Universitat Politècnica de Catalunya
Department of Telematics Engineering

Ph.D. Dissertation

Real-Time Pay-per-View of Protected Multimedia Content v:2.0

Author:
Antoni Martínez Ballesté

Advisors:
Prof. Josep Domingo Ferrer
Dr. Miquel Soriano Ibáñez

A dissertation
submitted to the Department of Telematics Engineering
and the Committee on Graduate Studies
of Universitat Politècnica de Catalunya
in partial fulfillment of the requirements
for the degree of Doctor.

June 2004

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor.

Prof. Josep Domingo Ferrer
(Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor.

Dr. Miquel Soriano Ibáñez
(Advisor)

Approved by the University Committee on Graduate Studies:

Abstract

Dir que aquesta tesi tracta de comerc electronic payperview de video. La primera part es mes aviat practica, unicast, basada en el projecte streamobile. La segona part tracta els temes de pagament i proteccio del copyright en l'entorn multicast.

La tercera part presenta un metode per tal que (SDC).

La primera part es mes aviat tecnica, la segona teorica, la tercera es de vital importancia per a les aplicacions e-commerce.

This thesis tells you all you need to know about...

Agraïments

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Situation and objectives

In the last years, the amount of multimedia services has grown exponentially due to the great development of the Internet, multimedia-enabled devices and wireless or fixed Internet access. Several of these services are obtained in exchange of a payment. Moreover, as new mobile communication technologies and home user bandwidth Internet access are becoming broadly available, there is urgent pressure to populate them with services that provide returns for the huge investments made by telecom operators. Services around video transmission are expected to play a key role: news broadcasting, video on demand, music downloading, movie channels, on-line gambling, etc.

Electronic Commerce, or *e-commerce*, can be defined as the action of buying and selling using Internet, specially the World Wide Web. Electronic mobile commerce, or *m-commerce* is the part of e-commerce related to the variety of e-commerce services that are accessed with mobile devices, such as

Table 1.1: 2002 estimated U.S. trade losses (in million USD) due to copyright piracy in 56 countries (source IIPA).

Sector	Estimated losses
Motion pictures	1322.3
Music	2142.3
Business software	3539.0
Entertainment software	1690.0
Books	514.5
TOTAL	9208.1

cellular phones, handheld computers, etc.

On the other hand, the Internet and the e-commerce allow trading with *digital* goods, for instance a song digitally encoded. Note that the latter can be transmitted from provider to the customer merely using the network. The so-called *pure* electronic commerce refers to this kind of trading.

The main purpose of most of multimedia content is to entertain a large audience who pays for accessing to it; moreover, the *intellectual property* and the copyright of the works traded must be protected. In fact it can be affirmed that, in pure e-commerce, *products can reach millions of people*; the bad news are that one of the properties of digital data is that they can be *easily copied without any quality degradation.* This is a serious threat to e-commerce of multimedia and to the intellectual property of digitally delivered content.

Internet sites for free exchange of music have been in operation for a while, and the efforts by the industry to legally prevent such an activity have

been rather unsucessful so far. In fact, according to figures given by Frankfurter Allgemeine Zeitung (15 August 2003), sales of music CDs in Germany decreased by 16.3 percent in 2002. As a consequence of the replacement of analog video by DVD and the increase of Internet bandwidth and computer disk space, the movie industry is likely to face the same problems as the music industry. On the U. S. side, the International Intellectual Property Alliance (www.iipa.com), a private-sector coalition formed in 1984 and representing U.S. copyright-based industries, has issued a 2003 Special 301 Report on Global Copyright Protection and Enforcement in which deficiencies in the copyright regimes of 56 countries are identified that caused U.S. copyright industries to lose more than 9.200 million USD in trade due to piracy. Table 1.1 lists the 2002 trade losses for five copyright-based industry sectors.

Finally, websites and Internet servers generate *statistical databases* containing surfing habits, which can be studied by third parties in order to improve revenues and also provide a better service to their users. Note that the privacy of the customer must be protected, *i.e.* nobody wants his identity to be disclosed if he has been accessing adult pay-per-view content and the frequency at he has been doing it.

In this thesis we focus on the electronic commerce of protected streamed multimedia content, covering these issues:

Payment model: in streamed content payments should be made in fine-grain manner, that is paying as the content is being accessed.

Copyright protection: illegal copies of the content sold should be securely identified.

Customer privacy: web statistics databases should be made *transferable* before being sent or sold to third parties.

Multicast content distribution: as streamed media can reach a large audience, multicast communications are likely to be widely used. In such communications, paying for the content and deploying copyright protection techniques is not straightforward.

Part of this thesis is related to the STREAMOBILE project (see Section 3.1). Thus, some implementation has also been carried out.

1.2 Structure of this thesis

This thesis is organized as follows:

Chapter 2 presents a background on the main concepts and technology around this thesis. In the first section, a scenario is sketched. Video streaming on the Internet, payment issues, copyright protection for digital video, multicast communications and statistical data protection are briefly described in successive sections. Finally, an overview of some current Internet platforms for e-commerce of video is addressed.

In Chapter 3 the results of the research done related to the STREAMOBILE project are presented. More precisely, a prototype for video payment over the Internet toward mobile devices is described. In this chapter, approach for copyright protection of digital media using copy detection techniques is proposed. Finally, the requirements that an Internet-based platform for pay-per-view videos should meet are depicted.

Chapter 4 presents the research carried out in order to extend the payment and copyright protection issues to a multicast scenario. Note that, unlike the previous chapter, Chapter 4 mostly covers theoretical aspects. We present a set of protocols in order to aggregate information in a multicast tree (from the leaves to the source) in a secure manner. A proposal for aggregatable payments is also presented in this chapter, as well as a critique to existing proposals for copy detection in multicast.

Finally, Chapter 5 addresses a method in order to make the statistical data collected Internet services such STREAMOBILE transferable. In this chapter we describe a procedure in order to generate transferable data with linear cost.

Concluding remarks and some guidelines for future research are given in Chapter 6.

Chapter 2

Background

This chapter contains a background in e-commerce of streamed content, centered in pay-per-view of video streams.

2.1 A scheme for multimedia e-commerce

An approach of a multimedia e-commerce system can be depicted using these elements:

- The **content** is the data to be retrieved and watched or listened. From now on and for sake of simplicity, content will be referred to a video stream.
- The **customer** is the buyer of the content. She must be able to watch the content she has paid for. She owns a *decoder* in order to carry out this task. She should be able to store the content, but digital copies should be prevented from illegal distribution. Some services will be

addressed to a small audience. Opposite to that, broadcasts of live events are addressed to thousands, even millions of people.

- The **merchant** or **content provider** is the business offering the multimedia content. The merchant receives several payments from the customer. On the other hand, the merchant can collect statistical data from her customers, in order to improve the services. These data can be sold to third parties.

- The **electronic payment** (e-payment) is something the customer gives to the merchant in order to get the content. There are several e-payment proposals in the literature. An e-payment system, such as an electronic coin, requires the same level of security as real-world payments, say cash.

- The **carrier network** is the entity where goods travel from merchants to customers, and payments go from customers to merchants. Some multimedia e-commerce platforms use private networks which can easily be eavesdropped. Open networks such as Internet are also widely used as carrier networks.

2.2 An approach to Internet video streaming

Downloading entire movies from the Internet usually takes a long time even in the presence of compression. Besides, video downloading precludes the delivery of live broadcasting. Streaming is better than downloading in order to avoid long waiting times and enable transmission of live events. With streaming, the client can view the contents as soon as they are received. In

fact, streaming has become the dominant way to deliver continuous media over the Internet.

Streamed data must be presented to the viewer at a specified bit rate, whereas the network delivering the content has a different and fluctuating bandwidth. A common solution is to use an internal *buffer* which carries out an isochronous transmission. The buffer will not deliver any data to the video decoder until it has accumulated enough data to guarantee a constant (or specific and dynamic) bit rate.

Digital video is greedy in terms of storage requirements. For example, a few minutes of high-quality video may need a few gigabytes to be stored. The MPEG [Mpeg03] formats are one possibility to mitigate this problem through compression.

2.2.1 Content compression

Next, some video formats in the MPEG family are briefly described:

MPEG-1. Very good quality, but fairly large files, therefore not really suitable for the web. It *was* used mainly for CDROM and PC multimedia.

MPEG-2. MPEG-2 video is not suitable for use on the web. Given the good quality and high bitrates it is maily used in DVD and broadcast digital television. It supports interlaced video and VBR encoding.

MPEG-4. It is intended for video conferencing, Internet distribution and similar applications using low bandwidths, but is also a contender for HDTV.

An MPEG stream usually contains two multiplexed streams, audio and video. From now on, a video stream is referred only to the visual content (both audio and video streams form a *system* stream). Note that a video stream can be defined as a sequence of uncompressed still images (frames).

MPEG compression divides the frames into *intra coded* frames (I-frames), *predicted* frames (P-frames) and *bidirectional* frames (B-frames). I-frames use DCT encoding only to compress a single frame without reference to any other frame in the sequence. Typically I-frames are encoded with 2 bits per pixel on average. For random playing of MPEG video, the decoder must start decoding from an I-frame not a P-frame. I-frames are inserted every 12 to 15 frames. P-frames are coded as differences from the last I or P frame. The new P-frame is first predicted by taking the last I or P frame and predicting the values of each new pixel. Finally, B-frames are coded as differences from the last or next I or P frame. B-frames also use prediction, but require both previous and subsequent frames for correct decoding.

2.2.2 Content transmission

Fixed access to broadband Internet via cable or ADSL is already reaching a substantial and increasing share of the European population. Mobile access to broadband Internet via UMTS and/or wireless IEEE 802.11 LANs and IEEE 802.16 MANs is looming or is already there in some urban areas. Next, there is shown a brief overview of wireless and fixed technologies. Figure 2.1 shows the increasing value of broadband wireless services.

GSM. With 2nd Generation wireless communications (2G), data applications are introduced in the cellular phones market. Internet browsing

using monochrome WAP services is feasible. Bandwidths of about 15 Kbps can be reached with GSM.

GPRS. The Internet turns the focus toward data transmission. Enhanced multimedia messages and streaming video are possible with 2.5G cellular phones. Up to 144 Kbps can be reached for data downloading. Tariffing is volume-based.

UMTS. Enhanced multimedia capabilities and streaming video capabilities are increased. Standards are created to allow universal access and portability across different device types. Up to 384Kbps can be reached currently with 3G.

WLAN. IEEE 802.11 standard provides broadband access to the Internet, up to 11Mbps. The mobile user connects to a hot-spot (free or for subscribers) in order to get access. Several sub-standards may provide higher bandwidths. The standard IEEE 802.16 allows building a wireless metropolitan area network.

ADSL. Broadband fixed access to the Internet, with bandwidths up to 2Mbps.

It can be observed that mobility of users and devices, and the retrieving of data and multimedia from the Internet or P2P networks is supported by a wide variety of standards. Note that, in volume-based tariffing (as in GPRS or UMTS), it is essential to strike a tradeoff between video quality and bitrate; optimal compression codecs must be used for that.

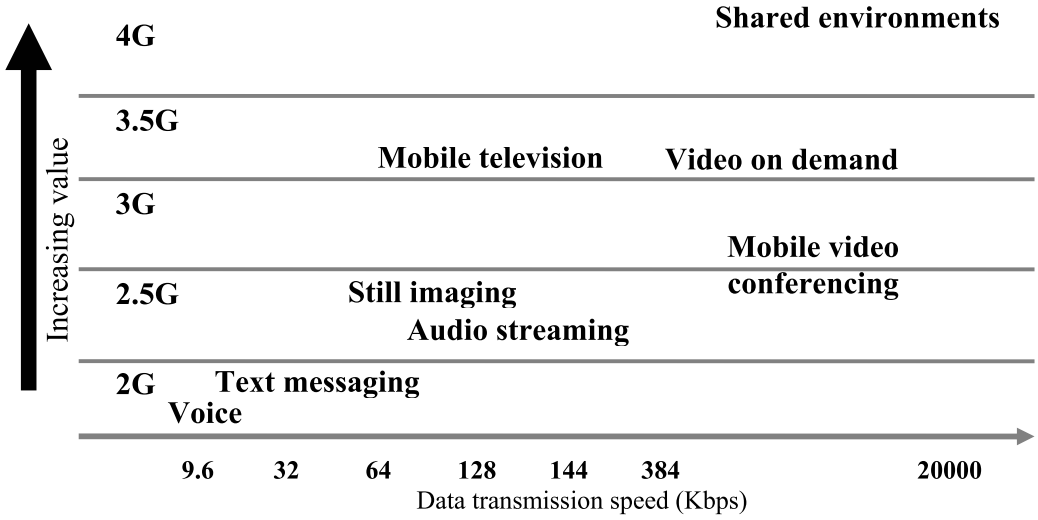

Figure 2.1: Services for wireless cellular phones.

2.2.3 Content playing

Adapting the streamed content to fit the device through which the media is streamed and watched is essential in order to provide a certain quality of service. A broadband user does not receive any quality of service if she receives a video which is too compressed. On the other hand, a customer with a small display should not be sent high-resolution videos. Thus, three kinds of customers can be considered depending on their equipment:

PC users. Desktop PC users would like to enjoy a full-screen video. Several broadband access providers offer full-screen Internet-based video-on-demand systems. But, in most cases, a 25 frames/second full-color 320x240 video already has enough quality. Either freeware and commercial software players, as well as open or proprietary decoders can be used on a PC.

PDA users. Multimedia-enabled personal digital assistants usually have a display of 240x320 pixels, and about 32K or 64K colors. Market forecasts predict shipments up to 60 millions of units in 2007, so providing access to video contents from an Internet-enabled PDA is more than justified. Several PDA manufacturers sell GPRS-enabled PDAs, which can also be connected to the Internet through a wireless LAN point of access. This year, some UMTS services have begun being offered through pluggable cards. On the dark side, streaming is mainly restricted to proprietary codecs of the operating system manufacturer.

Smartphone users. Mobile phone manufacturers and operators expect a growth of cellular data services. While the development of data services has steadily increased in Japan over the last years, the European market has stayed relatively flat since the relative failure of WAP[Hunt02]. In recent months, a plethora of multimedia enabled smartphones have made their way to the market. Multimedia messaging through GPRS connections are part of the market. Instead of using the low data transmission speeds of second-generation mobile networks, these phones use (or will use) faster data accesses, such as GPRS, UMTS or a combination of UMTS and GPRS. In such devices, streaming is even more *codec-restricted* than in PDAs.

It must also be considered that there will be customers that will use public devices to obtain these pay-per-view services. The use of smart cards in order to identify a customer (when at home or when traveling and using public Internet browsers) is a key issue in these systems.

2.3 Pay-per-view of multimedia content

Pay-per-view is currently a word related to television. Digital TV platforms allow viewing a pack of channels by paying a fixed amount per month. The buyer can view the so-called pay-per-view events (football matches, movie TV premieres, etc.) if she pays in advance for the full content. A problem can arise if the buyer does not like the movie at all, or if she has simply selected the wrong item to buy and view: she is losing a part of her money, because she has already paid for the content she is not going to view.

In contrast to pay-per-view, *pay-as-you-watch* (or *pay-as-you-listen*) [Domi02a], is defined as paying in real-time, *i.e.* paying as the content is being watched (or listened). The customer joins a pay-as-you-watch Internet service and makes a small value payment, say every minute. The video server sends a block of data to the buyer only if she has made the corresponding payment. When the buyer stops paying (switches off the wallet software) she has only paid for the minutes she has viewed so far. This kind of fine-grained pay-per-view services are likely to attract many potential customers of mobile video-based services.

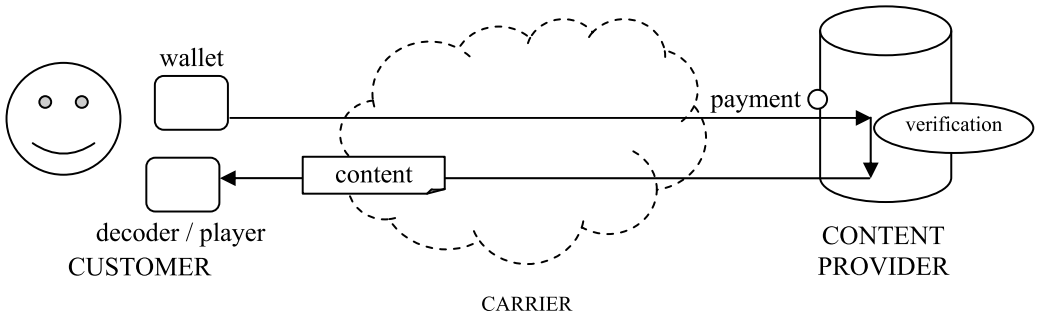

Figure 2.2: A scheme for real-time pay-per-view, or *pay-as-you-watch*.

