SELECTIVE VIDEO ENCRYPTION OF DISTRIBUTED VIDEO CODED BITSTREAMS AND MULTICAST SECURITY OVER WIRELESS NETWORKS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Hwa Young Um

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2006

Purdue University

West Lafayette, Indiana

To My Parents, Wife, and Kids

ACKNOWLEDGMENTS

I would like to thank my major advisor, Professor Edward J. Delp. I thank him for having my dream come true. It had been amazing for me to become a member of the VIPER lab. I am grateful for his guidance and support. He has provided many opportunities to explore the most interesting and challenging pursuits, and his teachings in scholarly research will be always kept in my mind.

I would also like to acknowledge my other committee members, Professor Charles A. Bouman, Professor Jan P. Allebach, and Professor Fabio A. Milner, for their support. I have been so much enjoying their classes and grateful for their imparting their knowledge to me.

I would like to thank my fellow officemates: Eugene T. Lin, Hyung Cook Kim, Jinwha Yang, Aravind K. Mikkilineni, Anthony F. Martone, Limin Liu, Ashok Raj Kumaran Mariappan, and Oriol Guitart Pla, for their great help and inspiration in my academic research and their sincere friendship. I had been enjoying the time working with them in VIPER.

I have dedicated this document to my mother and father. They have supported me throughout these many years while I have been pursuing my academic career. I thank them for giving me the chance to come to this world. I enjoy it.

Finally, I thank my wife, Yun Mi, and two beautiful kids, Hyun Joon (Andy) and Hyun Woo (Bryant), for their support and love. My wife is my best friend and confidante. I have been so lucky to have her at my side whose support, understanding, encouragement, care, and true love make all the difference in the world.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

λ	Mean of Poisson arrival process (calls/sec)
$1/\mu$	Mean of Exponential interarrival time (1/sec)
G	Generator Matrix
H	Parity-check Matrix

ABBREVIATIONS

AES	Advanced Encryption Standard
BER	Bit Error Rate
BS	Base Station
CAC	Call Admission Control
CCSI	Channel Coding with Side Information
CDMA	Code Division Multiple Access
COD	Coding Style Header
COM	Comments Header
CR	Conditional Replenishment
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DSC	Distributed Source Coding
DVC	Distributed Video Coding
GM	Group Manager
GPS	Global Positioning System
GSAs	Group Security Agents
GSC	Group Security Controller
GSIs	Group Security Intermediaries
IDEA	International Data Encryption Algorithm
IPMP	Intellectual Property Management and Protection
JPEG	Joint Photographic Experts Group
JPSEC	JPEG-2000 Part 8 Security standard
LAN	Local Area Network
LDPC	Low Density Parity Check

LKH	Logical Key Hierarchy
MCP	Motion-Compensated Prediction
MPEG	Moving Picture Experts Group
MS	Mobile Station
MX	Mobile switching eXchanger
NDME	Network Driven Motion Estimation
PCS	Personal Communications Service
PDA	Personal Digital Assistant
PRBG	Pseudo-Random Bit Generator
PSS	Prepositioned Secret Sharing
QCD	Quantization Deader
QoS	Quality of Service
RA	Registration Authority
RCPT	Rate Compatible Punctured Turbo
ROI	Region-Of-Interest
RVLC	Reversible Variable Length Code
SE	Selective Encryption
SHA	Secure Hash Algorithm
SI	Side Information
SCSI	Source Coding with Side Information
SGM	Subgroup Manager
SOT	Start of a Tile parts header
TGDH	Tree-based Group Diffie-Hellman
VEA	Video Encryption Algorithm
WAN	Wide Area Network

ABSTRACT

Um, Hwa Young Ph.D., Purdue University, August, 2006. Selective Video Encryption of Distributed Video Coded Bitstreams and Multicast Security over Wireless Networks. Major Professor: Edward J. Delp.

Here we discuss two security problems, video data encryption and multicast security, in wireless networks.

Selective encryption is a technique that is used to minimize computational complexity or enable system functionality by only encrypting a portion of a compressed bitstream while still achieving reasonable security. For selective encryption to work, we need to rely not only on the beneficial effects of redundancy reduction, but also on the characteristics of the compression algorithm to concentrate important data representing the source in a relatively small fraction of the compressed bitstream. These important elements of the compressed data become candidates for selective encryption. In this thesis, we combine encryption and distributed video source coding to consider the choices of which types of bits are most effective for selective encryption of a video sequence that has been compressed using a distributed source coding method based on LDPC and Turbo codes. Instead of encrypting the entire video stream bit by bit, we encrypt only the highly sensitive bits. By combining the compression and encryption tasks and thus reducing the number of bits encrypted, we can achieve a reduction in system complexity.

Secure multicast protocols are difficult to implement efficiently due to the dynamic nature of the multicast group and scarcity of bandwidth at the receiving and transmitting ends. Mobility is one of the most distinct features to be considered in a wireless network. Moving users onto the key tree causes extra key management resources even though they are still in service. To take care of frequent handoff between

wireless access networks, it is necessary to reduce the number of rekeying messages and the size of the messages. In this dissertation, we design a key management tree such that neighbors on the key tree are also physical neighbors on the cellular network. By tracking the user location, we localize the delivery of rekeying messages to the users who need them. This lessens the amount of traffic in wireless and wired intervals of the network. The group key management scheme uses a pre-positioned secret sharing scheme.

1. INTRODUCTION

As the technology and popularity of wireless systems such as 3G [1], cdma2000 [2], wireless local area network (WLAN) [3], and Portable Internet (WiBro, WiMAX) [4] grow, there has been considerable progress in the area of multimedia streaming over wireless networks in the last few years. Video and data communication is becoming an increasingly important part of the wireless information infrastructure. Here we will discuss video encryption and multicast security problems in wireless networks [5–8].

1.1 Video Encryption in Wireless Networks

Contrasting to classical encryption, security may not be the most important aim for an encryption system for images and videos. Depending on the type of application, other properties, such as speed or bitstream compliance after encryption, might be equally important as well.

A number of different encryption schemes for visual data types have been proposed over the last years. The so called naive method is to take the multimedia bitstream and encrypt this stream with the cryptographically strong ciphers such as AES [9] or DES [10, 11]. Since runtime performance is often very critical in video encoding and decoding, more efficient methods have been proposed. Such systems are denoted as selective or soft encryption that usually trade off runtime performance for security. Intuitively selective encryption (SE) seems to be a good idea in any case since it is always desirable to reduce the computational demand involved in signal processing applications. However the security of such schemes are always lower as compared to full encryption. The only reason to accept this drawback is significant saving in terms of processing time or power. Therefore the environment in which SE should be used

needs to be investigated thoroughly in order to decide whether its use is sensible or not [12].

