiation requirements are satisfied.

On the other hand, customer will not be able to remove the watermark inserted without rendering the content useless because he knows nothing about the positions of substituted bits and seller's substitution table. Consequently, neither content provider nor customer knows the exact watermark being embedded to the content. During dispute resolution process, a customer might want to present a random bit sequence instead of the information sequence he received from content provider. Nevertheless, it is not advantageous to do so for it only causes himself to be considered guilty. Thus, content provider can definitely prove a piracy act of a customer to a third party.

During dispute resolution process of this protocol, costumer is required to expose the private key of the homomorphic cryptosystem to the arbiter. Thus, we require the public-private key pair used in every transaction to be distinct. With customer's convenience in mind, we let certification authority (CA) generate this pair

of keys on customer's behalf. Although both keys are sent to content provider, assuming the public-key cryptosystem and its infrastructure are secure, he will not be able to obtain the private key as it is encrypted using customer's public key. We also rule out the possibility of collusion between CA and content provider by assuming CA's honesty. Otherwise, there will be no secure public-key infrastructure.

It is easy to observe that our third protocol is better than the previous two protocols in terms of the amount of work that customer does. In this protocol, the only thing that customer must do is to sign the generated information sequence. In general, this sequence is much shorter than the watermark frames, causing the amount of work done by customer in this protocol to be significantly smaller than that in the previous protocols. Moreover, customers only need to communicate with seller during a transaction. However, similar to the other two protocols, they have to take part in dispute resolution process and a single decryption key will only work if customers use the same public-private key pair in every transaction, at the cost of helping content provider to discover the private key. Therefore, this protocol does not fully satisfy the customer's convenience requirement, although it is better than the previous two protocols.

Summary

Please refer to the following table for the comparison among our three protocols.

Table 3. Comparison among the three buyer-seller-watermarking protocols we propose.

Requirements	First (MW without WCA)	Second (Bi-permutation)	Third (Encryption-Based)
Traceability	Yes	Yes	Yes
No Repudiation	Yes	Yes	Yes
No Framing	Yes	Yes	Yes
Collusion Resistance	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes
Unlinkability	Yes	Yes	Yes
No Additional TTP (WCA)	Yes	Yes	Yes
No Unbounding Problem	Yes	Yes	Yes
Customer's Convenience			
• Not watermark generator	No	Yes	Yes
• Number of parties to communicate with	1	1	1
• No participation in dispute resolution	No	No	No
• Single decryption key in multiple purchases	No	No	No

We can see clearly from the table that our proposed protocols successfully solve customer's right problem, which indicated by the fulfillment of no repudiation and no framing requirements, without having to rely on any additional trusted third party.

The first protocol shifts the watermark generator role to the buyer, causing the seller to have a smaller amount of computation to perform. Both the watermark and the key pair used in a transaction are provided by the buyer. Thus, this protocol is suitable in a scenario where seller has a limited amount of resources and the distribution network is relatively larger. In contrast to the first protocol, the third

protocol requires the seller to perform the watermark generation process. Additionally, the seller has to handle the public-private key pair used in the homomorphic cryptosystem, as well. Therefore, we should only use this protocol in a situation where the amount of resources the seller has is relatively larger and the size of distribution network is quite small. The second protocol is proposed as the middle-of-the-road solution. This protocol distributes the amount of computation to the seller and the buyer more evenly. The seller is responsible of generating the watermark used, whereas the buyer is required to handle the cryptographic key pair. Consequently, this protocol makes a good choice in a case where both parties have medium amount of resources and the distribution network is of medium size.

8. CONCLUSION

Three new buyer-seller watermarking protocols were presented in order to solve the customer's right problem in the conventional digital fingerprinting without having to hinge on the trustworthiness of watermark certification authority (WCA). In these protocols, WCA no longer takes part in any stage of the protocols and watermark generation is performed by either customer or content provider.