Figure 2.2 depicts a pay-as-you-watch service. The customer owns a wallet with e-coins which can be sent to the content provider. After verification, content provider sends the data to the decoder of the customer, who is able to play the content.

2.3.1 Non-verifiable and verifiable low-value payments

A first and easy solution for implementing a pay-as-you-watch service is to assume an agreement between the content provider and a telecommunications carrier; the buyer is then charged by the carrier for the time she has been enjoying the service and the carrier transfers payment to the content provider. Also concerning the payment to the carrier, access to the content can be paid using a high-rated SMS message. Due to the universal and free access inherent to Internet, these *carrier-provider* schemes does no longer work.

On the Internet, there is no carrier (or there are many of them), so contents should be paid directly to the provider as they are being received.

Note that the payment system can lead to verifiable schemes, in which the source obtains a proof of correct reception by the receiver(s). On the contrary, in non-verifiable schemes, receiver non-repudiation is not guaranteed, so there must be a trust relationship between source and receiver(s).

A *non-verifiable* pay-as-you-watch solution is one in which the provider obtains no proof of correct content reception by customers. Even without such a proof, the provider is aware of the content received by each customer: the reason is that the content is sent using individual connections. Therefore, the provider can charge the customer for the minutes she has received. Thus,

the non-verifiable solution would consist of the provider metering the contents sent to each customer and thereafter billing the customer accordingly. The main drawback of non-verifiable systems is the need for trust between provider and customers:

- On one hand, the customer must trust the service provider: the customer must believe that the service provider will not charge her for contents she has not received.
- On the other hand, the provider cannot prove a subscriber is receiving a certain content. In this way, a dishonest customer could repudiate having received a certain content. After repudiation, the dishonest customer could claim her money back and/or redistribute the received content without being punished.

The transaction costs of standard electronic payments are usually considered too high for small amounts such as those required for real-time payment of small time slots. These transaction costs can be split into communication and computation costs, the latter being caused by the use of complex cryptographic techniques such as digital signatures. Micropayments are electronic payment methods specifically designed to keep transaction costs very low and to provide *verifiable* payments.

In most micropayment systems in the literature, computational costs are dramatically reduced by replacing digital signatures with hash functions. At a rough estimate, hashing is about 100 times faster than RSA [Rive78] signature verification and about 10,000 times faster than RSA signature generation. Several schemes have been proposed in the literature, such as μ-*i*KP [Haus96], PayWord and MicroMint [Rive95], PayTree [Jutl96] and

spending programs [Domi99]. Commercial implementations include IBM's MiniPay[Mini] and Digital's MilliCent [Glas95], the latter being currently out of deployment.

PayWord micropayments

PayWord is designed for application in which customers engage in many repetitive micropayment transactions. Some examples could be non-free websites or the aforementioned pay-as-you-watch.

In PayWord [Rive95], the customer (payer) establishes an account with a broker who gives her a certificate that contains the customer's identity, the broker's identity, the customer's public key, an expiration date and some other information.

A *payword* is a hash chain and each value of it is meant to be a coupon or coin. One-wayness of the hash function $\mathcal{H}$ prevents the scheme from false coin minting. For example, for n coins, the customer creates a chain of paywords $w_1, w_2, \ldots w_n$, picking the last payword w_n at random and then computing

$$w_i = \mathcal{H}(w_{i+1})$$

for $i = n-1, n-2, \ldots, 0$. Here w_0 is the root of the payword.

When the customer wants to start making micropayments (to start receiving contents), she sends to the merchant a commitment to a chain. The commitment includes the merchant's identity, the broker certificate, the root of the chain, the current date, the length of the chain and some other information. In this scheme, the broker certificate certifies that the broker will

redeem any payment that the customer makes before the expiration date, and the customer commitment authorizes the broker to pay the merchant.

Note that, once a coin w_i is sent to the merchant, the latter only needs the previous coin in the chain w_{i+1} in order to verify it only computing

$$w_i =^{?} \mathcal{H}(w_{i+1})$$

On the other hand, the merchant only needs to keep the last coin received from the customer.

Providing anonymity

Low value e-payment systems should provide the same security properties of real-world coins, that is non-forgery and anonymity. Although the latter can be largely discussed (see [Odly03]), providing anonymity to e-payment systems may be attractive for customers.

In [Chaum82], the anonymity of the e-cash scheme presented is carried out by means of a *blind signature*, but the high cost of these operations and the communication cost due to the online verification that is needed to countermeasure double-spending, make this approach unsuitable for micropayments. In [Brand93], an offline verification scheme is presented. In this proposal, the identity of the payer is only disclosed if any coin has been double-spent.

Providing anonymity to PayWord would be somehow straightforward: a blind signature should be used for signing commitments; even though, the fact that every coin is arriving from a certain IP could be used for the merchant to disclose the identity of the customer.

Nevertheless, the computing power of current desktop PC is likely to make some proposals of anonymous digital cash suitable for low-value payments.

2.4 Copyright protection of multimedia content

Preventing unlawful copying requires either some hardware enforcement, copying licenses or some cryptographic software routines [Eski03], but these techniques are usually broken by hackers in a relatively short time. The DVD anti-copy technique was broken some years ago [DeCS], so anyone can obtain a digital copy simply using a freeware application downloaded from a public website. On the other hand, Microsoft Media Player's Digital Right Management has recently reported to be broken [Micr03].

Thus, *copy detection* techniques appear as the main solution for protecting the copyright of content in electronic format. The idea here is to track who made illegal copies rather than preventing these.

Fingerprinting techniques [Wagn83] use a *watermarking* [Kayz00] system in order to embed a mark that identifies the customer who has bought a certain copy of the content. The alterations applied to the content for mark embedding must be imperceptible. This means that the quality of the content should not decrease after watermarking. Moreover, watermarking techniques should be robust to content manipulation: for example, a mark should be recoverable from a JPEG picture even it has been rotated after marking. Several robust watermarking techniques have been proposed for pictures [Sebe00, Sebe01, Domi02, Liu02, Eski03a] and audio [Bone96].

Watermarking is used to embed a copyright message (*e.g.* who is the content author or owner): Thus, all copies sold are identical. On the contrary, in fingerprinting, every copy sold is slightly different from the other copies because the mark embedded depends on the identity of the customer.

2.4.1 Watermarking for digital video

Next paragraphs are focused on watermarking for MPEG video. Several MPEG watermarking techniques can be found in the literature, given the non-proprietary nature of MPEG standards, whereas not much information can be found for commercial video encodings such as Microsoft's WMV, Apple's QuickTime or RealNetwork's RealVideo. A video watermarking scheme should be robust to resampling, *i.e.* decompressing the video and compress at a different bitrate or quality, and to frame suppression. Finally, the mark itself should not increase the average bitrate of the watermarked stream.

There are two different approaches in order to embed a watermark in a video stream:

- Every frame is watermarked using robust picture watermarking techniques. In this approach, the video stream is decoded and then re-encoded, just after the mark has been embedded in the frames. This is a high time-consuming technique, not suitable for real-time watermarking.

- It can be assumed that the distributor or broadcaster of digital video will usually store the video in compressed format. Thus, if the watermark is embedded using compressed domain processing, the stream can be watermarked as it is being sent to the customer. Embedding

techniques described in [Hart98, Lang01] are as complex as MPEG decoding. These techniques are mostly based on manipulating the DCT of I-frame blocks, hence they operate on the frequency domain.

Low bitrate video watermarking

Although several video watermarking techniques have been proved to be robust and imperceptible, most of them are successfully over medium or high bitrates, as in the case of MPEG-1 or MPEG-2 streams. Providing a robust and imperceptible watermark for low bitrate encodings, such as MPEG-4 is currently an open issue. In such encodings, the mark embedded is likely to degrade the visual quality and increase the bandwidth of the stream.

In [Sety01], authors use and extended version of DEW algorithm [Lang01] in order to increase the watermark capacity and to decrease the distortion of the marked video stream. Both issues are reached embedding the watermark not just in a frame, but spreading it the temporal dimension. As a result, P-frames are vulnerable to reencoding, and the whole proposal does not resist severe frame deletion. On the other hand, results are quite different from quiet scenes to those with a lot of motion.

2.4.2 Collusion-resistant fingerprinting codes

Let the buyer of a video or picture be identified using a bitstring embedded in the content using a watermarking technique. The so-called *collusion attacks* allow a set of dishonest customers (colluders), who bought their own copy of the same content, to create a new copy whose mark does not identify any of the colluders. As each copy carries a different embedded mark, a set of

dishonest buyers can compare their copies and generate a new content whose mark does not allow identification of any of the colluders. This copy is to be redistributed illegally. In [Bone95], the *marking assumption* is introduced which states that, in a collusion attack, only *detectable positions* of the mark are alterable. Detectable positions are those in which colluders find some difference when comparing their copies bit by bit. This amounts to assuming that the underlying watermarking system is ideally robust.

In order to prevent collusion attacks from being successful, marks (*i.e.* the bitstrings) must be designed to be collusion-resistant. Thus, marks must be codewords of specially built codes. Several collusion-resistant fingerprinting codes have been described [Bone95, Domi00, Sebe02, Fern02].

The classical approach of Boneh and Shaw [Bone95] propose a construction for building binary codes secure against collusions of up to c colluders from a group of N buyers. Parameters c and N are chosen together with a security parameter $\epsilon > 0$ which is the maximum acceptable probability of failure in colluder re-identification. Codewords have length

$$l = 32c^4 \log(2N/\epsilon) \log(8cL/\epsilon) \tag{2.1}$$

where $L = 2c \log(2N/\epsilon)$. [Sebe02] presents shorter codes, providing security against up to 3 colluders. It must be noticed that a colluder must be a buyer.

2.5 Multicast communications

If a source is to communicate with n receivers, one could naively think of using n unicast communications (which results in the source being an output bottleneck) or one broadcast channel (which results in the entire network being flooded). Both solutions are wasteful in terms of bandwidth. A better option to avoid increasing network congestion is for receivers to join a multicast group, *i.e.* a set of receivers that are interested in receiving a particular kind of information, and have the content sent to them by using their multicast group IP address [Mill99]. Multicast communications are likely to be widely used for distributing multimedia.

The multicast approach is less versatile than the unicast distribution of video: a group of customers must watch or listen to the same content at the same time. Thus, multicast distribution is suitable for large scale live events or near-on-demand video services.

Efficient multicast design and implementation is currently an open issue. The multicast task is carried out by multicast routers, which join previously established multicast groups identified by a multicast IP address. These routers are capable of sending the data flow to multicast group.

The basic tasks to be performed in multicast communication are: advertise the multicast session, manage group enrollment by the customers who want to receive the stream and, concurrently to group enrollment, build the multicast routing tree. Several multicast protocols have been proposed in the literature, such as MOSPF[Moy94], PIM-DM[Deer98], PIM-SM[Deer96].

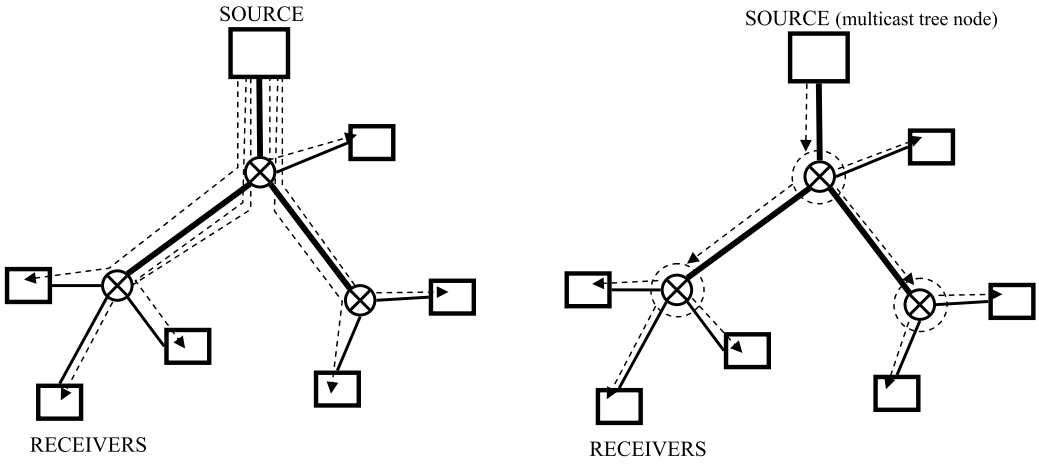

Figure 2.3: Unicast (on the left) and multicast (on the right) for content transmission.

Security in multicast

Multicast routers form a group that receives a multicast data stream. The router will possibly send the info to a hub that floods all its output connections, thus making the information reach every node in the subLAN, including nodes whose customers have not paid for the content. Cryptography should be used to prevent cheaters from being able to view the content by using packet sniffers. Customers in the multicast group have a *decoding key* to be able to decode the content they receive, just like in pay-per-view digital TV distribution.

In a pay-as-you-watch scheme, legitimate customers are those who pay every period t, that would typically last a few minutes. When a customer does not pay, she will be considered non-legitimate; in this case, a *rekeying* procedure will start which consists of distributing a new decoding key

to every remaining legitimate customer. As a result, the removal of a group member would involve as many unicast transmissions as legitimate customers remain in the group, leading to a high complexity in large tree. Fortunately, rekeying reaches a maximum cost of $O(\log n)$ when using tree structure controls[Snoe01].

2.6 Customer data privacy

Content providers and e-commerce sites collect huge quantities of statistical data from the transactions. Moreover, some websites send to customers *tracking cookies* in order to collect some of such data. As a result, many websites are generating millions of transaction data that can easily be used for datamining purposes. Thus, when these online sites suggest specific items to customers using pop-up banners, based on their past purchases or even just web surfing, these sites are using a combination of CRM and datamining to not only increase their revenue, but also provide service to the customers [Riad04]. This statistical data is high valued given that it can be sold to other vendors in order to improve their commerce models.

When this data is analyzed, *customer privacy* must be protected somehow. In that way, several statistical disclosure control techniques must be applyied to that set of data before being transfered or sold.

Statistical databases can either contain tabular data or individual data (microdata). Microdata can be continuous (or quantitative), *e.g.* age, weight and money spent, or categorical, for instance sex or hair color. When a microdata set is to be released for public use, confidentiality must be ensured. In that sense, the purpose of Statistical Disclosure Control (SDC) techniques

is twofold: on one hand, SDC methods must prevent the identity of the individual respondent from being disclosed; on the other hand, the published set of data should preserve as many statistical properties as possible from the original set.

One possibility for protecting a microdata set is to use a *masking method* (*e.g.* additive noise, microaggregation, etc., cf. [Domi01b]) to transform original data into protected, publishable data. An alternative to masking the original data is to generate a new data set (a *synthetic* data set) not from the original data, but from a set of random values that are adjusted in order to fulfill certain statistical requirements. A third possibility is to build a *hybrid* data set as a mixture of the masked original values and a synthetic data set [Dand02a].

2.7 Internet video pay-per-view initiatives

Finally, some significant initiatives of Internet distribution and e-commerce of streamed video are described:

- CinemaNow [Cinemanow] was founded in 1999 by a group of companies headed by Microsoft. If offers over 3000 films on a pay-per-view and subscription basis and is being accessed by 1 million customers per month. Licensors distributing their contents via this service include 20th Century Fox, MGM, Warner Bros. and others. Films are available for streaming and download. This web site uses a proprietary DRM. It is quite unclear what countermeasures (if any) exist to prevent a malicious client side from storing and redistributing the viewed contents in unprotected form. In addition to that, pay-per-view consists

of paying for the whole film before viewing it.

- Movielink [MovieL] is a more recent initiative that was launched in 2002 after several years of delay and discussion, mainly due to piracy threats. This on-line movie download service was established by Warner Bros, Paramount, Universal Studios, Sony Pictures Entertainment and Metro-Goldwyn-Mayer. A limited number of commercial films (about 170) are currently being featured and pay-as-you-watch is not offered. The service is restricted to the United States, which seems to suggest that an important part of anti-piracy measures are of a legal nature.

- Europe Online [EuroOn] are a privately held company based in Luxembourg. With multiple full Satellite Transponders and 180 Mbps fiber optic Internet Backbone, Europe Online are operating the first hybrid terrestrial/satellite Broadband Internet Network in Europe. Based on the DVB and IP open standards, the Europe Online network is compatible with all satellite, cable, ADSL, GPRS and other telecoms-based broadband distribution systems. This innovative hybrid platform allows consumers across Europe to receive an Internet-based interactive entertainment via their PCs and televisions. Subscribers can enjoy the best of interactive entertainment today, including streamed video and audio as well as a broad range of digital products for download, such as software, games, music, movies and other digital content in minutes without maintaining a telephone connection. Their Mediathek product provides live streams and video on demand downloads, which are sent to the customer via a multicast network of broadband routers. Copyright protection of the content is not specified, whereas payments are performed using a credit card. PayTV channels can be received via

a special decoding hardware attached either to TV or PC. The latter uses a special player software in order to decode the content.