1.1.1 Background

Unlike text or image, video communication requires huge volumes of data being transmitted in a timely manner, so highly efficient compression must be used. Extensive research in both academia and industry has resulted in a large number of standards in the past decade, such as H.261 [13], H.263 [14], H.263+ [15], H.264 [16–18], MPEG-1 [19], MPEG-2 [20], MPEG-4 [21–23], etc.

Recent technological developments have led to several mobile systems aimed at personal communications services (PCS), supporting both speech and data transmission. Mobile users usually communicate over wireless links characterized by lower bandwidths, higher transmission error rates, and more frequent disconnections in comparison to wired networks. Since most of the video compression standards and network protocols were designed for wired visual communications, they would not be effective if straightforwardly applied to the wireless case. In order to accommodate an evergrowing number of mobile users, the recently published distributed video coding containes several new functionalities for improved coding efficiency, increased error resilience, which were needed for wireless communications in error-prone environments [24].

1.1.2 Characteristics of Mobile Terminals and Wireless Channel

The prevalence of multimedia technology promotes digital images and videos to play a significant role than the traditional text data. Many digital applications, such as pay-TV, confidential video conferencing, medical and miliary imaging systems, require a reliable security in storage and transmission of digital multimedia data including images and videos. In recent years, many potable devices, such as mobile phone and Personal Digital Assistant (PDA), provide additional functions of savings

and exchanging multimedia messages. Such terminals increasingly include low to moderate resolution color displays, cameras, and moderate native processing power. However, battery size and battery life is always a concern for these mobile devices. Wireless systems are limited by wireless bandwidth and mobile terminal resources. Wireless bandwidth is scare because of its shared nature and the fundamental limitations of wireless spectrum. Mobile terminal resources are often practically limited by power constraints and by display, communication, and computational capabilities. In order to make the most efficient use of wireless bandwidth and mobile resources, it is desirable to send mobile terminals the lowest bandwidth video streams that match their display and communication capabilities [25].

Delivery of real-time video typically has quality of service (QoS) requirements, e.g., bandwidth, delay and error requirements. First, video transmission usually has minimum bandwidth requirements to achieve acceptable presentation quality. Second, real-time video has strict delay constraints. This is because real-time video must be played out continuously. If the video packet does not arrive in a timely manner, the playout process will pause, which is annoying to users. Third, video applications typically impose upper limits on bit error rate (BER) since too many bit errors would seriously degrade the video presentation quality. However, unreliability and bandwidth fluctuations of wireless channels can cause severe degradation to video quality. Furthermore, for video multicast, heterogeneity of receivers makes it difficult to achieve efficiency and flexibility. We discuss these issues in detail as follows [26].

Unreliability: Compared with wired links, wireless channels are typically much more noisy and have both small-scale (multipath) and large-scale (shadowing) fades [27], making the BER very high. The resulting bit errors can have a devastating effect on video presentation quality [28]. Therefore, it is crucial to develop robust transport mechanisms for video over wireless channels.

Bandwidth Fluctuations: The bandwidth fluctuates for several reasons. First, when a mobile terminal moves between different networks [e.g., from a wireless local area network (LAN) to a wireless wide area network (WAN)], the available bandwidth

may vary drastically (e.g., from a few megabits per second to a few kilobits per second). Second, when a handoff happens, a base station may not have enough unused radio resource to meet the demand of a newly joined mobile host. Third, the throughput of a wireless channel may be reduced due to multipath fading, co-channel interference, and noise disturbances. Last but not least, the capacity of a wireless channel may fluctuate with the changing distance between the base station and the mobile host. Consequently, bandwidth fluctuations pose a serious problem for real-time video transmission over wireless networks.

1.1.3 Problems

To fulfill security and privacy needs in various applications, encryption of images and videos is very important to prevent malicious attacks from unauthorized attackers. The security of multimedia data is provided by encryption in common. That is, the encryption process transforms a plaintext message into a ciphertext message, which is a unintelligible message. The traditional way to secure multimedia applications is encrypt multimedia data using secret key cryptography algorithms such as Data Encryption Standard (DES) [10, 11] or International Data Encryption Algorithm (IDEA) [29, 30] as shown in Figure 1.1. These algorithms require complicated computations because these encryption schemes treat digital images and videos as bit stream and encrypt them bit by bit.

1.1.4 Solutions for Security in Mobile Devices

Digital image/video processing techniques are used to enhance the quality of image and to reduce the data needed to represent an image without visually affecting the quality of image (compression). Cryptographic techniques are used to scramble images so that an adversary could not obtain the original image without knowing the secret key. Vast amount of data of these multimedia applications put great burden on the encoding process. Encryption or decryption will aggravate this problem. How

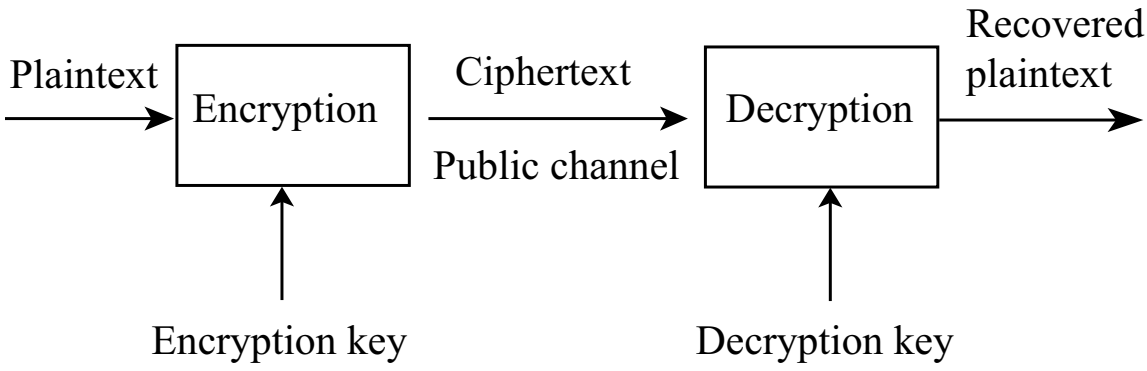


Fig. 1.1. Encryption and Decryption of a Cipher.

to encrypt and decrypt the vast amount of data efficiently is an important issue. Because of the high information rate of the digital multimedia data, the computational overhead introduced by the encryption procedure and decryption procedure should be as low as possible.