The first protocol, a variant of Memon and Wong's protocol, combines permutation and privacy homomorphic cryptosystem to prevent both buyer and seller from knowing the exact watermark inserted, whereas the use of watermark invariant to permutation is avoided by a watermark validity checking. In the second protocol, customer's right problem is tackled by using two kinds of permutations and homomorphic encryption system, which are used to conceal the watermark embedded. The validity of watermark is guaranteed as it is generated by content provider. In the third protocol, substitution, instead of permutation, is used along with homomorphic cryptosystem to achieve secrecy of watermark inserted. The problem of invariant watermark does not exist since the protocol uses no permutation.

Our protocols successfully eliminate the user-framing and false implication problem. Simultaneously, they enable content provider to prove customer's piracy act to a third party with no possibility of guilty users denying his wrongdoing. Nevertheless, they fail to provide a full convenience to customers. Although now customers need to communicate with only one party, they have to participate in dispute resolution process. Moreover, they need to maintain a list of decryption keys used in all transaction they made. Finding a solution to these two shortcomings will be our future work.

BIBLIOGRAPHY

[1] Bloom, J.A. (2003). Security and Rights Management in Digital Cinema. In Proceedings of IEEE International Conference on Multimedia and Expo. (Baltimore, USA, July 6-9, 2003), ICME, 2003, pp. 621-624.

[2] Boneh, D. and Shaw, J. (1995). Collusion-Secure Fingerprinting for Digital Data. In Proceedings of the 15th Annual International Cryptology Conference: Advances in Cryptology. (Santa Barbara, USA, August 27-31, 1995), CRYPTO, 1995, pp. 452-465.

[3] Byers, S., Cranor, L., and Cronin, E. (2003) Analysis of Security Vulnerabilities in the Movie Production and Distribution Process. In Proceedings of the 2003 ACM Workshop on Digital Rights Management. (Washington DC, USA, October 27, 2003), DRM, 2003, pp.1-12.

[4] Chang, C.C. and Chung, C.Y. (2003). An Enhanced Buyer Seller Watermarking Protocol. In Proceedings of International Conference on Communication Technology. (Beijing, China, April 9-11, 2003), ICCT, 2003, pp. 1779-1783.

[5] Cheng, Q. and Huang, T.S. (2000). Blind Digital Watermarking for Images and Videos and Performance Analysis. In Proceedings of IEEE International Conference on Multimedia and Expo. (New York, USA, July 30- August 2, 2000), ICME, 2000, pp. 389-392.

[6] Cheung, S.C. and Curreem, H. (2002). Rights Protection for Digital Contents Redistribution over the Internet. In Proceedings of the 26th Annual International Computer Software and Application Conference. (Oxford, England, August 26-29, 2002), COMPSAC, 2002, pp. 105-110.

[7] Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P., and Reginelli, M. (2002). A New Chaotic Algorithm for Video Encryption. IEEE Transactions on Consumer Electronics, Vol. 48(4): 838-844.

[8] Choi, J.G. and Park, J.H. (2005). A generalization of an Anonymous Buyer-Seller Watermarking Protocol and Its Application to Mobile Communications. In Proceedings of the Third International Workshop on Digital Watermarking. (Seoul, South Korea, October 30 - November 1, 2004), IWDW, 2004, pp. 232-243.

[9] Choi, J.G., Sakurai, K., and Park, J.H. (2003). Does It Need Trusted Third Party? Design of Buyer-Seller Watermarking Protocol without Trusted Third Party. In Proceedings of the First International Conference on Applied Cryptography and Network Security. (Kunming, China, October 16-19, 2003), ACNS, 2003, pp. 265-279.

[10] Chong, J.C.N., van Buuren, R., Hartel, P.H., Kleinhuis, G. (2002). Security Attributes Based Digital Rights Management. In Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems: Protocols and Systems for Interactive Distributed Multimedia. (Coimbra, Portugal, November 26-29, 2002), IDMS/PROMS, 2002, pp. 339-352.