- NDS [NDS] is a leading European supplier of open end-to-end digital systems and solutions for the secure delivery of entertainment and information to televisions and IP devices. NDS enables broadcasters, network operators and content providers to profit from the deployment of digital TV technologies including innovative interactive applications and personal TV, secure broadband and datacasting solutions. Around the world, NDS VideoGuard conditional access systems secure service revenues of more than 11 billion dollars, and are used by over 30 million subscribers. Their Videoguard product provides pay-per-view services as well as Video on Demand or Near Video on Demand services. The copyright of the content is protected by using fingerprinting: the content is watermarked with the customer's card identifier. Yet no collusion security seems to be in place. On the other side, payments are performed as usual, that is, in advance via credit card.

- Jump TV are an American company that provides Internet based international TV services. These TV channels are available worldwide to viewers with a simple Internet browser and Media Player. JumpTV are continuously adding new TV networks to their selection of channels. Once the customer has subscribed (by paying a certain amount per month, just a small fee per channel) she is able to view several channels by using her computer. Each of the channels offered on JumpTV is offered exclusively on a monthly subscriber basis, at costs ranging from 5.95 USD monthly for dial-up modem access to 9.95 USD monthly for high-speed broadband access. Payments are by credit card. All

subscriptions renew automatically on a monthly basis, and may be cancelled at any time.

- Some of biggest American entertainment companies, such as Universal [UnivPPV] and Fox [FoxPPV], provide pay-per-view services. For instance, in the case of Universal, the use of a set-top box allows the customer to download premiere movies from the Internet, which can be watched within 24 hours. Once downloaded, set-top boxes provide VCR functionalities, such as pausing the content. Once again, copyright protection mechanisms are unspecified and payments are performed as usual (beforehand via credit card).

Chapter 3

Unicast Real-Time Pay-per-View: the STREAMOBILE project

In this chapter unicast pay-as-you-watch of protected multimedia content is addressed. The whole chapter is based on research under the STREAMOBILE project, which this thesis is in part related to.

This chapter starts with an introduction to the STREAMOBILE project. Next, a prototype for the STREAMOBILE project, published in [Domi02a], is described. Our contribution [Mart03] is referred in Section 3.3. Finally, a design for a platform providing fine-grain pay-per-view and copyright protection, presented in [Mart03a], is sketched.

3.1 The STREAMOBILE project

STREAMOBILE (TIC2001-0633-C03-01, "Streaming of multimedia content to mobile devices with fee collection via micropayment") is a project supported by the Spanish Ministry of Science and Technology and the European FEDER fund. It begun in 2002 and finishes late in 2004. It is leaded by three research centers: Universitat Rovira i Virgili (URV), Universitat Oberta de Catalunya (UOC) and the Institut d'Investigació en Intel·ligència Artificial in Consejo Superior de Investigaciones Científicas (IIIA-CSIC).

The aim of STREAMOBILE is to demonstrate an Internet pay-per-view distribution system toward GPRS and UMTS mobile devices. Copyright-protected contents are streamed to the customer and contents are not paid for in advance, but as they are being received by the customer.

In the system proposed in STREAMOBILE, the customer pays as she is receiving the streamed content. The customer only pays for the part of the contents she actually watches and can quit at any time. Thus, the amount of individual payments is likely to be very small. Also, payments will take place very frequently: a payment will be made every minute or for every certain number of received frames.

It can be stated that the technology of STREAMOBILE can be also applied to fixed customers.

3.2 A prototype for STREAMOBILE

In this section, several aspects of the implementation of a prototype for STREAMOBILE are presented. This prototype consists of a client part and

a server part (see Figure 3.1).

The two main components of the client are:

- The web browser.
- A wallet application, to generate, store and deliver coins.

The server part (a desktop PC running under Linux) encapsulates the service provider and the content server functionalities. It consists of the following components:

- A web server that hosts the main page of the Internet pay-per-view video services.
- A content server that delivers content only if the client pays for it. It can be configured to request a payment before delivering n Kbytes, where n is a parameter.
- A running process (shop) which manages payments between the buyers and content servers. It acts as the micropayment broker.

3.2.1 Overview

PayWord coins([Rive95], see Section 2.3.1) are used for micropayments in the STREAMOBILE prototype. When the wallet program starts running, it sends a log-in message to the shop. This message includes the credit card number (so that the shop can pay off the customer's account whenever necessary).

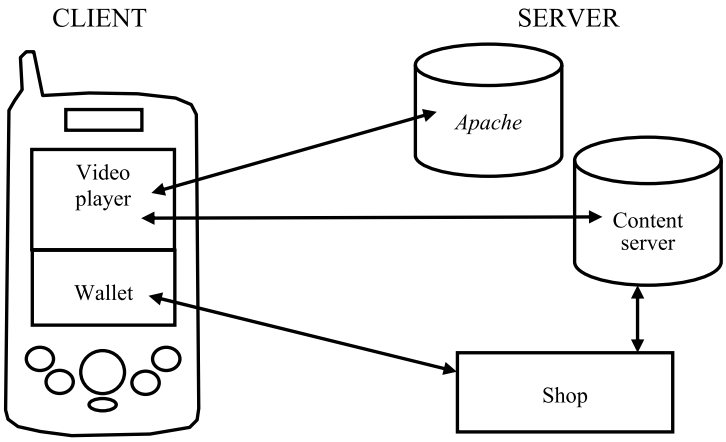

Figure 3.1: Components of the STREAMOBILE prototype.

The first time the customer has an empty wallet. In order to fill it with coins, the wallet sends a request to the shop telling it that the customer wants to create some coins. Once coin minting is allowed, the wallet program generates the coins. Then the wallet sends the last generated coin to the shop.

When the customer clicks on the link corresponding to a piece of content, say a movie, content streaming begins. The website hosted in the server points to the content hosted in the content server. Then the protocol below is followed:

Protocol 1 (Pay per stream)

1. *The content server opens a socket connection to answer the HTTP request. Thanks to this socket, the server knows the client's IP address.*

2. *The content server asks the shop for a client coin.*

3. *The shop sends a payment request to the client located at the IP specified by the content server.*

4. *The wallet receives a payment request. If there are coins left in the wallet, the wallet sends the next coin to the shop.*

5. *The shop checks that the received coin is an authentic one. If payment succeeds, a pair of acknowledgment messages are sent: one to the wallet (which deletes the last used coin) and another to the content server (which sends the first n Kbytes of the movie to the client).*

6. *The payment request process is repeated until the movie has been entirely sent or until the client drops the connection.*

3.2.2 Prototype implementation

In this section, the development and testing of the prototype is described. A picture of the environment in which the prototype has been tested can be seen in Figure 3.2. The shop and the pay-as-you-watch video server are implemented using Java language. The client part of the prototype has been tested on a PC and a HP-Compaq iPAQ (see Figure 3.2) PDA using the Java language. The portability of Java allows the client software to be run on any Java-enabled device (PC, PDA or mobile telephone). On the PC, the client application has been developed using the Sun Java Development Kit. The PDA allows a fluent streaming of MPEG or Windows Media Video. The latter is playable with the sole iPAQ Microsoft Pocket PC software, so no 3rd party components are needed.

With the PDA, two different developments were done:

Figure 3.2: The STREAMOBILE prototype on an iPAQ.

- Using the Jeode Runtime virtual machine by Insignia solutions, Inc., following the Personal Java specifications.
- Using Superwaba SDK which has got his own API.

Payword generation, using the SHA-1 [Nist93] hash algorithm and Base64 encoding, is faster using SuperWaba than using Jeode, by a factor of 20. Thus, it seems that the SuperWaba is more efficient for these treatments.

However the streaming obtained using SuperWaba is not fluent. An irregular video playing is obtained, whereas the quality of the sound is enough. It must be said that the Pocket PC becomes very slow when using concurrently the video streaming and the wallet application.

As a conclusion, sending micropayments over a WLAN link is suitable, as well as sending it through GPRS or UMTS. However, fast generation of

PayWord chains depends on the computing power of the hardware and the virtual machine used on the mobile device.

GPRS simulation

In order to try the STREAMOBILE prototype under GPRS or UMTS environments, a wireless link simulator has been developed [Camp03]. This software runs on a Linux PC and is able to apply a certain packet loss and packet delay model on a IP flow. This simulator was used in order to demonstrate that sending coupons over GPRS or UMTS is possible.

Results show that upload bandwidths for simulated GPRS and UMTS provide fast and reliable sending of PayWord coupons.

Results for UMTS communications

In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype

In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype

In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype In order to try the STREAMOBILE prototype

Table 3.1: UMTS times

	Number of variables			
Number of records	5	10	25	50
1,000	0.00	0.00	0.05	0.31
10,000	0.05	0.19	1.26	5.31
100,000	0.49	1.93	12.41	51.15

3.3 Secure fingerprinting for the customer

In this section, a scheme to obtain an implementable fingerprinting module presented in [Mart03] is described. Such scheme provides anonymity and asymmetry by means of a TTP. The concept of asymmetry is next introduced.

In fingerprinting schemes, the identity of honest customers should be kept secret unless they act dishonestly. This means a particular customer's *anonymity* will only be lost if a copy purchased by that customer is found to have been illegally redistributed. In *symmetric fingerprinting*, the mark is embedded into the content by the merchant who later sells the marked copy to the customer. So, both the merchant and the customer know the marked copy. The main problem of such schemes is that a dishonest merchant can redistribute himself a copy recently sold and accuse the customer of illegal redistribution. However, this argument can be used by a dishonest customer who can claim it was the merchant who redistributed her copy.

To prevent this situation, only the customer must know her marked copy. Schemes offering this property are called *asymmetric schemes* [Pfit96, Domi98]. In asymmetric schemes, the mark embedding procedure is replaced by a protocol in which both the merchant and the customer play an active

role. As a result of this protocol, the customer gets a marked object to which no one else, including the merchant, has had access.

Current asymmetric proposals in the literature are based on complex cryptographic protocols, such as secure multi-party computation [Pfit96] and zero-knowledge proofs [Domi98], whose implementation is impractical, if not infeasible.

3.3.1 FingerTrust: A public fingerprinting infrastructure

Obtaining security in a completely untrusted environment is a difficult task. It is common for current e-commerce solutions to rely on *trusted third parties*, *e.g.* public-key infrastructures. For instance, [Augo98] describes a watermarking system based on a TTP.

This fingerprinting system was implemented [Mart03c] using robust watermarking for images [Sebe01] and 2-secure fingerprinting codes [Domi00] (see Figure 3.3.

System entities

The system can be described by means of a scenario with the following entities:

Merchant or Content provider. This entity sells copyright-protected multimedia content.

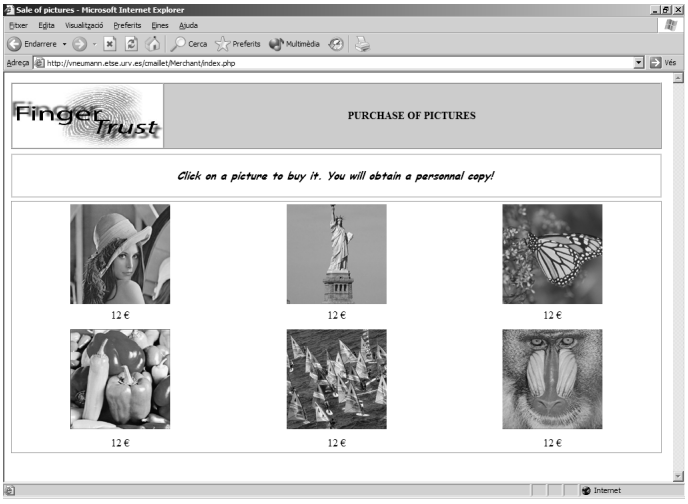

Figure 3.3: FingerTrust image list.

Customer. This entity purchases multimedia content. She is a completely untrusted entity, who may redistribute the contents she has bought.

Public fingerprinting authority. This trusted party carries out the mark embedding process.

Registration authority. This trusted party provides pseudonyms to customers that will be used for anonymous purchasing.

These entities are depicted in Figure 3.4. To summarize, the customer uses a pseudonym for each purchase. This pseudonym is obtained from the registration authority (1). In (2) the customer requests a product. The merchant asks the authority to mark the product (3). After marking, the fingerprinting authority delivers the marked product to the customer (4). Note that the merchant does not see the marked copy.

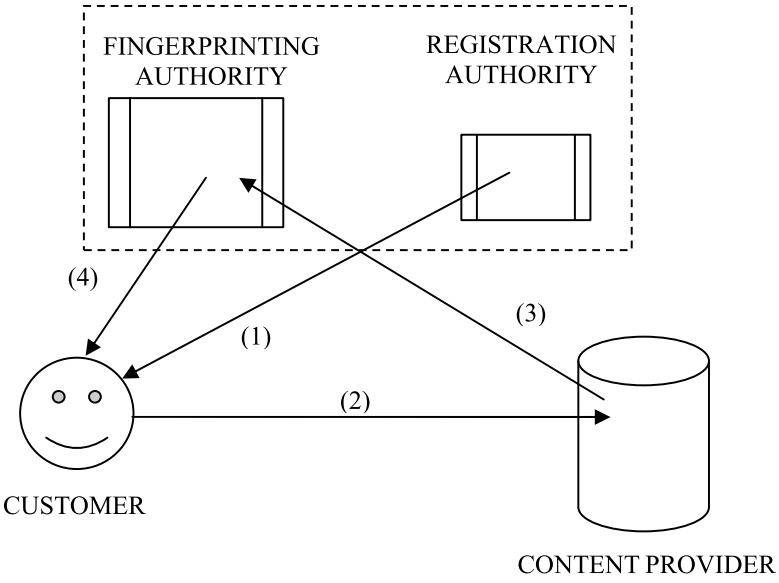

Figure 3.4: Marking process with a TTP.

3.3.2 Protocol suite

The interaction between parties is described next.

Merchant registration. The merchant registration protocol results in an agreement between the merchant and the fingerprinting authority. This agreement allows a merchant M to use her fingerprinting authority (FA) to mark multimedia objects (see Protocol 2) and identify the owner of an illegally distributed copy (as explained in Protocol 3). A secure channel is to be established between both parties (for example, by using authentication, encryption and digital signatures).

Customer registration. The registration authority (RA) provides a pseudonym to the customer. This protocol requires the customer to have a certified

public key, which is used to authenticate her identity when requesting pseudonyms. This request consists of a signed message indicating that a pseudonym is requested. RA generates the pseudonym randomly and stores it in its database, along with the identity of the customer. The pseudonym is sent to the customer encrypted with her public key. In this way, only the customer can decipher it.

The customer's pseudonym, ps_C, is composed of a serial number and its signature. In this way the pseudonym cannot be altered by the customer.

Content purchasing. The mark embedding protocol starts when a customer decides to make a purchase:

Protocol 2 (Mark embedding)

1. *Customer C composes a message msg containing: her pseudonym (ps_C), $Desc_X$ (a description of the product X) and her public key PK_C. msg is encrypted using the public key of FA.*
2. *The customer C contacts the merchant, by sending a signed message that contains: $E_{PK_{PFA}}(msg)$, $Desc_X$ and the date and time of the purchase.*
3. *Customer C performs an anonymous electronic payment. The e-cash schemes [Brand93, Chaum82] may be used for this purpose. Nevertheless, in a real deployment, a credit card payment gateway providing anonymity is likely to be used.*
4. *The merchant checks the validity of the signed message received in Step 2.*

5. *The merchant sends $E_{PK_{PFA}}(msg)$ and the product to be marked, X, to the fingerprinting authority.*
6. *FA decrypts msg, gets the pseudonym ps_C and checks its correctness by verifying that the signature is correct. It also checks that $Desc_X$ corresponds to X.*
7. *FA chooses a codeword w from a collusion-secure code:*
   (a) *If X has never been marked, FA constructs a collusion-secure fingerprinting code Γ_X with the appropriate parameters, and assigns the first codeword.*
   (b) *Otherwise, X is assigned the next unused codeword of Γ_X.*
8. *FA embeds w into X using a robust watermarking system.*
9. *FA stores ps_C and w together in its database.*
10. *The marked copy $\bar{X}$ is encrypted and sent to the customer, by using the* envelope encryption *technique: data is encrypted with a symmetric cryptosystem fed with a random key. The random key is sent to the customer encrypted with her public key. Note that for large files, a downloading link can be sent to the customer.*

Identification of dishonest customers. This last protocol is used in the case where the merchant finds an illegally redistributed copy of the content. For example, $\bar{X}$ can be found by the merchant being shared in a P2P environment such as Kazaa or eDonkey. Note that an altered copy of $\bar{X}$, $\hat{X}$, resulting from a collusion attack can also be found on the Internet.

This protocol identifies either the owner of $\bar{X}$, or some of the colluders that contributed to create $\hat{X}$.