Selective encryption can be used to reduce the power consumed by the encryption function for digital content when the content is protected by a digital rights management systems. Selective encryption is a technique to save computational complexity or enable interesting new system functionality by only encrypting a portion of a compressed bitstream while still achieve security. For selective encryption to work, we need to rely not only on the beneficial effects of redundancy reduction described by Shannon [31], but also on a characteristics of many compression algorithms to concentrate important data about reconstruction in a relatively small fraction of the compressed bitstream [32]. These important elements of compressed data become candidates for selective encryption.

1.2 Multicast Security

Multicasting is becoming more important in the development of network applications for providing efficient delivery of data from a source to multiple recipients.

As part of the new issues involved with multicast communications, multicast security and scalability have received particular attention due to the various vulnerabilities found in their application [33–39].

A lot of applications, such as video conferencing, video-on-demand, stock-quote distribution, and software update, have been developed for streaming digital multimedia contents to a set of clients. In such applications, the multicast protocol plays an important role because it can efficiently deliver data from a source to multiple receivers. It reduces the bandwidth of the wireless networks and the computational overhead of mobile devices. This makes multicast an ideal technology for communication among a large group of users because wireless channels are very limited and precious resources. An important issue is how to provide security to these applications. Security could involve a number of issues, like authentication of clients, secure data transmission and copyright protection. For each of these security needs, a number of security protocols (especially for multicast) have been developed and a great deal of research continues in this area. The problem then is how to flexibly integrate security protocols into multimedia streaming applications even though these applications are usually developed without security.

1.2.1 Basic Multicast Steps

The process of secure multicast is composed of two basic steps: key distribution and transmission of encrypted data. Once a group key has been securely established among the members of the multicast group, it can be used with any fast symmetric encryption algorithm to encrypt the data to be transmitted. Therefore, the challenge in developing secure multicast protocols is primarily in designing efficient schemes for the key distribution. Let us indicate the conditions that require establishment of a new group key during a secure multicast session:

- When the multicast group is formed - This is the first time that the group key is established at the beginning of the session.

- When a member of the group leaves or is expelled - Rekeying is required to prevent the member from using its key to decrypt future communication. This is called Forward Rekeying.

- When a new member joins the group - Rekeying is required if the new member is to be prevented from decrypting earlier communication (which it could have stored). This is called Backward Rekeying.

- When a timeout occurs - Keys are usually associated with a timeout after which they become potentially insecure. The length of this timeout depends on many factors like key length, encryption algorithm used, wired or wireless network etc. If such a timeout exists and occurs, rekeying is required. This is referred to as Periodic Rekeying.

We thus see that a key distribution can take place quite often during a multicast session. It is very important to optimize this process. Most of the secure multicast protocols, therefore, differ from each other only in the key distribution scheme.

Many secure multicasting protocols have been proposed in the past few years. Existing key distribution schemes can be classified into non-scalable and scalable protocols. Some of this discussion is drawn from [40–42].

1.2.2 Key Management Role

Key management plays an important role enforcing access control on the group key and consequently on the group communication. It supports the establishment and maintenance of key relationships between valid groups according to a security policy being enforced on the group. It encompasses techniques and procedures that can carry out [40]:

- Providing member identification and authentication. Authentication is important in order to prevent an intruder from impersonating a legitimate group

member. In addition, it is important to prevent attackers from impersonating key managers. Thus, authentication mechanisms must be used to allow an entity to verify whether another entity is really what it claims to be

- Access control. After a group has been identified, its join operation should be validated. Access control is performed in order to validate group members before giving them access to group communication, in particular the group key.

- Generation, distribution and installation of key material. It is necessary to change the key at regular intervals to safeguard its secrecy. Additional care must be taken when choosing a new key to guarantee key independence. Each key must be completely independent from any previous used and future keys, otherwise compromised keys may reveal other keys.

1.2.3 The Problem - Mobility

In wireless networks, secure multicast protocols are more difficult to implement efficiently due to the dynamic nature of the multicast group and the scarcity of bandwidth at the receiving and transmitting ends. Mobility is one of the most distinct features to be considered in wireless networks. Moving users onto the tree causes extra key management resources even though they are still in service. To take care of frequent handoff between access networks, it is necessary to reduce the number of rekeying messages and the size of the messages. The multicast protocol used in wired networks does not perform well in wireless networks because multicast structures are fragile as the mobile node moves and connectivity changes. When we choose a key management scheme, the structure of the wireless network should be considered very carefully. For example, the wireless cellular network has a unique hierarchy structure such that a key management scheme should be easy to deploy. Some papers already address the access control schemes in wireless networks. In [43], they propose the topology matching key management trees (TKMK) and test in respect to the communication cost. By matching the key tree to the network topology, the commu-

nication traffic is reduced by 33% $\sim$ 45% compared to the conventional key trees that are independent of the network. In [44], they propose baseline, immediate, delayed and periodic rekeying schemes and test them in the wireless Local Area Network (LAN) network. To our best knowledge, this is the first paper that computes the handoff impact of centralized key management scheme in the real wireless cellular networks.

1.2.4 Solutions for Multicast Security

Multicast communication is becoming the basis for a growing number of applications. It is therefore critical to provide sound security mechanisms for multicast communication in wireless networks. Yet, existing security protocols for multicast offer only partial solutions. We first present a multicast scenario on the wireless cellular systems and point out relevant security concerns.

Several protocols exist for the efficient key distribution during secure multicast. The main aims of these protocols include network architecture independence, robustness and scalability. In this dissertation, we design a key management tree such that neighbors on the key tree are also physical neighbors on the cellular networks. By tracking the user location, we localize the delivery of rekeying messages to the users who need them. This lessens the amount of traffic in wireless and wired intervals of network. The group key management scheme uses the pre-positioned secret sharing scheme.

1.3 Contribution of this Dissertation

The primary objectives of this research are to obtain a deeper understanding of video data security in wireless applications. The main contribution of this thesis, as described in later chapters are:

- **Fast and secure algorithm for DSC bitstreams**

Cryptographic techniques are used to scramble images so that an adversary could not obtain the original image without knowing the secret key. Vast amount of data of these multimedia applications put great burden on the encoding process. Encryption or decryption will aggravate this problem. How to encrypt and decrypt the vast amount of data efficiently is an important issue. Because of the high information rate of the digital multimedia data, the computational overhead introduced by the encryption procedure and decryption procedure should be as low as possible.

Our selective encryption algorithm reduce the power consumed by the encryption function for digital content when the content is protected by a digital rights management systems. Also our encryption algorithm save computational complexity or enable interesting new system functionality by only encrypting a portion of a compressed bitstream while still achieve security.