[11] Chu, H.H., Qiao, L., and Nahrstedt, K. (2002). A Secure Multicast Protocol with Copyright Protection. ACM SIGCOMM Computer Communication Review, Vol. 32(2): 42-60.

[12] Cohen, J.D. and Fischer, M.J. (1985). A Robust and Verifiable Cryptographically Secure Election Scheme. In Proceedings of 26th IEEE

Symposium on Foundations of Computer Science. (Portland, USA, October 21-23, 1985), FOCS, 1985, pp. 372-382.

[13] Conrado, C., Kamperman, F., Schrijen, G.J., and Jonker, W. (2003). Privacy in an Identity-Based DRM System. In Proceedings of the 14th International Workshop on Database and Expert Systems Applications. (Prague, Czech Republic, September 1-5, 2003), DEXA, 2003, pp. 389-395.

[14] Cox, I.J., Kilian, J., Leighton, T., and Shamoon, T. (1997). Secure Spread Spectrum watermarking for Multimedia. IEEE Transactions on Image Processing, Vol. 6: 1673-1687.

[15] Cox, I.J., Miller, M.L., and Bloom, J.A. (2002). Digital Watermarking. Morgan Kaufmann Publishers, San Francisco, 2002.

[16] Dittmann, J., Steinebach, M., Kunkelmann, T., and Stoffels, L. (2000). H2O4M-Watermarking for Media: Classification, Quality Evaluation, Design Improvements. In Proceedings of the 2000 ACM Workshops of Multimedia. (Los Angeles, USA, October 30-November 3, 2000), 2000, pp. 107-110.

[17] Emmanuel, S. and Kankanhalli, M. (2003). A Digital Rights Management Scheme for Broadcast Video. ACM Multimedia Systems Journal, Vol. 8(6): 444-458.

[18] Feigenbaum, J., Freedman, M.J., Sander, T., and Shostack, A. (2001). Privacy Engineering for Digital Rights Management Systems. In Proceedings of ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management. (Philadelphia, USA, November 5, 2001), DRM, 2001, pp.76-105.

[19] Ferrer, J.D. and Joancomarti, J.H. (2000). Simple Collusion-Secure Fingerprinting Schemes for Images. In Proceedings of International

Conference on Information Technology: Coding and Computing. (Las Vegas, USA, March 27-29, 2000), ITCC, 2000, pp. 128-132.

[20] Fetscherin, M. and Schmid, M. (2003). Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry. In Proceedings of the Fifth International Conference on Electronic Commerce. (Pittsburgh, USA, September 30-October 3, 2003), ICEC, 2003, pp. 316-325.

[21] Furht, B. and Kirovski, D. (2004). Multimedia Security Handbook. CRC Press, Florida, 2004.

[22] Gamal, T.E. (1985). A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In Proceedings of Advances in Cryptology. (Santa Barbara, USA, August 19-22, 1984), CRYPTO, 1984, pp. 10-18.

[23] Goi, B.M., Phan, R.C.W., Yang, Y., Bao, F., Deng, R.H., and Siddiqi, M.U. (2004). Cryptanalysis of Two Anonymous Buyer-Seller Watermarking Protocols and an Improvement for True Anonymity. In Proceedings of the Second International Conference on Applied Cryptography and Network Security. (Yellow Mountain, China, June 8-11, 2004), ACNS, 2004, pp. 369-382.

[24] Grimm, R. and Aichroth, P. (2004). Privacy Protection for Signal Media Files: A Separation-of-Duty Approach to the Lightweight DRM (LWDRM) System. In Proceedings of Multimedia and Security Workshop on Multimedia and Security. (Madgeburg, Germany, September 20-21, 2004), MM&Sec, 2004, pp. 93-99.

[25] Ju, H.S., Kim, H.J., Lee, D.H., and Lim, J.I. (2003). An Anonymous Buyer-Seller Watermarking Protocol with Anonymity Control. In Proceedings of the

Fifth International Conference on Information Security and Cryptology. (Seoul, South Korea, November 28-29, 2002), ICISC, 2002, pp. 421-432.