Protocol 3 (Identity recovery) *Assuming that the merchant has found $\bar{X}$ or $\hat{X}$, the merchant proceeds as follows:*

1. *The merchant sends the illegal copy to FA.*
2. *FA recovers the embedded mark m from the illegal copy using the corresponding mark recovery algorithm of the watermarking system.*
3. *If a mark has been found, m is decoded so that a guilty codeword is obtained. The mark is decoded as it may have been altered as a result of a collusion attack.*
4. *The pseudonym related to the guilty codeword is sent to RA in order to disclose the identity of guilty customer.*
5. *Guilty customer may be prosecuted by the merchant.*

The TTP being used here as fingerprinting authority must gather a consensus and general acceptance similar to those required by a certification authority (CA). Indeed, the CA must be such that no court accepts doubts on the CA's honesty as valid arguments for signature repudiation; similarly, the FA must be such that no one can expect to cast doubts on the FA's honesty to deny having redistribution (or worse, to accuse the FA from redistribution).

A dishonest merchant cannot ask the FA to mark objects using the customer's pseudonym in products other than the one specified by the customer in her request (the pseudonym is encrypted along with other data concerning the purchase).

If the identity of a dishonest customer is recovered by the fingerprinting authority as a result of the identity recovery protocol, several measures can

be taken, the most obvious of which is to blacklist the customer so that no further copies of any product are sold to her.

3.3.3 Adapting FingerTrust for STREAMOBILE

In proposal described in Section 3.3 the content marked is sent by the fingerprinting authority to the customer. If FingerTrust is to be used directly as a module for STREAMOBILE, the next issues must be taken into account:

- The content (the video) should be sent to the FA.
- As the customer pays the content provider, the FA should mark the content and deliver it to the customer.

With this approach, it can be seen that the FA carries out not only with marking, but also with sending the video to the customer in real-time.

It can be seen that providing asymmetric fingerprinting by means of a TTP is likely to be unfeasible due to the high communication efforts between the content provider, the fingerprinting authority and the customer.

A more realistic approach is to mark the content using the player [Bao00]. To summarize, as the content is being played, the mark to be embedded is sent to the customer from the fingerprinting authority.

3.4 A proposal for a pay-as-you-watch service of protected content

In [Mart03a], a Multi-device system for an INternet-based PAY-as-you-watch (MINPAY system) is presented. The system, which is part of the STREAMOBILE project, allows on-demand streaming of videos with three main features: the customer pays only while the contents are being downloaded; copyright protection through identification of illegal copies; device-specific bitrate coding of the streamed video. It is intended that commercial platforms such as CinemaNow, MovieLink and JumpTV (see Section 2.7) should follow the specifications below.

The MINPAY system handles two kinds of entities: customers (the users of the system) and contents. Banks will also be contacted in a real deployment of the system.

3.4.1 System architecture

There are several components involved in the MINPAY system. At the front-end and back-end there are the customers and the contents, whereas other parties act as intermediate systems. These elements and their interaction are explained next and depicted in Figure 3.5:

- The *web portal* is the entrance to the web service. When entering the site, a customer check is performed (reading a cookie or some data from the smart card is attempted on the customer's side). If the customer is already registered, the main page redirects her to a device-oriented web (in order to show the portal adapted to the display of the device).

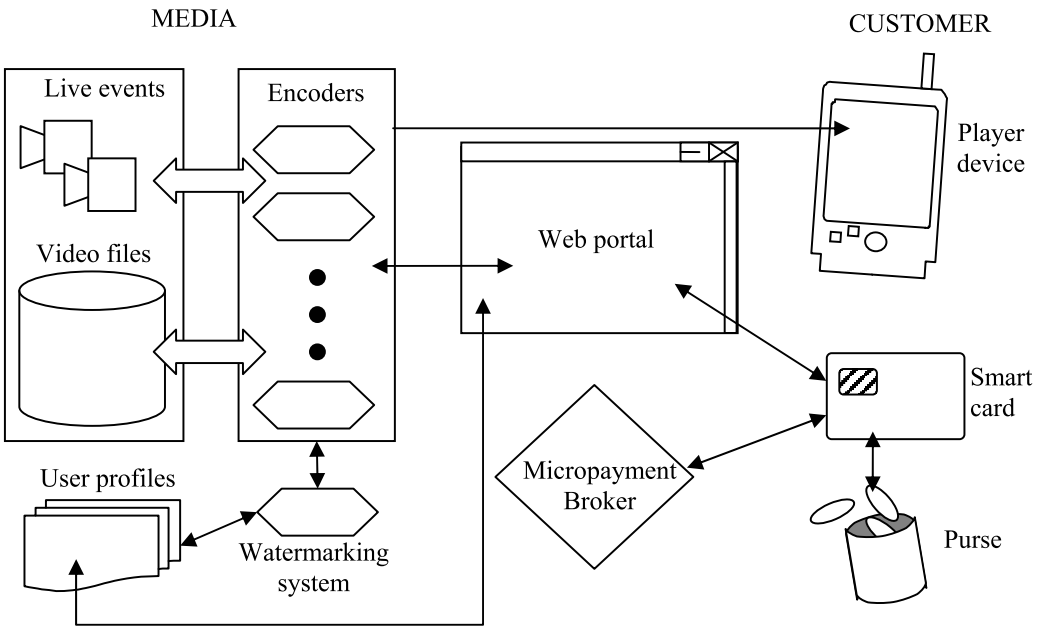

Figure 3.5: Components of the MINPAY system.

If the customer is not registered, the browser sends a simple page with information about the services and allows the customer to join them. Once registered, a customer profile is stored in the database.

- The *player device* allows the customer to watch the streamed content. A smart card should be plugged into the customer's device to cryptographically assure the customer's identity and perform micropayments instead of using the purse software.

- *Content streams* can be obtained either from video data files or can be directly grabbed from video capturers (*i.e.* for live events). Video files are stored at highest quality. Several resolutions and compression ratios are possible, but a resolution of 320x240 pixels should be enough

for high-quality videos.

Intermediate systems

In addition to the web server that will serve HTML content to the customer, there are several other intermediate systems between the customer and the content:

- The *video encoders* take content streams as inputs and recode them in order to match the bit rate that best fits the needs of the customer. Either hardware or software encoders may be used, but the cost of having an encoder for each active customer stream has to be analyzed. Hardware encoders certainly increase the price of the system, whereas software encoders seem a better solution when there are encoding processors running on several computers. On the other hand, the system may have several videos encoded at different bitrates and simply stream the data to the customer. Anyway, the main purpose of the system must be to achieve a bitrate as close as possible to the customer's bandwidth, in order to avoid the buffer problem (see Section 3.4.2 below).

- Each video is watermarked by the *watermarking system* while being streamed. Embedding algorithms may differ depending on the content handled, but the basic idea is to embed the message where it can barely be perceived by humans. The use of collision-secure fingerprinting prevents copyright marks from being erased from the source.

- The *micropayment broker* interacts with the customer in order to obtain and validate micropayment coins. These microcoins are aggregated and

sent to the banks in order to settle the accounts of content providers and customers (for instance, once per month).

When a video streaming is requested, a customer-dedicated video encoder interacts with the watermarking system and they both contribute to generate a stream with customer-specific bitrate and watermarks.

The same customer may, on different occasions, be using devices which can cope with different bandwidths. The system allows a customer profile to contain several device specifications. When a customer logs into the system, if her profile contains several device specifications, she is asked about which device she is currently using.

Concurrently, the system asks the micropayment broker to request several micropayments to the customer in order to get paid for the service. The server process sends the stream to the customer's player at a specified bitrate, as explained in next section.

3.4.2 The buffer drawback

The client player uses a buffer in order to compensate packet delay or packet loss so that the video stream can be drained out of the buffer at a specified bit rate.

Thus, some initial payments will probably be needed in order to fill the buffer with enough data to start viewing some movie frames; at this moment, the customer has paid but she has seen nothing of the movie. However, the difference between the amount of content paid for and the amount viewed is usually small as compared to the size of the entire content.

Depending on the transport protocol used, the size of the buffer and the bandwidth of the incoming Internet connection, the customer's player buffer is likely to be swamped with data. The explanation is that the network tends to bring to the receiver as much data as it possibly can. Note that the customer would pay for all data entering the player's buffer, even if she quits watching before viewing the entire buffer content.

Due to this drawback, coding must be done so as to produce an output stream that ensures that the customer's buffer as empty as possible. One solution is to measure the maximum sustained bandwidth from the server to the customer [Ison03], and encode the video at approximately the same bitrate. In the case of wireless connections, where bandwidth fluctuation is higher than for fixed IP networks, that would imply offering poor qualities to these customers. For that kind of customers, a periodic monitoring should be used in order to encode at a variable bandwidth [Mont01, Pack].

In conclusion, overcoming the so-called buffer drawback would satisfy the customers of a MINPAY-based service.

Chapter 4

Multicast Real-Time Pay-per-View

In the near future, most multimedia delivery services are likely to operate in multicast mode to send content over the Internet. When applying concepts as fee collection for pay-as-you-watch or copy detection by fingerprinting in a multicast environment, the next issues must be taken into account:

- Given that a large audience is possible, collecting pay-as-you-watch payments from all customers may overwhelm the source or payment collector. This bottleneck, known as the *implosion problem* [Quin01], may arise in any communication from N parties to one party, *i.e.* many-to-one communications.

- On the other hand, since in multicast the same content is sent over to many different customers it seems unnatural that every copy can be marked for each customer.

Current research and standards drafting on multicast security basically focus on multicast groups, group membership and rekeying. No real effort is devoted to issues such as traitor tracing of multicast content (multicast fingerprinting) or non-implosive reverse many-to-one secure payment. Next sections, try to provide solutions to the aforementioned drawbacks.

In Section4.1, our contribution [Domi04] for secure aggregation of information, from the leaves to the source, in a multicast tree is presented. It can provide non-implosive and non-verifiable subscription pay-as-you-watch (see Section 4.2.1) as well as a secure protocol for large-scale bingo game, the latter described in Section 4.2.2. In Section 4.3 our proposal [Mart04] for aggregatable and verifiable payments is presented. The last part of this chapter deals with fingerprinting schemes for multicast delivery, presenting our contribution [Mart03b].

4.1 Secure aggregation of information in many-to-one communications

Besides real-time collection of payments for a multicast audience, several applications require a large group of senders to transmit some real-time information to a single receiver. A network of sensors sending status information to a control center is one example of such applications. Other examples include network monitoring, resource discovery in networks, acknowledgment of messages in reliable multicast protocols, etc.

In addition to requiring solutions to the aforementioned implosion problem, some many-to-one applications (like real-time pay-per-view of multicast

content) require secure and real-time transmission. Security usually means that transmission from each sender to the receiver should be confidential and authentic.

The best way to avoid the implosion problem is that intermediate nodes aggregate the information sent from the large set of senders to the unique receiver. A few contributions about aggregation of data streams in many-to-one communications can be found in the literature. In [Wolf03], a technique called aggregated hierarchical multicast is presented, whereby packets are aggregated in multicast nodes. This work is based, even similar, to the Concast system presented in [Calv01]. Concast and aggregated hierarchical multicast are introduced as aggregation mechanisms that basically suppress duplicate packets. It must be noticed that large data packets can be output from inner nodes, depending on the information sent and the number of senders attached to a node. However, the network layer seems to be the natural place to carry out the aggregation of information. According to this, as stated in [Wolf03], the aggregation operation of data packets inside the network requires the support of the network infrastructure in terms of processing resources. The scheme described in our paper also requires the support of an *active network* [Psou99].

Active networks allow information to be handled in the core nodes of the network. Not only workstations can be acting as *active nodes* and perform data operations: state-of-the-art routers and switches may be able to run some algorithm on data packets they receive in order to deliver a new and processed packet to their parent node in the routing/switching tree. Mobile *ad-hoc* networks are an increasingly popular example of active networks.

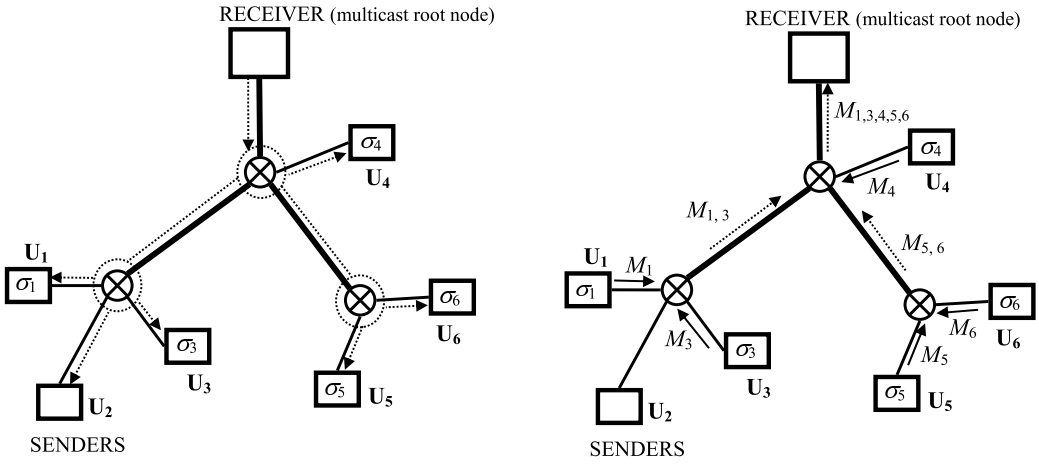

Figure 4.1: The implosion problem solved using secure aggregation of data.

4.1.1 Overview

The scheme consists of a set-up protocol to be run before any transmissions are started, and a transmission protocol to be run for each symbol transmission. From now on, the *senders* are the leaves in the routing tree, whereas the final *receiver* is the root of the tree. The main goals are:

- A dramatical reduction of the number of connections to the receiver, in order to avoid the implosion problem.
- Secure communication between the senders and the receiver. Security consists of communication confidentiality and sender authentication.
- Generation of fixed-length aggregated packets. Previous proposals did little more than merging packets.

The operation of the protocol can be summarized as shown in Figure 4.1.1. On the left, there are several senders U_i want to transmit a symbol σ_i; first,

a challenge message is multicast (dotted line). On the right, senders send encoded messages M_i, whereas core nodes (routers) aggregate the information and send it to their parent node (dotted line).

In order to receive the symbols from the senders U_i, a challenge message is multicast by the receiver to all senders, via the routing tree, in order to synchronize the transmission. Routers in the tree aggregate encoded messages M_i received from their child nodes/senders and send aggregated information up to their parent nodes. The last aggregation procedure, performed at the active node closest to the receiver, produces a final message containing all symbols σ_i transmitted by the senders. The receiver is then able to decode the aggregated symbols from the final message. Since our scheme is designed for real-time traffic, it is assumed that information cannot be buffered but should be sent symbol by symbol as these are generated by the senders. In practice, a mapping between the application-level language and the symbol-level language is likely to be used, whereby sending a single word in the application-level language may require sending two or more symbols.

Our proposal is based on super-increasing sequences [Merk78] and probabilistic additive public-key privacy homomorphisms (PH, [Okam98]). These two concepts are briefly explained next:

Super-increasing sequences. Given a sequence of positive integers $\mathcal{S} = \{S_1, S_2, \ldots, S_{m-1}, S_m\}$ and a value T which is the sum of some elements in $\mathcal{S}$, the *knapsack problem* consists of finding a subset $\mathcal{S}' = \{S_a, S_b, \ldots, S_j\}$, of $\mathcal{S}$ whose sum equals T. The general knapsack problem is known to be an NP-complete problem, but there are some cases in which the problem can be solved polynomially. This is the case when the sequence $\mathcal{S}$ is *super-increasing*. The scheme proposed in this

paper uses super-increasing sequences to aggregate integer values in a reversible way. The knapsack problem is used for symbol extraction from the aggregated message.

Privacy homomorphisms (PHs). PHs are encryption transformations mapping a set of operations on cleartext to another set of operations on ciphertext. A PH is called *additive* when its set of cleartext operations contains addition. A PH is called *probabilistic* if the encryption algorithm involves some random mechanism that chooses the ciphertext corresponding to a given cleartext from a set of possible ciphertexts. Privacy homomorphisms that will be used in our proposal below must be additive, probabilistic and public-key. The Okamoto-Uchiyama [Okam98] probabilistic public-key cryptosystem (OUPH) has an additive homomorphic property. This probabilistic public-key cryptosystem is proven to be as secure against passive adversaries as the intractability of factoring $n = p^2p'$, where p and p' are two large primes.

4.1.2 Construction

We first describe a set-up protocol whereby the receiver chooses the parameters that ensure that symbols can be aggregated and later be extracted.