A framework for implementing selective video encryption for a LDPC and Turbo codes based distributed source coding method is proposed. Secrecy results from a tradeoff between processing power and speed, but real-time processing is achievable. We showed that the encryption of 50% or more bits reveals no useful information in the reconstructed video. Hence the proposed method has some advantages over conventional full data encryption with regard to complexity. We are investigating how the compressed bit stream can be exploited by imposing a syntax on the output of the DSC coder.

- **New secure group key management scheme**

  We proposed a new efficient key distribution algorithm for the secure multicast. The main aims of this protocol include network architecture independence, robustness and scalability. In this dissertation, a key management tree is designed such that neighbors on the key tree are also physical neighbors on the cellular networks. The group key management scheme uses the pre-positioned secret sharing scheme. By localizing the user location, we lessens the amount of traffic

in wireless and wired intervals of network. It is known that LKH approaches the theoretical limit.

We find that each call undergoes an average of $3 \sim 8$ handoffs during a call duration according to the user mobility model. We proposed a new handoff scheme to minimize the key updating transactions. This new handoff scheme reduces one of the two key update transactions in the handoff region - adding a new channel when a call enters the handoff region. In the handoff area, only a new traffic channel is added to minimize the interruption time of the data transmission. With a the revised handoff scheme, the number of handoffs per call is reduced by almost 20% compared to that of the soft handoff. Also a simple CAC function is used to maintain key updating transactions to a level defined by the system manager.

Our new algorithm has almost same computational complexity with LKH and has some advantages for the frequent handoff.

1.4 Overview of the Dissertation

Chapter 2 is a background of distributed video coding. The background includes an overview of the fundamentals of distributed source coding, duality between source coding and channel coding with side information, and practical designs and applications of distributed video coding.

Chapter 3 provide a survey and classification of the conventional video encryption schemes and discuss the current issues. This overview include naive, pure permutation, Zig-Zag permutation, video encryption algorithm (VEA), selective video encryption algorithms, and JPEG-2000 security standard (JPSEC). Then we propose our selective bitstream encryption method.

Chapter 4 show the use of selective encryption on a compressed bit stream that has been encoded using distributed source coding. Low-density parity-check (LDPC) and Turbo codes based distributed source coding are used to test the effectiveness of

the selective encryption on a compressed bit stream. Experimental results show that our method is robust against replacement attacks and direct recovery.

Chapter 5 a background of the group key management method for multicast protocols. This chapter include an overview of the previous work on the group key management schemes.

Chapter 6 investigate mobility impact of a new group key management scheme. We designed a group key management tree such that the neighbors on the key tree are also physical neighbors on the cellular network. The group key management scheme uses the pre-positioned secret sharing scheme. By tracking the user location, we localized the delivery of rekeying messages to the nodes that need them. This lessens the amount of traffic in the cellular network.

Conclusions are in Chapter 7 with future work.

2. DISTRIBUTED SOURCE CODING

Distributed source coding (DSC) is a new paradigm for video compression, based on Wyner-Ziv [45] and Slepian-Wolf [46] information theoretic results. DSC refers to separate encoding of correlated sources with joint decoding. DSC exploits the source statistics in the decoder and, hence, the encoder can be very simple, at the expense of a more complex decoder. The traditional balance of complex encoder and simple decoder is essentially reversed. Such algorithms hold great promise for new generations of mobile video cameras [47].

Although the theoretical result given by the Slepian-Wolf theorem has been known for more than 30 years, practical approaches to DSC did not appear until 1999. Since then, DSC has become a very active area of research. This chapter reviews the recent development of practical distributed source coding schemes.

2.1 Foundations of Distributed Source Coding

In this section, we will show the foundations of distributed coding and compression techniques that exploit receiver side information.

2.1.1 Slepian-Wolf Theorem for Lossless Distributed Coding

The Slepian-Wolf theorem [46], which forms the basis of lossless distributed source coding (DSC) problem, defines the achievable rate region when two physically separated and statistically correlated sources are independently encoded and jointly decoded for a lossless channel. Distributed compression refers to the coding of two (or more) dependent random sequences, but with the special twist that a separate encoder is used for each. Each encoder sends a separate bit stream to a single decoder

which may operate jointly on all incoming bit streams and thus exploit the statistical dependencies.

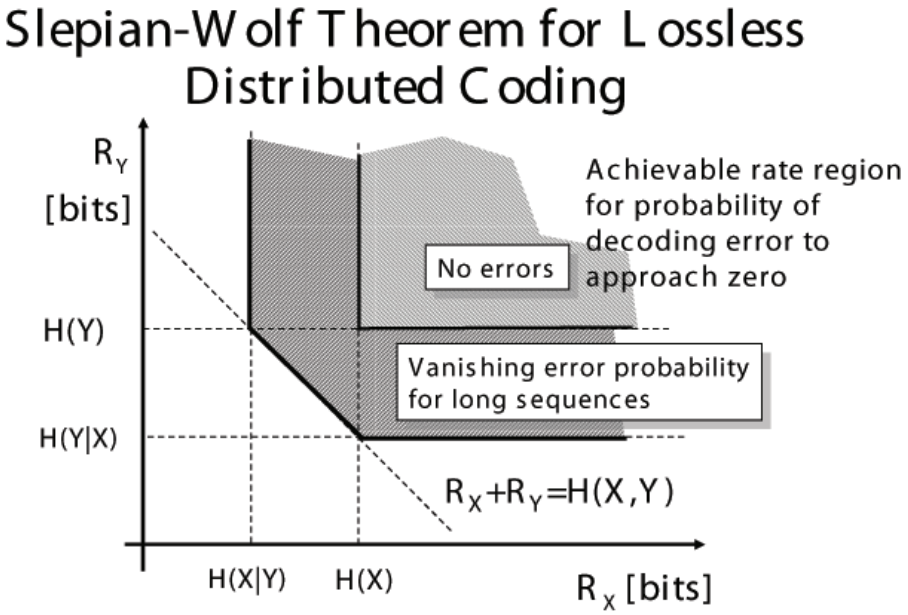


Fig. 2.1. Slepian-Wolf theorem.

Consider two statistically independent identically distributed (i.i.d.) finite-alphabet random sequences **X** and **Y** (in Figure 2.1). With separate conventional entropy encoders and decoders, one can achieve $R_X \geq H(X)$, $R_Y \geq H(Y)$, where $H(X)$ and $H(Y)$ are the entropies of **X** and **Y**, respectively. However if two discrete signals, X and Y, are compressed using two independent encoders but are decoded by a joint decoder, then Slepian-Wolf theorem on distributed source coding states that even if the encoders are independent, the achievable rate region for probability of decoding error to approach zero is $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$, and $R_x + R_y \geq H(X,Y)$.