[26] Kirovski, D., Peinado, M., and Petitcolas, F.A.P. (2001). Digital Rights Management for Digital Cinema. In Proceedings of International Symposium on Optical Science and Technology. (San Diego, USA, July 29-August 3, 2001), SPIE, 2001, pp 105-120.

[27] Kundur, D. and Karthik, K. (2004). Video Fingerprinting and Encryption Principles for Digital Rights Management. Proceedings of IEEE, Vol. 92(6): 918-932.

[28] Kwok, S.H. (2003). Watermark-Based Copyright Protection System Security. ACM Communications, Vol. 46(10): 98-101.

[29] Lei, C.L., Yu, P.L., Tsai, P.L., and Chan, M.H. (2004). An Efficient and Anonymous Buyer-Seller Watermarking Protocol. IEEE Transactions on Image Processing, Vol. 13(12): 1618-1626.

[30] Lin, E.T., Cook, G.W., Salama, P., and Delp, E.J. (2001). An Overview of Security Issues in Streaming Video. In Proceedings of International Conference on Information Technology: Coding and Computing. (Las Vegas, USA, April 2-4, 2001), ITCC, 2001, pp. 345-348.

[31] Lin, E.T., Eskicioglu, A.M., Lagendijk, R.L., and Delp, E.J. (2005). Advances in Digital Video Content Protection. IEEE: Special Issue on Advances in Video Coding and Delivery, Vol. 93(1): 171-183.

[32] Lindkvist, T. (2000). Characteristics of Some Binary Codes for Fingerprinting. In Proceedings of the Third Information Security Workshop. (Wollongong, Australia, December 20-21, 2000), ISW, 2000, pp. 97-107.

[33] Linnartz, J.P., Talstra, J., Kalker, T., and Maes, M. (2000). System Aspects of Copy Management for Digital Video. In Proceedings of IEEE International Conference on Multimedia and Expo. (New York, USA, July 30- August 2, 2000), ICME, 2000, pp.203-206.

[34] Liu, Q., Naini, R.S., and Sheppard, N.P. (2003). Digital Rights Management for Content Distribution. In Proceedings of Australasian Information Security Workshop. (Adelaide, Australia, February 4-7, 2003), AISW, 2003, pp. 49-58.

[35] Liu, Z. and Li, X. (2004). Motion Vector Encryption in Multimedia Streaming. In Proceedings of the Tenth International Multimedia Modelling Conference. (Brisbane, Australia, January 5-7, 2004), MMM, 2004, pp. 64-71.

[36] Lookabaugh, T. and Sicker, D.C. (2004). Selective Encryption for Consumer Application. IEEE Communication Magazine, Vol. 42(5): 124-129.

[37] Lu, C.S., Chen, J.R., and Fan, K.C. (2004). Resistance of Content-Dependent Video Watermarking to Watermark-Estimation Attacks. In Proceedings of IEEE International Conference on Communications. (Paris, France, June 20-24, 2004), ICC, 2004, pp. 1386-1390.

[38] Lubin, J., Bloom, J.A., and Cheng, H. (2003). Robust, Content-Dependent, High-Fidelity Watermark for Tracking in Digital Cinema. In Proceedings of the International Society for Optical Engineering. (San Diego, USA, August 3-8, 2003), SPIE, 2003, pp. 536-545.

[39] Memon, N. and Wong, P.W. (1998). A Buyer-Seller Watermarking Protocol. In Proceedings of IEEE Second Workshop on Multimedia Signal Processing. (California, USA, December 7-9, 1998), MMSP, 1998, pp. 291-296.

[40] Memon, N. and Wong, P.W. (2001). A Buyer-Seller Watermarking Protocol. IEEE Transactions on Image Processing, Vol. 10(4): 643-649.