Protocol 4 (Set-up)

1. *The receiver chooses parameters l, u, where l will be used below and u is the number of senders. Let t be the bit length of each symbol to be transmitted.*

2. *The receiver computes tu intervals as follows:*

$$\mathbf{I}_j = [I_j^{min}, I_j^{max}] = [(2^j - 2)2^l - 2^{j-1} + 2, (2^j - 1)2^l - 2^{j-1} + 1]$$

for $j = 1$ to tu. Each sender is assigned t intervals among the above; specifically $\mathbf{I}_{(i-1)t+1}$ to $\mathbf{I}_{it}$ correspond to the i-th sender.

3. *The receiver generates a secret value k_i[1] for each sender i, for $i = 1$ to u.*

4. *The receiver generates a key pair for a probabilistic additive public-key privacy homomorphism such that its cleartext space is $CT = \{0, 1, 2, \cdots, p-1\}$ where p should be larger than $2I_{tu}^{max}$. After some manipulation, it can be checked that the lower bound on p is*

$$p > (2^{tu} - 1)2^{l+1} - 2^{tu} + 2 \tag{4.1}$$

5. *The receiver multicasts the public key PK of the PH and I_j^{min} for j=1 to tu. In addition, the receiver secretly sends k_i to each sender U_i, who should keep it confidential (storing it in a tamper-resistant device such as a smart card would seem appropriate).*

After set-up, the normal operation of the scheme will consist of several real-time symbol transmissions. In order to collect symbols from each sender, the following four-step protocol is used:

Protocol 5 (Real-time symbol transmission)

[1]Large enough for exhaustive search infeasibility.

1. Transmission request. *A challenge message is multicast by the receiver to all senders. This challenge contains a random value v.*

2. Message generation.

   (a) *When a sender U_i receives the challenge message, she computes her own t values:*

$$S_{ti-t+j} = I^{min}_{ti-t+j} + \mathcal{H}(v + j - 1||k_i) \tag{4.2}$$

   for j=1 to t where $\mathcal{H}$ is a one-way collision-free hash function yielding an l-bit integer as output. This condition on the output of $\mathcal{H}$ ensures that $S_{ti-t+j} \in \mathbf{I}_{ti-t+j}$, which in turn guarantees that the entire sequence $\mathcal{S} = \{S_j\}$ *for $j = 1, \cdots, tu$ is* super-increasing. *Note that, since v and the parameters in Protocol 4 were chosen by the receiver, the latter can readily compute the subset of $\mathcal{S}$ corresponding to any sender. On the other hand, Condition (4.1) ensures that no overflow in CT will occur when adding encrypted terms of the super-increasing sequence over the ciphertext space CT'. Now, U_i can transmit $2^t - 1$ different symbols by sending the encrypted sum of a subset chosen among the $2^t - 1$ non-empty subsets of $\{S_{ti-t+1}, \cdots, S_{ti}\}$. For instance, if the symbol σ is mapped to the sum of values S_{ti-t+1} and S_{ti-t+3}, sender U_i computes the following message:*

$$M_i = E_{PK}(S_{ti-t+1} + S_{ti-t+3})$$

   where

$$E_{PK}$$

*stands for the encryption function of the probabilistic additive public-key privacy homomorphism used. Since a probabilistic cryptosystem is being used, the same cleartext message can result in different encrypted messages. Note that the encrypted sum of the empty subset (*i.e. $E_{PK}(0)$*) cannot be used to encode a value in a secure transmission because anyone can send it (no authentication) or guess it (no confidentiality).*

(b) Finally U_i sends M_i up to her parent node. The size of M_i is discussed in Section 4.1.3.

3. Message aggregation. *Intermediate nodes receive messages from their child nodes/senders and do the following:*

(a) Once all expected messages $\{M_i\}_i$ have been received, the node aggregates them as $M = \sum'_i M_i$, where $\sum'$ stands for the ciphertext operation of the privacy homomorphism corresponding to cleartext addition.

(b) The node sends M up to its parent node. The size of M is discussed in Section 4.1.3.

4. Symbol extraction. *When the previous process completes, the receiver finally receives an aggregated message M, from which the transmitted symbols are extracted as follows:*

(a) The receiver constructs the entire super-increasing sequence $\mathcal{S} = \{S_j\}$ for $j = 1$ to tu using, for each sender U_i, Equation (4.2).

(b) The receiver decrypts M using its private key of the PH to recover a value T which is used to solve the super-increasing knapsack problem and obtain the sequence $\mathcal{S}' = \{S_1, S_2, \ldots\}$ that yields the

values sent by the senders. From these values, the symbol σ_i sent by every sender U_i is easily retrieved.

4.1.3 Security

A basic assumption when analyzing security is correctness in protocol execution, *i.e.* that Protocol 5 is followed by all senders without deviations. If one or more senders deviate, symbol extraction at the reception might fail; for example, this would be the case if a sender used intervals corresponding other senders, which might result in a sequence $\mathcal{S}$ that is not super-increasing. Note that correctness in protocol execution can be enforced if senders are forced to using a computing device trusted by the receiver (*e.g.* a smart card). The receiver can use Protocol 4 to force senders, by refusing to give the secret keys k_i to anyone except sender smart cards issued or trusted by the receiver.

Assuming that Protocol 5 is correctly followed by all senders, we next state the security properties of our scheme.

Property 1 (Confidentiality) *If a secure probabilistic additive public-key PH is used in which there is a negligible probability of obtaining the same ciphertext as a result of two independent encryptions of the same cleartext, then an intruder cannot determine the symbol transmitted by a sender in Protocol 5.*

Proof: Without loss of generality and to keep the proof simple, we can restrict ourselves to the case $t = 2$, or binary symbol transmission. Now, assume that the intruder captures a message M sent by U_i during Protocol 5. This message is either $E_{PK}(S_{2i-1})$, $E_{PK}(S_{2i})$ or $E_{PK}(S_{2i-1}+S_{2i})$. Decryption

of M is not possible because the PH is secure and the intruder does not have access to the private key. Exhaustive search of the cleartext carried out by M is the other attack strategy to be examined. Now, exhaustive search of the sequence values S_{2i-1} or S_{2i} by encrypting candidate values and comparing the result to M will fail with overhelming probability by the assumption on PH. □

Property 2 (Authentication) *If a secure public-key PH and a one-way collision-free hash function with l-bit output are used, the following holds:*

1. *the probability of successfully impersonating another sender when sending a bit value to the receiver is 2^{-l};*
2. *substituting a false message M' for a legitimate message $M \neq M'$ in the current transmission is at least as difficult as impersonation;*
3. *substituting a message M' for a legitimate message $M \neq M'$ in future transmissions using information from the current transmission is infeasible.*

Proof: As in the previous proof, we can restrict the proof to $t = 2$. In the impersonation attack, an intruder who wants to impersonate sender U_i tries to generate a message $E_{PK}(S_{2i-1})$, $E_{PK}(S_{2i})$ or $E_{PK}(S_{2i-1} + S_{2i})$, Now, the intruder needs to compute S_{2i-1} or S_{2i}. Each term S_j of the super-increasing sequence $\mathcal{S}$ is pseudo-randomly chosen within an interval $\mathbf{I}_j$ containing 2^l integer values. The choice is made using a one-way collision-free hash function of the challenge and the secret key k_i unknown to the intruder, as shown in Equation (4.2). Thus, the probability of the intruder randomly hitting S_j is at most 2^{-l}. Remark that exhaustive search is not feasible, since there is no

way of checking whether the right S_j has been hit (there is no way for the intruder to make sure whether the message generated with the candidate S_j is correct). A substitution attack can be mounted in the current transmission or in future transmissions:

- In the current transmission, assume the intruder wants to substitute a false message M' for an authentic message M sent by U_i, with $M' \neq M$. Without loss of generality, let $M = E_{PK}(S_{2i})$; the intruder wants to transform M into $M' = E_{PK}(S_{2i-1})$ or $M' = E_{PK}(S_{2i-1} + S_{2i})$. This requires the following steps: i) recover S_{2i} from M; ii) compute S_{2i-1} with knowledge of S_{2i}; iii) compute M'. Thus, even if decrypting M at step i) was easy (which it is not), solving step ii) is as difficult as mounting a successful impersonation attack (see above).

- A second possibility is for an internal intruder to use information derived from a current transmission of a message by U_i to alter future messages sent by U_i. But this is infeasible, because in subsequent executions of Protocol 5, a different super-increasing sequence will be used to encode the messages which does not depend on the current super-increasing sequence (see Equation (4.2)).

□

Performance

Before presenting the performance comparison below, some preliminary remarks are required:

- The performance criterion considered is the bandwidth required by the aggregated traffic.
- In order to benchmark the performance of our system, we will consider an alternative system based on unicast transmissions from each sender to the receiver. Like in our system, the unicast transmissions in the benchmark system will be symbol-wise. We assume that the communication is real-time, so that symbols are transmitted as they are generated, rather than being buffered and transmitted in batches.
- We will require that each symbol transmission in the alternative unicast system has the same security properties as transmissions in our system.
- For the sake of concreteness, we will use OUPH as a privacy homomorphism in this section.

A benchmark unicast system. In order to avoid the need for public-key encryption for a sender to send a confidential and authenticated symbol, we must assume that each sender U_i shares with the receiver a key k_i corresponding to a block cipher (*e.g.* AES). The message M containing the symbol σ will thus look like

$$M = E_{k_i}(\sigma||ts||ck), U_i$$

where $E_{k_i}(\cdot)$ stands for the encryption function of the block cipher, ts is a time-stamp, ck is a checksum and U_i is the identity of sender U_i. Integrity is ensured by ck and ts (the time-stamp prevents replacing future transmissions with past transmissions).

Comparison. When u senders simultaneously send their encrypted symbols with the benchmark unicast system, $u(B + \log_2 u)$ bits are received by

the receiver, assuming that B is the block bitlength of the block cipher and $\log_2 u$ is the bitlength of the sender identifier U_i. We assume also that the bitlength of $\sigma||ts||ck$ is less than or equal to B. For a block cipher such as AES, at least one has $B = 128$, so the previous assumption is reasonable. When u senders send their encrypted symbols with our system, all symbol transmissions are eventually aggregated into a single message

$$M = \prod_i M_i \pmod{n}$$

which is the only one reaching the receiver. M can be at most n, so its length is $\log_2 n$. Equivalently, the bitlength of M is

$$|M| = \log_2 n = \log_2(p^2 p') = 2\log_2 p + \log_2 p' = 3\log_2 p$$

where we have used that, in OUPH, $n = p^2 p'$ with $|p| = |p'|$. Now, already for a moderate number u of senders, p can be chosen close to its lower bound (4.1) while remaining large enough for factoring of $n = p^2 p'$ to stay hard, as required by OUPH. Therefore, if we use the generalized bound (4.1) we have

$$|M| \approx 3\log_2[(2^{tu} - 1)2^{l+1} - 2^{tu} + 2] \tag{4.3}$$

It can be seen that Expression (4.3) is dominated by $3tu$ as the number of senders grows. Therefore, if the number u of senders is moderate to large and if the symbol bitlength is $t < (B + \log_2 u)/3$, the bandwidth $3tu$ required by our scheme is *less* than the bandwidth $u(B + \log_2 u)$ required by the benchmark unicast system. Since typical block sizes are as large as $B = 64, 128, 192$ or 256, the previous assumption on the symbol bitlength is reasonable. Besides, our proposal only requires

one incoming connection to the receiver, whereas the unicast alternative requires u connections to the receiver, which calls for allocation of additional overhead bandwidth not included in the above comparison. Finally, it must be noticed that bandwidth reduction is achieved without increasing the computational burden at the receiver. Symbol extraction during Protocol 5 requires the receiver to build tu terms of a super-increasing sequence and to solve a super-increasing knapsack problem. The computational cost of doing this is similar to the cost of the u block decryptions required by the unicast benchmark.

4.2 Some utilities for aggregation of information

In this section two utilities for the aforementioned secure aggregation of information are presented.

The first one is directly related to a pay-as-you-watch application, since the aggregation of information protocol is used for a subscription pay-as-you-watch scheme. Finally, our contribution [Mart04] for a secure large-scale video is briefly sketched.

4.2.1 Subscription pay-as-you-watch for multicast

Next it is described how the previous scheme can be used for a pay-as-you-watch for large multicast audiences. In this approach, the customer subscribes to the provider's pay-as-you-watch multicast service and, consequently, gets the public/secret parameters specified in Section 4.1.2.

However, the provider needs to know how long each customer has been receiving the content in order to bill her accordingly.

In encrypted multicast communications [MSEC03], the content is encrypted under a symmetric session key known only to the set of registered receivers. Thus, when a customer is interested in registering to a multicast session, she must request the decryption key. In this way, the source/provider knows exactly the moment at which the customer starts receiving the content. On the other side, the customer can disconnect without notifying the source. Hence, in a subscription-based service, customers must periodically confirm that they stay connected. This many-to-one confirmation communication must be private and authenticated and, as mentioned before, can lead to implosion problems at the source.

To remedy this, the protocol described in this chapter must be used, specifically, using the case for t=2 symbols. Each U_i generates her message as follows:

- If she wants to transmit a 0 bit value, she generates the message $M_i = E_{PK}(S_{2i-1})$ where $E_{PK}(\cdot)$ stands for the encryption function of the probabilistic additive public-key privacy homomorphism used.
- If she wants to transmit a 1 bit value, she generates $M_i = E_{PK}(S_{2i}))$.
- Ternary symbols could also be transmitted. Thus, a third symbol (other than 0 and 1) could be transmitted if U_i sent $M_i = E_{PK}(S_{2i-1} + S_{2i})$.

Hence, if a user/customer sends any of her secret values s_u^0 and s_u^1, it means that she is still online. This is useful for the source/provider to learn that the customer must be billed till at least the moment of receiving the last s_u^i. The symbol mapping in this case is described below.

4.2.2 A secure large-scale bingo protocol

In [Mart04] we presented a bingo protocol based on Protocols 4 and 5. Such protocols allow the deployment of a large-scale bingo, using multicast routers or a distributed network of trusted nodes, and provide security in terms of secrecy and authentication.

Conventional, *i.e.* physical, bingo is played in a large hall. Players meet at the hall and the game begins. In this way, many bingo games are played one after the other. A *bingo game* proceeds as follows:

1. There are 99 possible bingo numbers: $\mathcal{B} = \{1..99\}$. Each of these numbers is represented by a ball in a large rotating bin. Each ball is painted with its unique bingo number.

2. Every player receives a bingo card with 15 different numbers. These numbers are distributed in three rows, with five numbers each.

3. An announcer spins the bin and selects a ball (or a computer randomly selects a number). This number is announced to the audience.

4. Then each player checks her card to see whether the announced number appears on it.

5. This is repeated until a bingo or a line are called out. There are two winning configurations:

   - **Line.** If all the numbers in a row of a card have been selected, its owner calls out line. The game pauses while the card is verified. If a line has been completed, the player receives about 8 percent

of the total bet. Then, the game goes on, but nobody else can call out a line.

- **Bingo.** When all the numbers on a card have appeared, its owner must call out bingo. The game pauses while the card is verified. If a whole card has been completed, the game ends. Note that the game always ends, because someone will complete a card sooner or later.

The above description corresponds to European and, more specifically, Spanish bingo. In other versions of bingo (*e.g.* American), the winning configurations may vary, but the basic operation of the game stays the same.

A protocol for secure large-scale bingo

An electronic version of the above bingo rules is described next. Let u be the number of players, where $u \gg 1$.

Enrollment. The source announces an *enrollment period* in order to allow new players to join. When a player is interested in joining a virtual bingo hall, she registers to the source using a unicast communication. The parameters used in Protocols 4 and 5 are supplied by the source. This would require sending renewed parameters to players already in the system.

Game start. In conventional physical bingo, cards are generated and supplied by the source. In the electronic version, each player randomly creates her bingo card in the following way:

1. Every player U randomly chooses 15 different numbers from $\mathcal{B}$ in order to fill her bingo card $\mathcal{C}_U$. These numbers are sorted in ascending order and split in three rows of five numbers each.

$$\mathcal{L}_l^U = \{b_{l,1}, b_{l,2}, b_{l,3}, b_{l,4}, b_{l,5}\}$$

for l=1..3

$$\mathcal{C}_U = \{\mathcal{L}_1^U, \mathcal{L}_2^U, \mathcal{L}_3^U\}$$

2. The player computes a one-way hash of each line l, in the following way

$$\mathcal{H}_l^U = \mathcal{H}(b_{l,1}, b_{l,2}, b_{l,3}, b_{l,4}, b_{l,5}, salt_l)$$

where $salt_l$ is a random number, and sends the resulting hash to the source. In this way, the player commits to the selected values. These hash values are sent to the source one bit at a time using Protocol 5. This procedure avoids information implosion at the source. Moreover, no customer-dedicated secure connections are needed.

3. The use of Protocol 5 guarantees that the source knows the identity of the new bingo game players, as shown below.

4. Note that during this step, players may perform a micropayment that allows them to play. Otherwise, a subscription model may be used for payment.