It has been shown that the Slepian-Wolf boundary is achievable both asymptotically and with finite-length sequences [48, 49]. Specifically, the corner points of the Slepian-Wolf boundary, where one source is losslessly available at the decoder (e.g. Y compressed to $H(Y)$ via a conventional entropy-compression method) and the other is maximally compressed utilizing the statistical correlation between the two sources (X compressed to $H(X|Y)$), may be modeled as an equivalent channel coding problem with decoder side information (SI) where the equivalent transmission channel is specified by the correlation of the sources. To get close to the theoretical limit, two key issues need to be resolved: (i) finding a capacity-approaching channel code for the equivalent transmission channel and (ii) bridging the practice and solution of channel coding with that of source coding. Although closely related, the two issues reflect different aspects of the DSC problem. While the former can take advantage of the rich literature available on channel coding, the latter is much less studied. One of the most successful approaches that bring the solution of channel coding to serve the problem of source coding is the coset/syndrome/binning approach [46, 48, 49].

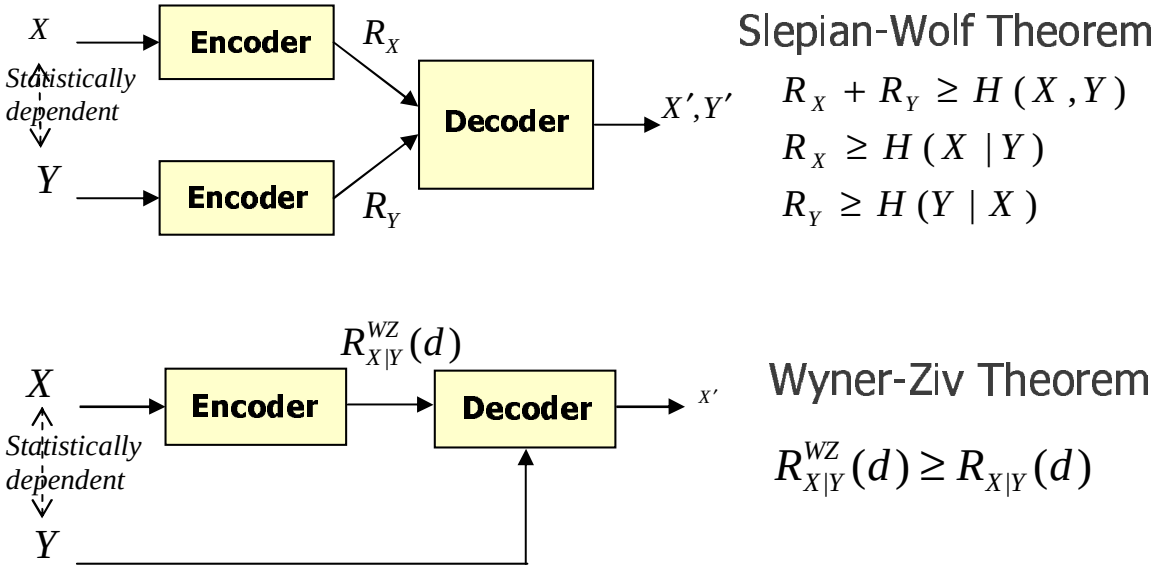


Fig. 2.2. Distributed Source Coding.

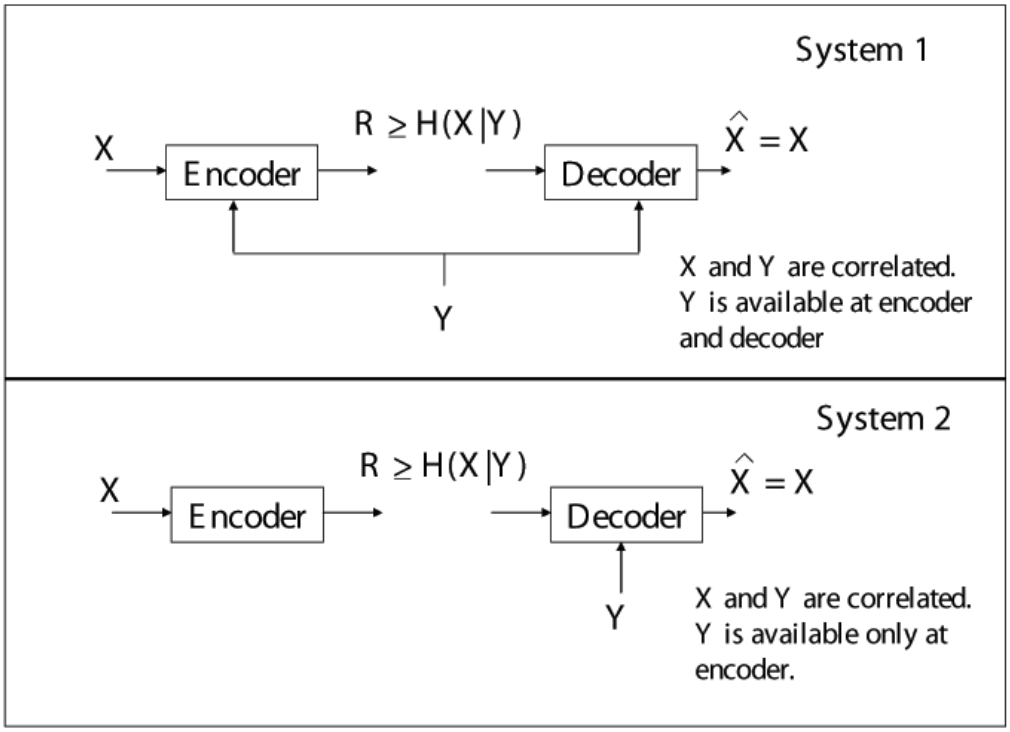


Fig. 2.3. Wyner-Ziv coding with side information at the decoder.

2.1.2 Wyner-Ziv Theorem for Lossy Distributed Coding

The counterpart of this theorem for lossy source coding is Wyner and Ziv's work on source coding with side information [45]. The Wyner-Ziv Theorem deals with the problem of source coding with side-information. The encoder needs to compress a source X when the decoder has access to a source Y. X and Y are correlated sources and Y is available only at the decoder. It is known from information theory that for the case when the mean squared error is the distortion measure and $X = Y + N$ where N has a Gaussian distribution, the rate-distortion performance for coding X is the same whether or not the encoder has access to Y. Let X and Y be statistically dependent Gaussian random processes, and let Y be known as side information for

encoding X. For the problem of source coding with side information, the encoder needs to encode the source within a distortion constraint, while the decoder needs to be able to decode the encoded codeword subject to the correlation noise N (between the source and the side-information). While, the results proven by Wyner and Ziv are non-constructive and asymptotic in nature, a number of constructive methods to solve this problem have since been proposed [48–56] wherein the source codebook is partitioned into cosets of a channel code that is matched to the correlation noise N.