[41] Murphy, S. and Robshaw, M. (February 6, 2005). Public key Cryptography (II): Discrete Logarithm Based Systems. [Online]. Available: http://www.isg.rhul.ac.uk/msc/teaching/opt8/week8-2005.pdf

[42] Niederreiter, H. (1986). Knapsack-Type Cryptosystem Based on Algebraic Coding Theory. Problems of Control and Information Theory, Vol. 15(2): 159-166.

[43] Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Proceedings of Advances in Cryptology. (Prague, Czech Republic, May 2-6, 1999), EUROCRYPT, 1999, pp. 223-238.

[44] Qiao, L. and Nahrstedt, K. (1998). Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights. Journal of Visual Communication and Image Representation, Vol. 9(3): 194-210.

[45] Rivest, R., Shamir, A., and Adelman, L. (1978). A Method for Obtaining Digital Signatures and Public Key Cryptosystems. ACM Communication, Vol. 21: 120-126.

[46] Schonberg, D. and Kirovski, D. (2004). Fingerprinting and Forensic Analysis of Multimedia. In Proceedings of the 12th ACM International Conference on Multimedia.(New York, USA, October 10-16, 2004), MM, 2004, pp. 788-795.

[47] Senoh, T., Ueno, T., Kogure, T., Shen, S., Ji, M., Liu, J., Huang, Z., and Schultz, C.A. (2004). DRM Renewability & Interoperability. In Proceedings of the 2004 IEEE Consumer Communications and Networking Conference. (Las Vegas, USA, January 5-8, 2004), CCNC, 2004, pp. 424-429.

[48] Shieh, J.R.J. (2003). On the Security of Multimedia Video Information. In Proceedings of IEEE 37th Annual 2003 International Carnahan Conference on Security Technology. (Taipei, Taiwan, October 14-16, 2003), ICCST, 2003, pp. 51-56.

[49] Simpson, S. (September 20, 1999). PGP DH vs. RSA FAQ. [Online]. Available: http://www.scramdisk.clara.net/pgpfaq.html

[50] Skraparlis, D. (2003). Design of an Efficient Authentication Method for Modern Image and Video. IEEE Transactions on Consumer Electronics, Vol. 49(2): 417-426.

[51] Tistaert, L. (March 2005). Shrek 2 Box Office. [Online]. Available: http://www.leesmovieinfo.net/wbotitle.php?t=2501&section=2&format=3&order_by=dor%20ASC,%20d_period%20ASC

[52] Tosun, A.S. and Feng, W. (2000). Efficient Multi-layer Coding and Encryption of MPEG Video Streams. In Proceedings of IEEE International Conference on Multimedia and Expo. (New York, USA, July 30- August 2, 2000), ICME, 2000, pp.119-122.

[53] Tosun, A.S. and Feng, W. (2001). On Error Preserving Encryption Algorithms for Wireless Video Transmission. In Proceedings of the Ninth ACM International Conference on Multimedia.(Ottawa, Canada, September 30-October 5, 2001), MM, 2001, pp. 302-308.

[54] Trappe, W., Wu, M., and Liu, K.J.R. (2002). Collusion-Resistance Fingerprinting for Multimedia. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. (Orlando, USA, May 13-17, 2002), ICASSP, 2002, pp. 3309-3312.

[55] Veerubhotla, R.S., Saxena, A., and Gulati, V.P. (2002). Reed Solomon Codes for Digital Fingerprinting. In Proceedings of the Third International Cryptology Conference. (Hyderabad, India, December 15-18, 2002), INDOCRYPT, 2002, pp. 163-175.

[56] Wessely, U., Eichner, S., and Albrecht, D. (2003). Watermarking of Analog and Compressed Video. In Proceedings of International Workshop for Technology, Economy, Social, and Legal Aspects of Virtual Goods. (Ilmenau, Germany, May 22-24, 2003), Virtual Goods, 2003, pp. 20-26.

[57] Zeng, W. and Lei, S. (1999). Efficient Frequency Domain Video Scrambling for Content Access Control. In Proceedings of the Seventh ACM International Conference on Multimedia. (Orlando, USA, October 30-November 5, 1999), MM, 1999, pp. 285-294.