Game operation. Once the game has started, bingo numbers are successively and randomly chosen by the source and checked by players, until the game ends:

1. A bingo number b is randomly chosen. This number cannot appear again during the current bingo game (due to bingo rules). b is multicast to every player.

2. Every player checks whether b is in her bingo card. If so, b is marked.

3. A transmission request is multicast by the source to all players. Player U_i generates a message M_i, according to the following message-to-symbol mapping:

   - S_{2i-1} = player has received the number, but has not completed a line or bingo.
   - S_{2i} = all five numbers of $\mathcal{L}_0^{U_i}$, $\mathcal{L}_1^{U_i}$ or $\mathcal{L}_2^{U_i}$ have appeared (line).
   - $S_{2i-1} + S_{2i}$ = all numbers of $\mathcal{C}_{U_i}$ have appeared (bingo).

   Messages are sent and aggregated following Steps 3a and 3b of Protocol 5.

4. The source finally receives an aggregated message M, from which the transmitted symbols are extracted.

   - If a user U_i has sent a line message, a unicast connection is set up between the source and U_i. The latter sends a message to the source containing the bingo numbers of the winning line, and the $salt_l$ value used for that line. The source then checks that the hash U_i committed to at the beginning of the game corresponds to the sent line. Once all messages from line-winning players have been checked, no more line messages will be accepted (due to bingo rules).
   - If a user U_i has sent a bingo message, U_i sends to the source

the three lines of $\mathcal{C}_{U_i}$ and the corresponding $salt_l$ values for checking.

4.3 Aggregatable payments for multicast

Note that in the previous subscription approach, the provider does not obtain any proof, *i.e.* she could not convince a third party, that the customer is correctly receiving the multicast content. So a scheme where keepalive bits are sent using Protocols 4 and 5 is non-verifiable.

As mentioned before, the main barrier to using traditional micropayment schemes for fee collection in multicast environments would be the implosion problem. Nevertheless, due to the increase in the computational power of processors and the advances in digital signatures techniques, it is no longer obvious that the computational cost of a digital signature is still unaffordable for micropayments [Mica02].

By using verifiable payment subscription, the source can prove that a certain customer has received a certain portion of content. On the other hand, a customer cannot deny having received the content (non-repudiation).

In [Domi02b] we proposed the aggregation of payments as a solution to payment implosion in large-scale pay-as-you-watch. This proposal, based on ... **Caldria posar-ho?**

Our proposal is secure and scalable. It is based on the concept of multisignature [Bold03].

4.3.1 Multisignatures

Multisignatures allow any subgroup of a group of entities to jointly sign a document in such a way that any verifier is convinced that each member of the subgroup participated in the signature.

Computational Diffie-Hellman problem (CDH). The CDH problem consists of finding $h = g^{\log_g u \cdot \log_g v}$ given three random elements $\{g, u, v\}$ of a group.

Decisional Diffie-Hellman problem (DDH). The DDH problem consists of deciding whether four elements $\{g, u, v, h\}$ in a group satisfy $\log_g u = \log_v h$.

Gap-Diffie-Hellman group. A Gap-Diffie-Hellman (GDH) group is one in which the CDH problem is hard but the DDH problem is easy.

In [Bold03], a multisignature scheme is proposed which can be built over any Gap-Diffie-Hellman (GDH) group. We next recall that scheme.

Protocol 6 (GDH multisignature)

1. Key generation. *Let G be a GDH group and g a generator of G. In order to generate her public-private key pair, a customer U chooses a random positive integer x_u (her private key) and publishes $y_u = g^{x_u}$ (her public key).*

2. Signature computation. *In order to sign a message m, a customer U computes her signature as $sig_u = \mathcal{H}(m)^{x_u}$, where $\mathcal{H}$ is a one-way hash function.*

3. Signature verification. *This signature is verified by solving the DDH problem over the elements*

$$\{g, y_u, \mathcal{H}(m), sig_u\}$$

If the answer to the DDH problem is yes, then the signature is accepted as valid.

4. Multisignature computation. *Given a customer U_1 with a public-private key pair (y_{u_1}, x_{u_1}) and a customer U_2 with a public-private key pair (y_{u_2}, x_{u_2}), a multisignature of U_1 and U_2 on a message m is computed as follows:*

(a) U_1 computes $sig_{u_1} = \mathcal{H}(m)^{u_1}$

(b) U_2 computes $sig_{u_2} = \mathcal{H}(m)^{u_2}$

(c) The multisignature on m is then computed as

$$sig_{u_1,u_2} = sig_{u_1} \cdot sig_{u_2}$$

5. Multisignature verification. *The multisignature can be verified by solving the DDH problem over the elements*

$$\{g, y_{u_1} \cdot y_{u_2}, \mathcal{H}(m), sig_{u_1,u_2}\}$$

If the answer to the DDH problem is yes, then the multisignature is accepted as valid.

The generalization of the above multisignature computation and verification to a set of n customers is straightforward.

4.3.2 Protocols

Next, we propose a scalable solution whereby the multicast source/provider can collect a proof that all customers registered in a multicast session have received a specific piece of content.

Protocol 7 (Aggregation using multisignatures)

1. Customer registration. *In our proposal, we require each customer U_i to have a public-private key pair (y_{u_i}, x_{u_i}). The public key must be certified by a trusted certification authority.*
2. Payment request.

   (a) *The source generates a message m specifying the content, the time slot and the amount of money to be paid. This message m is multicast to the set of registered customers who are currently receiving the content.*

   (b) *Upon reception, customers check the content of m for correctness and sign it. That is, customer U_i computes $sig_{u_i} = \mathcal{H}(m)^{u_i}$ and sends it up to her parent router in the multicast tree.*

   (c) *Intermediate routers check the correctness of the received signatures, aggregate them by generating a multisignature on m and send the aggregated signature up to their parent router in the multicast tree.*

   (d) *Upon reception of the final multisignature, the source checks its correctness.*

Some remarks on Protocol 7 are in order:

- The final multisignature received by the source can be used to prove to a third party that a specific subset of customers signed the receipt and the payment corresponding to the content described in m. Thus Protocol 7 results in a verifiable solution for pay-as-you-watch multicast transmission.

- In case one of the customers leaves the group or fails to send a valid signature, a new rekeying process will be performed so that the failing customer is excluded from knowledge of the new session key.

- The size of the multisignature does not increase when aggregating signatures. This makes our proposal scalable, because the source will not be imploded by reception of a final aggregated signature which has the same size as the individual customer signatures.

Note that, whereas the security of the system relies on the security of the cryptographic tools themselves, the performance of the system is directly related to the computing power of the nodes and the client hardware.

4.4 Multicast fingerprinting

In this section an overview of fingerprinting schemes for multicast is described. bla bla bla blabla bla bla bla bla bla bla bla bla bla bla blabla bla bla blabla bla bla blabla bla bla blabla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

4.4.1 Proposals for multicast fingerprinting

Three main solutions for multicast fingerprinting can be found in the literature, which are described next.

Trusted receiver devices

In [Bao00], a multicast/broadcast source sends encrypted content to receivers, who can decrypt it using a tamper-resistant device (*e.g.* smart card). Each receiver device is identified by its serial number sn and stores a private key SK_{sn}. There is a public key PK_{sn} corresponding to SK_{sn}. Broadcast content is encrypted under a symmetric session key. This session key is sent encrypted under the public keys of the receivers who should be able to access the content.

When receiver device sn decrypts a session key using SK_{sn}, it also fingerprints the decrypted content by embedding sn in it.

This solution can be really implemented but has some drawbacks: collusions are not resisted unless codewords of a collusion-secure fingerprinting scheme are embedded in lieu of serial numbers sn. Last but not least, if trusted receiver devices could really be trusted, copy detection could be abandoned in favour of copy prevention (but copy prevention based on trusted devices has always been cracked).

Distributed marking fingerprinting

The Watercasting system [Brwn99] is based on dividing the content into packets and preparing d versions of the same packet, being d the depth of

the multicast tree. All packet versions are sent to the multicast tree and routers must discard some of the versions, in such a way that clients at the leaves of the multicast tree will get the complete content, but marked differently depending on which version has been received for each packet.

This system has several drawbacks:

- The policy of packet discarding at each router must be centrally scheduled by the source.

- Assuming that a trusted router infrastructure is available may not be realistic.

- Given that d marked versions of each packet are prepared, then the source needs to transmit d times the actual content (bandwidth waste, mainly near the source). On the other hand, d is assumed to be moderately large for a typical large-audience multicast tree.

On the other hand, the WHIM system [Judge00] is another example of distributed marking fingerprinting in that the content is marked by distributed trusted multicast routers that embed a different portion of the mark into the content as the latter is being transmitted through the routing path. In this way, recovering the fingerprint from an illegal copy allows tracing the path followed by the content, which discloses the final recipient (who is the likely redistributor). Despite of being an attractive solution, it has the drawback that deploying a publicly-available trusted distributed system is not an easy task. Furthermore, in [Judge00] the problem of collusion attacks is not addressed.

Encryption-based fingerprinting

In [Ande98], a solution called Chameleon is proposed, whereby contents are encrypted once by the source and recovered with slight differences (marks) by each receiver. This is possible because each receiver has a key that slightly differs from the keys used by the source and other receivers. The main advantage of Chameleon is that content is only transmitted once; its main shortcoming in that noise addition or content compression/decompression cause the system to fail.

In [Parvi01] an encryption-based solution is proposed for fingerprinting in a multicast environment. This solution is based on dividing the multimedia stream into packets. In it has been shown that this proposal is not resistant for collusions of 3 buyers. This is explained next.

4.4.2 Encryption-based fingerprinting

This solution described in [Parvi01] is based on dividing the multimedia stream into packets. The source generates two versions of each packet by embedding two different watermarks. Then each watermarked packet is encrypted with a different random key. The problem is now reduced to assigning a random bit string to each customer.

The proposal can be described as follows:

- For a media stream divided into k packets, $2k$ random encryption keys are needed.
- For each packet P_i, two different watermarked packets, P_i^0 and P_i^1, are generated.

- Packet P_i^0 is encrypted using key k_i^0 and P_i^1 is encrypted using key k_i^1. Both encrypted packets are multicast to all receivers.

- For the i-th packet, each customer is randomly given one of its two decryption keys. If a customer is assigned $\{k_1^0, k_2^1, k_3^1, k_4^0, \ldots\}$ then she will be able to recover $P_1^0, P_2^1, P_3^1, P_4^0, \ldots$ after decryption.

Each buyer receives a slightly different copy of the content as a composition of the different decrypted packets.

After finding an illegal copy of the content, recovery of the watermark embedded into each packet allows the sequence of keys assigned to the subscriber who received the content to be determined. The fingerprint is built from the sequence of keys by simply assigning a 0 value to the i-th bit if the subscriber had key k_i^0 and a 1 if she had k_i^1.

Collusion attacks and random key assignment

The proposal in [Parvi01] provides collusion security as long as the number of colluders remains reduced and the collusion-generated stream consists of a random composition of packets. It is based on accusing the subscriber whose fingerprint is the nearest, using Hamming distance, to the recovered one.

From our point of view, the assumption that colluders use a random collusion strategy is not realistic. In fact, they can use the so-called *minority* strategy. We show in what follows that such a strategy defeats the tracing properties of the scheme.

Let us assume that k-bit long marks are embedded into the content. This means each buyer will obtain a different version of the stream, embedding a

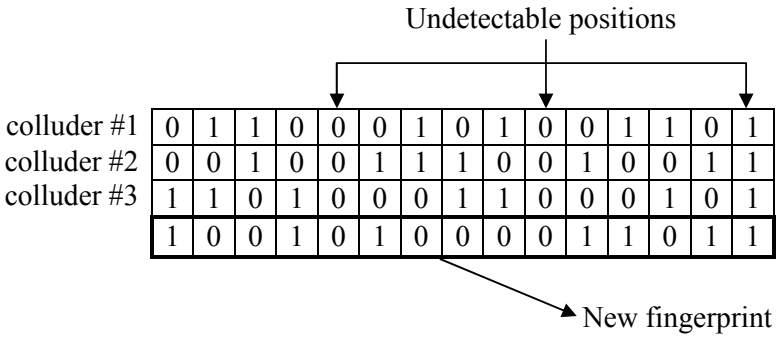

Figure 4.2: A new bit string generated using the minority strategy in a 3-collusion attack.

k-bit long fingerprint. Next, we show how a collusion of only 3 participants can defeat the tracing properties of such a system:

- Detectable fragments of the stream are those for which not all colluders receive the same version. In detectable fragments of a 3-collusion, there are always two colluders who have the same version, which is called the *majority* fragment, while the third colluder has a different version, the *minority* fragment.

- The so-called *minority* collusion strategy consists of always choosing the *minority* fragment of detectable fragments when constructing the new stream. In Figure 4.2, a new fingerprint is obtained from a collusion of 3 buyers.

- If participants in a 3-collusion were assigned random k-bit keys, one would expect each stream to contain $k/4$ undetectable fragments, $k/2$ majority fragments and $k/4$ minority fragments.

- The above means that, after a collusion using a *minority* strategy,

one expects each colluder's fingerprint to differ in $k/2$ bits from the mark contained in the collusion-generated fingerprint. Note that the expected Hamming distance to all other fingerprints assigned to buyers who did not participate in the collusion is also $k/2$ bits. Thus, in this case, colluder tracing is not possible.

Empirical results

The proposal in [Parvi01] was simulated. When dealing with 100-bit or longer fingerprints and collusions of 3 customers using a minority strategy, it was easy to find random bit strings that were nearer to the collusion-generated fingerprint than to fingerprints assigned to colluders.

Simulations clearly showed that the fingerprint generated through a collusion using a minority strategy was at a distance from colluder fingerprints which were similar to the distance to random strings belonging to non-colluders.

For instance, for fingerprints of length 100, $N = 13$ buyers, and collusions of $c = 3$ buyers, the following results were obtained:

- The collusion-generated fingerprint was at a distance of 46, 49 and 49 bits from colluders' fingerprints.
- The collusion-generated fingerprint was at a distance of 48, 47, 59, 50, 61, 45, 53, 53 ,52, 50 from non-colluders' fingerprints.

In this example, it can be seen that the honest customer whose fingerprint is at distance 45 would be erroneously accused of redistribution.

Experiments also showed that, the longer the fingerprints, the higher is the probability of accusing innocent customers.

4.4.3 c-secure fingerprinting for multicast delivery

In the previous section, it has been stated that encryption-based fingerprinting is a good solution as it does not require distributed trust. Nevertheless, a random assignment of decryption keys, as proposed in [Parvi01], enables a set of colluders to easily generate a stream whose fingerprint does not identify any of them, by using the *minority* strategy.

To solve the aforementioned drawback, a key assignment based on Boneh-Shaw collusion secure fingerprinting codes is proposed in [Mart03b]. In our proposal:

- The merchant estimates the maximum number of potential subscribers to the multicast session N and chooses the desired parameters c and ϵ.
- A c-secure fingerprinting code consisting of N codewords with probability ϵ of failed re-identification is constructed. The codewords will have length l (See Equation 2.1).
- As in [Parvi01], the source divides the multimedia stream into l packets and generates two watermarked encrypted versions of each packet.
- Then, each receiver is assigned a different codeword and receives the decryption keys as follows: For the i-th packed, if the i-th bit of the assigned codeword is 0, the receiver will receive k_i^0 otherwise she will receive k_i^1.

The fingerprint is recovered in the same way described previously. After recovery, the tracing algorithm described in [Bone95] is applied, which will identify one of the colluders with probability at least $1 - \epsilon$, as long as the collusion size is not greater than c.

Chapter 5

Privacy of Customer Data

As mentioned before, customer privacy must be preserved from disclosing by third parties who may use data mining techniques in order to improve customer-merchant relationships. This thesis ends presenting a proposal for statistical data protection, which combines perturbative methods with a process of synthetic data generation, the latter presented in our contribution [Mate04].

5.1 Making customer statistics transferable

Let D be a microdata set with customer data collected by a pay-as-you-watch service. This set includes either categorical or continuous/quantitative variables, such as: customer name, customer IP, customer age, money spent, time spent in the service, etc.

In order to make the customer statistics transferable, the following procedure is followed (See Figure 5.1):

1. For categorical data, two kinds of variables must be distinguished:

   - Variables that should be anonymized, for example name or IP, are converted to an identifier or pseudonym.
   - Variables that should not be anonymized, are using a rank swapping procedure [Moor96]. Rank swapping is a perturbative method: its purpose is to swap each value of a certain variable with another value randomly chosen in the same variable.

2. For continuous or quantitative data, a set of synthetic data is generated, using the algorithms described in Section 5.2 below.

3. Either categorical and continuous data are merged in the transferable data set D_t.

Algorithms in Section 5.2 provide a non-iterative method for generating continuous synthetic microdata is proposed. The implementation of this method results in a fast algorithm which *exactly* reproduces the means and the covariance matrix of the original data set and whose running time grows *linearly with the number of records*. Exact preservation of the original covariance matrix implies that variances and Pearson correlations are also exactly preserved in the synthetic data set. Like in any synthetic data generator, the number of records in the synthetic data set can differ from the number of records in the original data set.