Wyner and Ziv showed that the conditional Rate-Mean Squared Error Distortion function for X is the same whether the side information Y is available only at the decoder, or both at the encoder and the decoder. We refer to lossless distributed source coding as Slepian-Wolf coding and lossy source coding with side information at the decoder as Wyner-Ziv coding [49, 50].

2.1.3 Duality Between Source Coding and Channel Coding with Side Information

Information-theoretic duality between source coding with side information (SCSI) at the decoder and channel coding with side information (CCSI) at the encoder is discussed in [57]. The second scenario was first studied by Costa in the "dirty paper" problem [58] and exploited again recently because of its expansive applications in data hiding, watermarking and multi-antenna communications [59].

Figure 2.4(a) is identical to the scenario shown in Figure 2.2 with the switch off, while here we apply the notations corresponding to [57]. The correspondence between the variables in Figure 2.4 is listed in Table 2.1.

The duality can be summarized in the following aspects [57]:

- For SCSI, $p(x|s)$ and $p(s)$ are fixed, the goal is to minimize the convex rate-distortion function:

$$R^*(d) = \min_{p(u|x),p(\hat{x}|u,s)} [I(U;X) - I(U;S)] \tag{2.1}$$

  where $I(\cdot;\cdot)$ represents the mutual information.

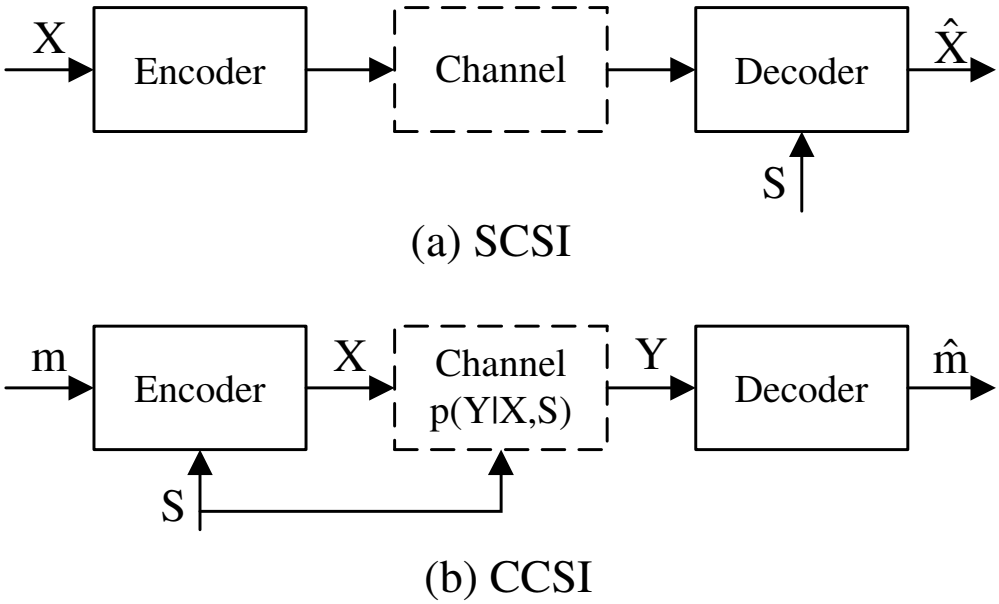


Fig. 2.4. The duality between source coding and channel coding with side information (a) Source coding with side information at the decoder (SCSI); (b) Channel coding with side information (CCSI)

- For CCSI, $p(x|\hat{x}, s)$ and $p(s)$ are fixed, the goal is to maximize the capacity function of the concave system with a cost constraint W:

$$C^*(W) = \max_{p(u|s),p(x|u,s)} [I(U;Y) - I(U;S)] \tag{2.2}$$

- The mappings of the encoder of SCSI and the decoder of CCSI are in the identical domain and range: $\mathcal{X} \to \{1, 2, \cdots, 2^{RL}\}$, where L denotes the block length of the operations. Correspondingly, the mappings of the decoder of SCSI and the encoder of CCSI are: $\{1, 2, \cdots, 2^{RL}\} \times \mathcal{S}^L \to \hat{\mathcal{X}}^L$.

- When the block length of the data sequences approximates infinity, the duals are functionally identical, i.e., if we could find the solution of one problem, the other is solved as well.

Table 2.1
Correspondences of variables between SCSI and CCSI

SCSI	**CCSI**
source input - X	Y - channel output
side information - S	S - side information
auxiliary variable - U	U - auxiliary variable
source reconstruction - $\hat{X}$	X - channel input

2.2 Practical Designs and Applications of Distributed Video Coding

The basis for DSC were set some thirty years ago by the work from Slepian-Wolf for the lossless case and Wyner-Ziv for the lossy case. Wyner-Ziv coding refers to lossy source coding with side information at the decoder. Recently some practical applications of Wyner-Ziv coding to video compression have been studied due to its advantage of error robustness over standard video coding standards [60].

2.2.1 Slepian-Wolf Coding

While a unified theory of network information flow still remains elusive, recent developments in DSC algorithms are truly causes for excitement. For example, practical approaches to DSC started appearing in the literature in 1999. In particular, the first practical solution to the Slepian-Wolf problem for binary memoryless sources using advanced channel codes such as low-density parity-check (LDPC) codes [61–65] is proposed, which achieving performance results very close to the theoretical limit. For the more general Wyner-Ziv coding problem, a Slepian-Wolf coded nested quantization paradigm that performs only 0.2 dB away from the Wyner-Ziv rate-distortion function with trellis-coded quantization and ideal Slepian-Wolf coding is offered [66]. Thus we are approaching the theoretical performance limit of DSC and the stage is set for practical applications of DSC to sensor networks and related areas (e.g., error-robust video coding).

2.2.2 Wyner-Ziv Coding

In the implementation of a practical Wyner-Ziv video coding scheme, different coding strategies are proposed: Aaron's "intraframe encoding+interframe decoding system" [53, 67–73], Puri and Ramchandran's PRISM system [50, 51], layered Wyner-Ziv coding scheme by Xu and Xiong [74], and Sehgal's statefree causal video encoding system [75]. The performance of a Wyner-Ziv video codec is greatly dependent on the

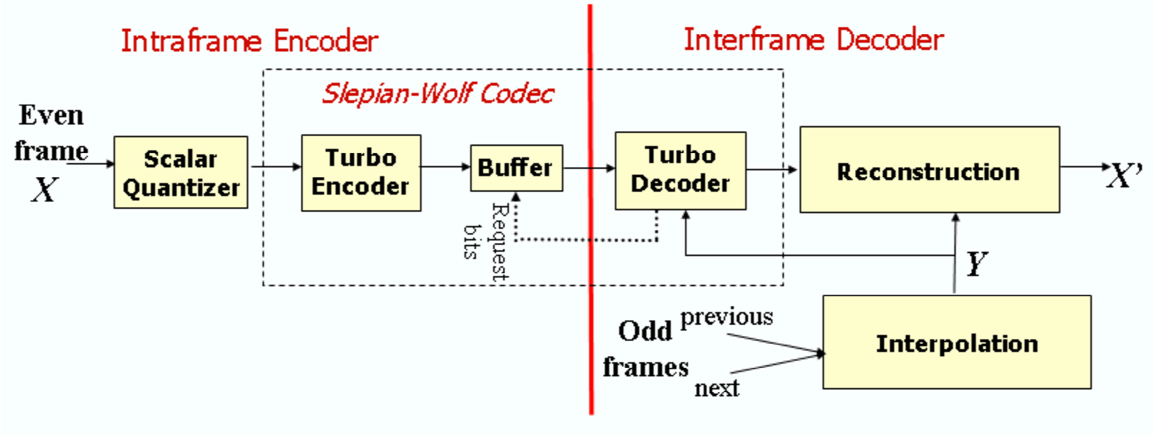


Fig. 2.5. An Example of Wyner-Zip Coding.

quality of reconstructed side information, that is, an approximation of current frame. Unlike conventional motion compensated video coding schemes, current frame is not available at the decoder. One has to rely on previously decoded intra frames to either interpolate or extrapolate the side information. Extrapolation from decoded frame leads to a low quality of approximation, particularly when motion among interframes is large. Most of the existing Wyner-Ziv schemes adopt a frame interpolation strategy, i.e., intra frames called key frames are inserted periodically. Side information for a frame can be formed by using two neighboring intra frames and motion compensated interpolation algorithm. This is similar to the "B" frame used in MPEG and H.26x video coding schemes. This method can be used for applications such as video downloading etc. However, it is not suitable for real-time, low delay applications because an extra waiting time for the next intra frame increases the delay in the whole system. For real-time application the coding structure of "I-P-P-P" has to be employed. To adopt this coding structure in Wyner-Ziv video coding for real-time applications, we can either have a joint decoder with motion compensation using the Wyner-Ziv bits, or send additional helper information from the encoder [73]. The problem with such helper information is that additional information (e.g., hash bits) consumes extra bits

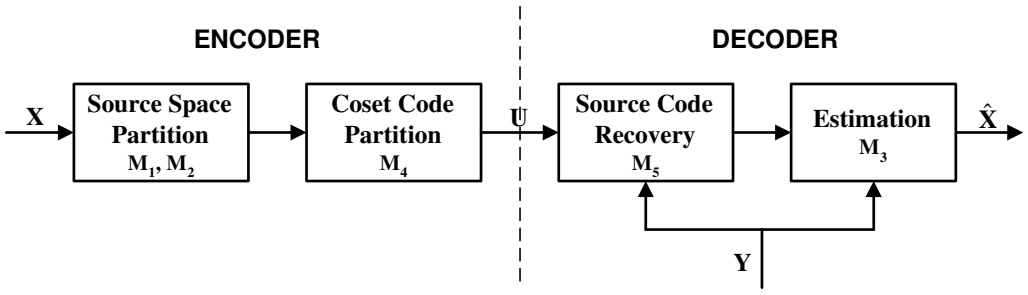


Fig. 2.6. Design algorithm of distributed source coding using syndromes (DISCUS)

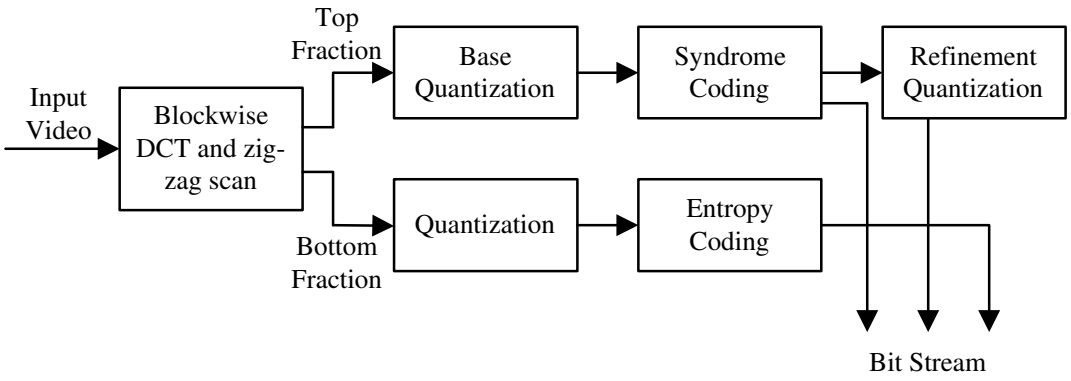


Fig. 2.7. Block diagram of the encoder using PRISM

to transmit. In general, the Wyner-Ziv codec consists of an inner channel codec and an outer quantization-reconstruction pair as shown in Figure 2.5.

2.2.3 Low-Complexity Distributed Source Encoding in Wireless Networks

Distributed Source Coding (DSC) allows the complexity of the encoders to be reduced, so power or complexity limited devices like mobile phones or surveillance cameras can transmit multimedia data such as video and streaming. The combination of such a Simple-Encoder-Complex-Decoder system with existing Complex-Encoder-