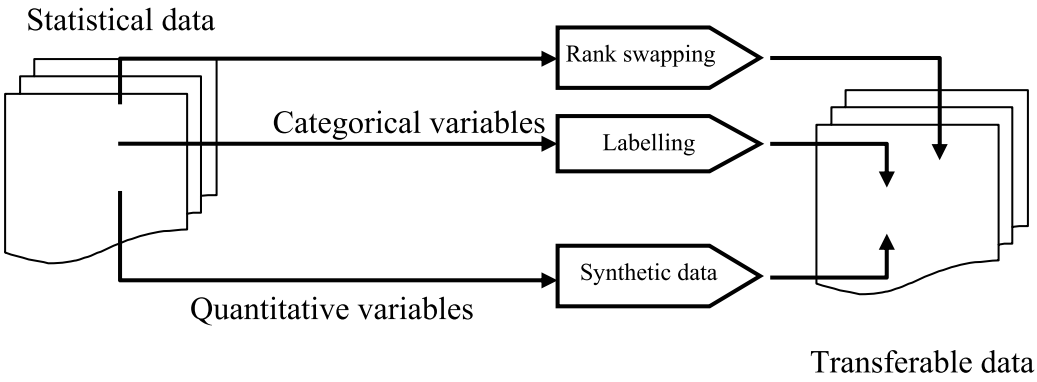

Figure 5.1: Procedure for generation of transferable data.

5.1.1 Background on synthetic data generation

Publication of simulated —*i.e.* synthetic— data was proposed long ago as a way to guard against statistical disclosure. In fact, as early as 1993, Rubin [Rubi93] suggested creating an entirely synthetic data set based on the real survey data and multiple imputation. Specific case studies of synthetic microdata generated by multiple imputation were presented in [Kenn99, Kenn99a]. Although the results were fairly promising, the multiple imputation approach requires complex models and software, which greatly reduces its appeal in many situations.

In [Domi01a, Domi01] comparisons were presented for measuring the performance of microdata masking methods in terms of information loss and disclosure risk. Based on the proposed measures, it was shown in [Sebe02a] how to improve the performance of any particular masking method. In particular, post-masking optimization was discussed for preserving as much as possible the moments of first and second order (and thus multivariate statistics) without increasing the disclosure risk. The technique proposed could also be used

for synthetic microdata generation and could be extended for preservation of all moments up to m-th order, for any m. The shortcoming of this approach is its computational complexity: the optimization problem is solved using an iterative refinement approach, which may be quite time-consuming when the involved data sets are large.

Latin Hypercube Sampling (LHS) appears in the literature as another method for generating multivariate synthetic data sets. In [Hunt98], authors improve the LHS updated technique of [Flor92], but the proposed scheme is still time-intensive even for a moderate number of records. In [Dand02], LHS is used along with a rank correlation refinement to reproduce both the univariate (*i.e.* mean and variance) and multivariate structure (in the sense of rank correlation) of the original data set. This method also permits flexibility in the size of the synthetic data set that is generated. In summary, LHS-based methods rely on iterative refinement, are time-intensive and their running time does not only depend on the number of values to be reproduced, but on the starting values as well.

5.2 A low-cost method for synthetic microdata generation

Let X be an original microdata set, with n records and m variables. Let X' be the synthetic microdata set to be generated, with n' records and m variables. In fact, X can be viewed as an $n \times m$ matrix and X' can be viewed as an $n' \times m$ matrix. The method presented in this section will ensure that both univariate and multivariate statistical properties of X, such as mean and covariance, are exactly reproduced in the resulting X'.

The algorithm below constructs X' from X:

Algorithm 1 (Basic procedure)

1. *Generate A, which is a random $n' \times m$ matrix, such that the covariance matrix of A is the identity matrix.*

2. *Compute the covariance matrix C of the original microdata matrix X.*

3. *Use the Cholesky decomposition on C to obtain*

$$C = U^t \times U$$

where U is an upper triangular matrix and U^t is the transposed version of U.

4. *Obtain the synthetic microdata set X' as a matrix product:*

$$X' = A \cdot U$$

Note that the covariance matrix of X' equals the covariance matrix of X [Sche62].

5. *Due to the construction of matrix A, the mean of each variable in X' is 0. In order to preserve the mean of variables in X, a last adjustment is performed. If $\bar{x}_j$ be the mean of the j-th variable in X, then $\bar{x}_j$ is added to the j-th column (variable) of X':*

$$x'_{ij} := x'_{ij} + \bar{x}_j \quad \textit{for } i = 1, \cdots, n' \textit{ and } j = 1, \cdots, m \tag{5.1}$$

We now need to specify how to construct a random $n' \times m$ matrix A, whose covariance matrix is the $m \times m$ identity matrix.

Algorithm 2 (Construction of matrix A)

1. *Generate A as an $n' \times m$ matrix with random elements $a_{i,j}$. View the m columns of A as samples of variables $A_1, \cdots, A_m$. If $Cov(A_j, A_{j'})$ is the covariance between variables A_j and $A_{j'}$, the objective of the algorithm is that*

$$Cov(A_j, A_{j'}) = \begin{cases} 1 & \text{if } j = j' \\ 0 & \text{otherwise} \end{cases}$$

   for $j, j' \in \{1, \cdots, m\}$.

2. *Let $\bar{a}_1$ be the mean of A_1. Let us adjust A_1 as follows:*

$$a_{i,1} := a_{i,1} - \bar{a}_1 \quad i = 1, \ldots, n'$$

   The mean of the adjusted A_1 is 0.

3. *In order to reach the desired identity covariance matrix, some values of variables $A_2, \cdots, A_m$ must change. For $v = 2$ to m do:*

   (a) *Let $\bar{a}_v$ be the mean of variable A_v.*

   (b) *For $j = 1$ to $v - 1$, the covariance between variables A_j and A_v is*

$$Cov(A_j, A_v) = \frac{\sum_{i=1}^{n'} a_{i,j} \cdot a_{i,v}}{n'} - 0 \cdot \bar{a}_v = \frac{\sum_{i=1}^{n'} a_{i,j} \cdot a_{i,v}}{n'}$$

   (c) *In order to obtain $Cov(A_j, A_v) = 0, \quad j = 1 \ldots v - 1$, some elements $a_{i,v}$ in the v-th column of A are assigned a new value. Let $x_1, \ldots, x_{v-1}$ be the unknowns for the following linear system of $v - 1$ equations:*

$$\frac{\sum_{i=1}^{n'-v+1} a_{i,j} \cdot a_{i,v} + \sum_{i=1}^{v-1} a_{n'-v+1+i,j} \cdot x_i}{n'} = 0 \text{ for } j = 1 \ldots v - 1$$

that is

$$\sum_{i=1}^{n'-v+1} a_{i,j} \cdot a_{i,v} + \sum_{i=1}^{v-1} a_{n'-v+1+i,j} \cdot x_i = 0 \;\; \textit{for } j = 1 \ldots v-1$$

Once the aforementioned linear system is solved, the new values are assigned:

$$a_{n'-v+1+i,v} := x_i \;\; \textit{for } i = 1 \ldots v-1$$

(d) Let $\bar{a}_v$ be the mean of variable A_v. A final adjustment on A_v is performed to make its mean 0:

$$a_{i,v} = a_{i,v} - \bar{a}_v \;\; \textit{for } i = 1 \ldots n'$$

4. In the last step, values in A are adjusted in order to reach $Cov(A_j, A_j) = 1$ for $j = 1 \ldots m$. If σ_j is the standard deviation of variable A_j, the adjustment is computed as:

$$a_{i,j} := \frac{a_{i,j}}{\sigma_j}, \;\; i = 1 \ldots n', j = 1 \ldots m$$

With the construction proposed in this section, the number of records n' in X' does not depend on the number of records n in X. Thus, disclosure of n is prevented, which may be useful in some situations. On the other hand, Algorithm 2 does not need to be run each time Algorithm 1 is run. In other words, if $X_1, X_2, \ldots X_u$ are original microdata sets, each with n_i records, $i = 1 \ldots u$, and m variables, then u synthetic microdata sets $X'_1, X'_2, \ldots X'_u$ can be generated, each with n' records and m variables, *with a single $n' \times m$ matrix A.*

Table 5.1: Running time (in seconds) on a 1.7 GHz desktop Intel PC under a Linux OS. Note that time for random matrix generation is included

	Number of variables			
Number of records	5	10	25	50
1,000	0.00	0.00	0.05	0.31
10,000	0.05	0.19	1.26	5.31
100,000	0.49	1.93	12.41	51.15

5.2.1 Properties of the proposed scheme

Performance and complexity

To simplify the performance and complexity analysis presented here, we assume that a synthetic data set of size $n \times m$ is generated from an original data set of the same size, *i.e.* $n' = n$. The method has been tested with several data set sizes and execution times are shown in Table 5.1.

The computational complexity for the proposed method will next be estimated. Let n be the number of records, m the number of variables and assume for simplicity $n' = n$. Then the complexities of the various operations are as follows:

- Calculation of the covariance matrix: $\mathcal{O}(n + m^2)$.
- Cholesky decomposition: $\mathcal{O}(m^3/6)$ (see [Pres93]).
- Calculation of A: $\mathcal{O}(2nm + 2m^3 + 2m^4/3)$, where the term $2m^4/3$ is the cost of solving a Gauss system m times [Pres93].
- Matrix product: $\mathcal{O}(nm^2)$.

- Mean adjustment: $\mathcal{O}(nm)$.

In summary, the overall complexity is $\mathcal{O}(nm + 2m^4/3) = \mathcal{O}(n + m^4)$. To understand this complexity, one should realize that, in general, the number of records n is much larger than the number of variables m, *i.e.* $n >> m$. The strong point of this proposal is that *its complexity is linear in the number of records*. It must also be kept in mind that, as pointed out at the end of Section 5.2, matrix A can be re-used to generate several synthetic microdata sets, which greatly reduces computation.

Data utility

As stated in Section 5.1, the proposed scheme exactly reproduces the statistical properties of the original data set. In particular:

- The means of variables in the original data set X are exactly preserved in the synthetic data set X'.

- The covariance matrix of X is exactly preserved in X' (see [Sche62]). Thus, in particular:

  - The variance of each variable in X is preserved in X'.

  - The Pearson correlation coefficient matrix of X is also exactly preserved in X', because correlations are obtained from the covariance matrix.

5.2.2 Empirical work

The score

In the experiments conducted to measure the disclosure risk and the information loss in the synthetic data sets produced by our method, we use the $Score'$ defined in [Dand02a]. $Score'$ is a modification of the original $Score$ defined in [Domi01] to deal with synthetic data generation in which the number of records of the synthetic data set differs from the number of records in the original data set. We briefly recall the definition of $Score'$:

$$Score' = 0.5 \cdot IL + 0.25 \cdot DLD + 0.25 \cdot ID$$

where IL stands for information loss, DLD refers to distance-based record linkage and ID stands for interval disclosure. DLD and ID are disclosure risk measures. IL measures how different is the synthetic data set from the original one. IL_1 is a component of IL which compares the individual original and synthetic values, whereas the remaining components for IL reflect how different are univariate and multivariate statistics between X and X'. See [Dand02a] on how to compute IL, DLD and ID.

The data set

The microdata set for testing was constructed using the Data Extraction System of the U.S. Census Bureau (http://www.census.gov/DES) and contains $n = 1080$ registers for $m = 13$ continuous variables. This data set was also used in [Dand02a, Domi01a, Domi01].

The results

As mentioned in Step 1 of Algorithm 2, A is initially composed of random values. It must be noticed that whatever the magnitude of the values in X is, the range in which the initial random values for A are picked —say between 0 and 100— does not affect the results. The score for a typical execution is shown in Table 5.2.

Note that, since most of the values in A are random, the result for IL_1 shows that most of values in X are substantially different in X'. The point is that the remaining components IL_2, IL_3, IL_4 IL_5 of the information loss show that the statistical properties listed in Section 5.2.1 are exactly fulfilled (those measures do not appear as exactly 0 due to rounding errors). On the other hand, the disclosure risk measures DLD and ID are lower than those obtained for the LHS-Based method and reported in [Dand02a].

Due to the randomness of matrix A, different runs of the method will result in different synthetic data sets and, consequently, the resulting *Score'* will change. In 10 executions, an average value of 12.14 for DLD was obtained, with a standard deviation of 0.97; the average obtained for ID was 37.99 with a standard deviation of 5.11.

If synthetic data sets are generated whose number n' of records is not the same as the number n of original records, the values for DLD and ID are maintained (see Table 5.3). Hence, the disclosure risk measures do not depend on the number of records of the synthetic data set.

Non-random matrix A

In order to reduce the information loss component IL_1 (individual record comparison), one could think of choosing the initial values of matrix A in a "clever" way rather than using initial random values. For example, A could be the result of masking the original data set X using a perturbative masking method (see [Domi01b]). This leads to a number of records in X' which equals the the number of records in X.

Table 5.4 shows the results obtained when different perturbative masking methods are used. The lowest value for DLD and ID is reached when A has been obtained using microaggregation [Domi03] with parameter $k = 20$ The lowest value for IL occurs for additive noise with parameter 2%.

Table 5.2: Values of *Score'* for the synthetic data

Measure	Value
IL_1	544.53
IL_2	3.53565e-05
IL_3	3.25579e-04
IL_4	1.81034e-03
IL_5	1.90171e-04
IL	108.90
DLD	9.10
ID	33.21
$Score'$	65.0338

Table 5.3: DLD and ID values for synthetic data sets with n' records

Number of records	DLD	ID
500	13.40	33.27
2,000	12.65	45.65
8,000	15.14	34.14
10,000	12.54	34.77
20,000	11.86	39.60

Table 5.4: Results when using a masked data set as A

Masking method	$Score'$	IL'	DLD'	ID'
noise.02	36.42	36.41	20.85	52.00
noise.08	38.04	40.69	19.48	51.32
noise.14	35.61	36.50	18.20	51.24
microag.k=5	37.74	39.70	20.19	51.37
microag.k=10	40.96	45.21	23.28	50.15
microag.k=20	37.42	42.85	16.30	47.68
rankswap.5	38.76	43.89	16.77	50.50
rankswap.15	42.28	50.77	16.93	50.64

Chapter 6

Conclusions

6.1 Results of this thesis

6.2 Future research

Appendix A

List of acronyms

ADSL Asynchronous Digital Subscriber Line

CA Certification Authority

CDH Computational Diffie-Hellman problem

CDROM Compact Disc Read-Only Memory

CRM Customer Relationship Management

DCT Discrete Cosine Transform

DDH Decisional Diffie-Hellman problem

DEW Difference Energy Watermarking

DLD Distance-based record linkage

DRM Digital Rights Management

DVB Digital Video Broadcast standard

DVD Digital Versatile Disc / Digital Video Disc

GDH Gap-Diffie-Hellman group

GPRS General Packet Radio Service

GSM Global System for Mobile communication

HDTV High-Definition Television

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

ID Interval Disclosure

IL Information Loss

IP Internet Protocol

JPEG Joint Picture Experts Group

Kbps Kilobits per second

LAN Local Area Network

LHS Latin Hypercube Sampling

MPEG Moving Picture Experts Group

P2P Peer to Peer network

PC Personal Computer

PDA Personal Digital Assistant

RSA Rivest, Shamir and Adleman

SDC Statistical Disclosure Control

SMS Short Messaging System

TTP Trusted Third Party

UMTS Universal Mobile Telecommunication System

VBR Variable Bit Rate

WAP Wireless Application Protocol

WHIM Watermarking with a Hierarchy of InterMediaries

WLAN Wireless Local Area Networks

WMV Windows Media Video

Our Contributions

[Domi04] J. Domingo-Ferrer, A. Martínez-Ballesté, F. Sebé, "Secure reverse communication in a multicast tree", in *Third International IFIP-TC6 Networking Conference - NETWORKING 2004*, LNCS 3042, pp.807-816, Berlin: Springer-Verlag, 2004.

[Mart04] A. Martínez-Ballesté, F. Sebé and J. Domingo-Ferrer, "Large-Scale Pay-As-You-Watch for Unicast and Multicast Communications", in *1st International Conference on Trust and Privacy in Digital Business-TrustBus'04*, LNCS , pp. , Berlin: Springer-Verlag, 2004 (to appear).

[Mate04] J. M. Mateo-Sanz, A. Martínez-Ballesté and J. Domingo-Ferrer, "Fast Generation of Accurate Synthetic Microdata", in *Privacy in Statistical Databases-PSD 2004*, LNCS XXXX, pp. , Berlin: Springer-Verlag, 2004.

[Mart04] A. Martínez-Ballesté, F. Sebé and J. Domingo-Ferrer, "Secure large-scale bingo", in *IEEE International Conference on Information Technology: Coding and Computing-ITCC'2004*, Piscataway NJ: IEEE Computer Society, pp. 758-762, 2004.

[Mart03] A. Martínez-Ballesté, F. Sebé, J. Domingo-Ferrer and M. Soriano, "Practical asymmetric fingerprinting with a TTP", in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications-DEXA'2003*, Los Alamitos CA: IEEE Computer Society, 2003.

[Mart03a] A. Martínez-Ballesté, J. Domingo-Ferrer and Francesc Sebé, "MINPAY: a Multi-device INternet PAY-as-you-watch system", in *IEEE International Conference on Information Technology: Coding and Computing-ITCC'2003*, Piscataway NJ: IEEE Computer Society, pp. 258-262, 2003.

[Mart03b] A. Martínez-Ballesté, F. Sebé and J. Domingo-Ferrer, "Fingerprinting schemes for multicast delivery", in *IEEE International Conference on Information Technology: Research and Education-ITRE'2003*, Los Alamitos CA: IEEE Computer Society, 2003.

[Domi02a] J. Domingo-Ferrer and A. Martínez-Ballesté, "STREAMOBILE: Pay-per-view video streaming to mobile devices over the Internet", in *Proceedings of the 13th International Workshop on Database and Expert Systems Applications-DEXA'2002*, eds. A. Min Tjoa and R. R. Wagner, Los Alamitos CA: IEEE Computer Society, pp. 418-422, 2002.

[Domi02b] J. Domingo-Ferrer, A. Martínez-Ballesté and F. Sebé, "MICROCAST: Smart card based (micro)pay-per-view for multicast services", a *Proceedings of IFIP/USENIX 5th Smart Card Research and Advanced Application Conference-CARDIS'2002*, Berkeley CA: USENIX, pp. 125-134, 2002.

[Mart03c] A. Martínez-Ballesté, F. Sebé and J. Domingo-Ferrer, "Aspectos prácticos de la protección de propiedad intelectual en contenidos multimedia", in *Actas del Simposio Español de Comercio Electrónico - SCE 2003*, Fundación DINTEL, pp. 219-228, 2003.

[Domi02c] J. Domingo-Ferrer, A. Martínez-Ballesté and F. Sebé, "Vídeos de pago en Internet", in *Boletín RedIRIS*, n. 62-63, pp. 6-9, ISSN 1139-207X, December 2002/January 2003.

Bibliography

[Ande95] R. Anderson, C. Manifavas and C. Sutherland, "NetCard - A practical electronic cash system", 1995. Available from author: Ross.Anderson@cl.cam.ac.uk

[Ande98] R. Anderson and C. Manifavas, "Chameleon – A New Kind of Stream Cipher.", in *Fast Software Encryption*, pp. 107-113, 1997.

[Augo98] D. Augot, J.F. Delaigle and C. Fontaine "DHWM: a scheme for managing watermarking keys in the Aquarelle multimedia distributed system", *Computer Security - ESORICS 98*, LNCS 1485, Springer-Verlag, pp 241-255, 1998.

[Bao00] F. Bao, "Multimedia Content Protection by Cryptography and Watermarking in Tamper-resistant Hardware", in *Proceedings of the 2000 ACM workshops on Multimedia*, pp. 139-142, 2000.

[Bone95] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data", in *Advances in Cryptology-CRYPTO'95*, LNCS 963, Springer-Verlag, pp. 452-465, 1995.

[Bold03] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme",

in *Int. Workshop on Theory and Practice in Public Key Cryptography-PKC'2003*, LNCS 2567, Springer-Verlag, pp. 31-46, 2003.

[Bone96] L. Boney, A. H. Tewfik and K. N. Hamdy, "Digital Watermarks for Audio Signals", in *International Conference on Multimedia Computing and Systems*, pp.473-480, 1996.

[Brand93] S. Brands, "Untraceable off-line cash in wallets with observers", in *Advances in Cryptology-CRYPTO'93*, LNCS 773, Berlin: Springer-Verlag, pp. 302-318, 1993.

[Brwn99] I. Brown, C. Perkins and J. Crowcroft, "Watercasting: Distributed Watermarking of Multicast Media", in *Proceedings of the First International Workshop on Networked Group Communication*, pp. 286-300, 1999.

[Calv01] K. L. Calvert, J. Griffioen, B. Mullins, A. Sehgal and S. Wen, "Concast: Design and implementation of an active network service", *IEEE Journal on Selected Area in Communications (JSAC)*, vol. 19, no. 3, Mar. 2001.

[Camp03] J. Camps Aragonès, *Servei WWW sobre GPRS*, Graduate thesis, Department of Computer Enginering and Maths, Universitat Rovira i Virgili, 2003.

[Cinemanow] CinemaNow, `http://www.cinemanow.com`

[Chaum82] D. Chaum, "Blind signatures for untraceable payments", in *Advances in Cryptology - CRYPTO'82*, Plenum Press, pp. 199-203, 1983.

[Dand02] R. A. Dandekar, M. Cohen and N. Kirkendall, "Sensitive micro data protection using latin hypercube sampling technique", in *Inference*

Control in Statistical Databases, vol. LNCS 2316, pp. 245-253, Springer, 2002.

[Dand02a] R. A. Dandekar, J. Domingo-Ferrer and F. Sebé, "LHS-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection", in *Inference Control in Statistical Databases*, ed. J. Domingo-Ferrer, vol. LNCS 2316, pp. 153-162, Springer, 2002.

[DeCS] `http://www.lemuria.org/DeCSS`

[Deer96] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, "The PIM architecture for wide-area multicast routing", *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 153-162, Apr. 1996.

[Deer98] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer and L. Wei, "Protocol independent multicast version 2 dense mode specification", IETF Internet Draft, Nov. 1998.

[Domi98] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Efficient smartcard based anonymous fingerprinting" in *Smart Card Research and Advanced Application, CARDIS'98*, LNCS 1820, Springer-Verlag, pp. 231-238, 1998.

[Domi99] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Spending programs: A tool for flexible micropayments", in *Information Security-ISW'99*, eds. M. Mambo and Y. Zheng, LNCS 1729, Springer-Verlag, pp. 1-13, 1999.

[Domi00] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Short collusion-secure fingerprints based on dual binary Hamming codes", *Electronics Letters (IEE)*, vol. 36, no. 20, pp. 1697-1699, Sep. 2000.

[Domi01] J. Domingo-Ferrer, J. M. Mateo-Sanz and V. Torra, "Comparing SDC methods for microdata on the basis of information loss and disclosure risk", *Proceedings of ETK-NTTS 2001*, Luxemburg: Eurostat, pp. 807-825, 2001.

[Domi01a] J. Domingo-Ferrer and V. Torra, "A quantitative comparison of disclosure control methods for microdata", in *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, eds. L. Zayatz, P. Doyle, J. Theeuwes and J. Lane, Amsterdam: North-Holland, 2001, pp. 111-134.

[Domi01b] J. Domingo-Ferrer and V. Torra, "Disclosure protection methods and information loss for microdata", in *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, eds. L. Zayatz, P. Doyle, J. Theeuwes and J. Lane, Amsterdam: North-Holland, 2001, pp. 91-110.

[Domi02] J. Domingo-Ferrer and F. Sebé, "Enhancing watermark robustness through mixture of watermarked digital objects", in *IEEE Intl. Conf. on Information Technology: Coding and Computing-ITCC'2002*, IEEE Computer Society, pp. 85-89, 2002.

[Domi03] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control", *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 189-201, Feb. 2002.

[Eski03] A. M. Eskicioglu, "Protecting intellectual property in digital multimedia networks", *IEEE Computer*, vol. 38, no. 7, pp. 39-45, July 2003.

[Eski03a] A. M. Eskicioglu, "An Optimal Watermarking Scheme based on Singular Value Decomposition", in *IASTED International Conference on Communication, Network and Information Security (CNIS 2003)*, 2003. Can be downloaded from author homepage: `http://www.sci.brooklyn.cuny.edu`

[EuroOn] Europe Online, `http://www.europeonline.com`

[Fen97] W. Fenner. Internet Group Management Protocol, Version 2. RFC 2236, November 1997.

[Fern02] M. Fernandez and M. Soriano, "Fingerprinting concatenated codes with efficient identification", in *International Security Conference ISC*, LNCS 2433, pp. 459-470, Springer-Verlag, 2002.

[Flor92] A. Florian, "An efficient sampling scheme: updated latin hypercube sampling", *Probabilistic Engineering Mechanics*, no. 7, pp. 123-130, 1992.

[FoxPPV] Fox Pay-per-View, `http://www.foxppv.com`

[Glas95] S. Glassman, M. Manasse, M. Abadi, P. Gauthier and P. Sobalvarro, "The Millicent protocol for inexpensive electronic commerce", in *World Wide Web Journal, 4th Intl. World Wide Web Conference Proceedings*, O'Reilly, pp. 603-618, 1995.

[Hart98] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video", in *Signal Processing*, vol. 66, no. 3, pp. 283–301, 1998.

[Haus96] R. Hauser, M. Steiner and M. Waidner, "Micro-payments based on *i*KP", IBM Research Report 2791, presented also at SECURICOM'96. http://www.zurich.ibm.com/Technology/Security/ publications/1996/HSW96.ps.gz

[Hunt98] D. E. Huntington and C. S. Lyrintzis, "Improvements to and limitations of Latin hypercube sampling", *Probabilistic Engineering Mechanics*, vol. 13, no. 4, pp. 245-253, 1998.

[Hunt02] B. Hunt, "Industry ready to bring up revenues from mobile data", Financial Times IT and FT journal, Nov. 2002.

[Ison03] ISONIFY, Internet Audio and Video Technologies, `http://www.insonify.com/`

[Judge00] P. Judge and M. Ammar, "WHIM: Watermarking multicast video with a hierarchy of intermediaries", in *Proceedings of NOSSDAV 2000*, 2000.

[Jutl96] C. Jutla and M. Yung, "PayTree: " Amortized-signature" for flexible micropayments", in *Second USENIX Workshop on Electronic Commerce*, Oakland CA, 1996.

[Kenn99] A. B. Kennickell, "Multiple imputation and disclosure control: the case of the 1995 Survey of Consumer Finances", in *Record Linkage Techniques*, Washington DC: National Academy Press, 1999, pp. 248-267.

[Kenn99a] A. B. Kennickell, "Multiple imputation and disclosure protection: the case of 1995 Survey of Consumer Finances", in *Statistical Data Protection*, Luxemburg: Office for Official Publication of the European Communities, 1999, pp. 177-206.

[Lang01] G.C. Langelaar, R. L. Lagendijk, "Optimal Differential Energy Watermarking of DCT Encoded Images and Video", in *IEEE Transactions on Image Processing*, vol.10, no.1, January 2001.

[Liu02] R. Liu and T. Tan, "A SDV-Based Watermarking Scheme for Protecting Rightful Ownership", in *IEEE Transactions on multimedia*, vol. 4 no. 1, pp. 121-128, March 2002.

[Kayz00] S. Katzenbeisser and F. Petitcolas, *Information Hiding. Techniques for Stenography and Digital Watermarking*, Artech House, 2000.

[Merk78] R. C. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks", *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 525-530, 1978.

[Mica02] S. Micali and R. L. Rivest, "Micropayments revisited" in *Topics in Cryptology - CT-RSA 2002*, LNCS 2271, Berlin: Springer-Verlag, pp. 149-163, 2002.

[Micr03] Microsoft's Digital Rights Management Scheme - Technical details, `http://cryptome.org/ms-drm.htm`

[Mill99] C. K. Miller, *Multicast Newtorking and Applications*. Reading MA: Addison Wesley, 1999.

[Mini] MiniPay, http://www.minipay.com

[Mont01] F. Montelius, O. Larsson, *Streaming Video in Wireless Networks: Service and Technique*, Master Thesis, Link¨opings universitet (Sweden), 2001. It can be downloaded from `http://www.ep.liu.se/exjobb/isy/2002/3227/`

[Moor96] R. Moore, "Controlled data swapping techniques for masking public use microdata sets", U. S. Bureau of the Census, 1996 (unpublished manuscript).

[Moy94] J. Moy, "Multicast extensions to OSPF", Internet RFC 1584, March 1994.

[MovieL] MovieLink, `http://www.movielink.com`

[Mpeg03] MPEG Pointers and Resources, `http://www.mpeg.org`

[MSEC03] Multicast Security Working Group (MSEC WG). `http://www.securemulticast.org`

[Nist93] National Institute of Standards and Technology, NIST FIPS PUB 180, *Secure Hash Standard*, U. S. Department of Commerce, May 1993.

[NDS] NDS, `http://www.nds.com`

[Odly03] A. M. Odlyzko "The case against micropayments", in *Financial Cryptography: 7th International Conference - FC 2003*, LNCS 2742, pp. 77-83, Berlin: Springer-Verlag, 2003.

[Okam98] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", in *Advances in Cryptology - EUROCRYPT'98*, ed. K. Nyberg, LNCS 1403, Berlin: Springer-Verlag, pp. 308-318, 1998.

[Pack] PacketVideo streaming and player, `http://www.packetvideo.com`

[Parvi01] R. Parviainen y P. Barnes, "Large scale distributed watermarking of multicast media through encryption", *Proceedings of IFIP Communications and Multimedia Security*, pp. 149-158, 2001.

[Pfit96] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting", in *Advances in Cryptology-EUROCRYPT'96*, LNCS 1070, Springer-Verlag, pp. 84-95, 1996.

[Pres93] W. Press, W. T. Teukolsky, S. A. Vetterling and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1993.

[Psou99] K. Psounis, "Active networks: Applications, security, safety and architectures", *IEEE Communication Surveys*, vol. 2, no. 1, pp. 1-16, 1999.

[Quin01] B. Quinn and K. Almeroth, "IP multicast applications: challenges and solutions", Internet RFC 3170, Sept. 2001. `http://www.ietf.org`

[Riad04] S. Riad Ahmed, "Applications of Data Mining in Retail Business", in *IEEE International Conference on Information Technology: Coding and Computing-ITCC'2004*, Piscataway NJ: IEEE Computer Society, pp. 455-459, 2004.

[Rive78] R. L. Rivest, A. Shamir, L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, February 1978.

[Rive95] R. L. Rivest and Adi Shamir, "PayWord and MicroMint: two simple micropayment schemes", Technical Report, MIT LCS, 1995. http://theory.lcs.mit.edu/ rivest

[Rubi93] D. B. Rubin, "Discussion on statistical disclosure limitation", *Journal of Official Statistics*, vol. 9, no. 2, pp. 461-468.

[Sebe00] F. Sebé, J. Domingo-Ferrer and J. Herrera-Joancomartí "Spatial-domain image watermarking robust against compression, filtering, cropping and scaling", in *Information Security - LNCS 1975*, pp.44-53, Berlín, 2000. *LNCS*, vol. 2200, pp. 420-432, 2001.

[Sebe01] F. Sebé and J. Domingo-Ferrer,
"Oblivious image watermarking robust against scaling and geometric distortions", LNCS 2200, Vol. *Information Security*, Springer-Verlag, pp. 420-432, 2001.

[Sebe02] F. Sebé and J. Domingo-Ferrer, "Scattering codes to implement short 3-secure fingerprinting for copyright protection", in *Electronics Letters*, vol. 38, no. 17, pp. 958- 959, August 2002.

[Sebe02a] F. Sebé, J. Domingo-Ferrer, J. Mateo-Sanz and V. Torra, "Post-masking optimization of the tradeoff between information loss and disclosure risk in masked microdata sets", in *Inference Control in Statistical Databases*, ed. J. Domingo-Ferrer, vol. LNCS 2316, pp. 163-171, Springer, 2002.

[Sche62] E. M. Scheuer and D. S. Stoller, "On the generation of normal random vectors", *Technometrics*, no. 4, pp. 278-281, 1962.

[Sety01] I. Setyawan and R. L. Lagendijk "Low bit-rate video watermarking using temporally extended Differential Energy Watermarking algorithm", in *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, vol. 4314, 2001.

[Snoe01] J. Snoeyink, S. Suri and G. Varghese, "A lower bound for multicast key distribution", in *Proceedings of IEEE INFOCOM 2001.* Piscataway NJ: IEEE Computer and Communications Society, pp. 422-431, 2001.

[UnivPPV] Universal Pay-per-View, `http://www.universalppv.net`

[Wagn83] R. Wagner, "Fingerprinting", in *IEEE Symposium on Security and Privacy*, Oakland, pp. 18-22, 1983.

[Wolf03] T. Wolf and S. Y. Choi, "Aggregated hierarchical multicast - A many-to-many communication paradigm using programmable networks", *IEEE Transactions on Systems, Man and Cybernetics - part C: Applications and Reviews*, vol. 33, no. 3, pp. 358-369, Aug. 2003.

Antoni Martínez Ballesté

June 2004