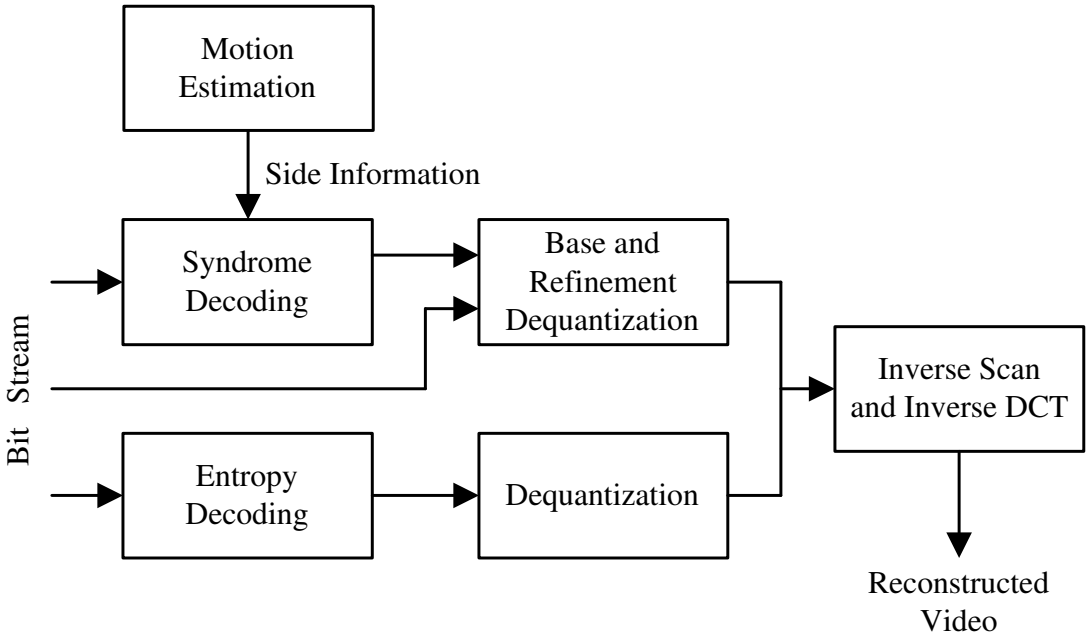


Fig. 2.8. Block diagram of the decoder using PRISM

Simple-Decoder systems should finally allow mobile-to-mobile videoconferencing (by using a transcoder located on the access network).

To achieve low-complexity encoding, an asymmetric video compression scheme is proposed where individual frames are encoded independently (intraframe encoding) but decoded conditionally (interframe decoding) [73]. Two results from information theory suggest that an intraframe encoder - interframe decoder system can come close to the efficiency of an interframe encoder-decoder system.

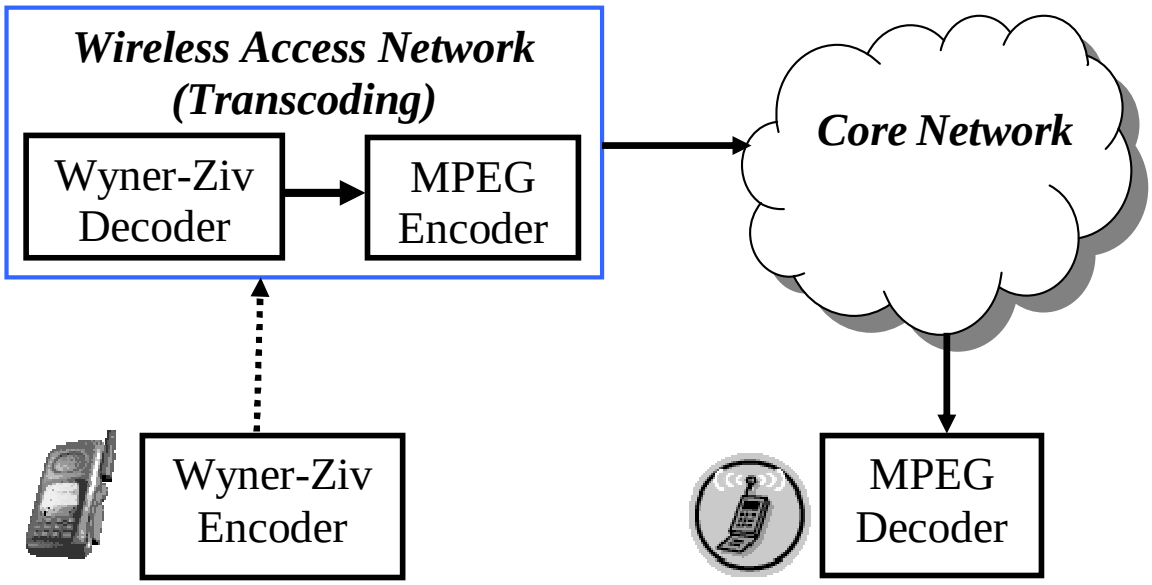


Fig. 2.9. An Application of Distributed Video Coding.

A Wyner-Ziv video coder would have a great cost advantage, since it compresses each video frame by itself, requiring only intraframe processing. The corresponding decoder in the fixed part of the network would exploit the statistical dependence between frames, by much more complex interframe processing. Beyond shifting the expensive motion estimation and compensation from the encoder to the decoder, the desired asymmetry is also consistent with the Slepian-Wolf and Wyner-Ziv coding algorithms, which tend to have simple encoders, but much more demanding decoders. Even if the receiver is another complexity-constrained device, as would be the case

for video messaging or video telephony with mobile terminals at both ends, it is still advantageous to employ Wyner-Ziv coding in conjunction with a transcoding architecture depicted in Figure 2.9. A mobile cameraphone captures and compresses the video using Wyner-Ziv coding and transmits the data to the fixed part of the network. There, the bitstream is decoded and re-encoded using conventional video standards, such as MPEG. This architecture not only pushes the bulk of the computation into the fixed part of the network, but also shares the transcoder among many users, thus providing additional cost savings.

3. VIDEO ENCRYPTION SCHEMES

In this chapter, we provide a survey and classification of the proposed video encryption schemes and discuss the current issues. Then we present our selective encryption scheme.

3.1 Naïve Algorithm

The most straight-forward method is to encrypt the entire Moving Picture Experts Group (MPEG) [76] stream using standard encryption methods such as DES and Advanced Encryption Standard (AES) [9, 77]. This is called the Naïve algorithm approach [78]. Naïve algorithm treats the MPEG bit-stream as the traditional text data and does not use any of the special structure.

This encrypts entire MPEG stream by secure encryption algorithm. The Naïve algorithm is arguably the most secure MPEG encryption algorithm because there is no effective algorithm to break DES, double DES or triple DES so far. However, It is slow especially when we use double or triple DES to achieve the top security level. The size of the encrypted stream does not change because most standard encryption algorithms preserve the size.

3.2 Pure Permutation Algorithm

Based on the statistical results, the byte stream of MPEG video show a very low frequency of diagrams (pair of bytes). Therefore, the usual cryptanalysis of using data frequency, diagram frequency, etc. is useless or at least very hard. This motivates the Pure Permutation Algorithm [79], which simply scrambles the byte stream by permutation. The cardinality of the permutation key can be varied and depends

on the security level and the application requirement. For example, we can use a permutation of 64 numbers or we can use a long permutation list with 1/8 of an I frame.

The pure permutation algorithm is vulnerable to the known-plaintext attack. Once the permutation list is figured out by comparing the ciphertext with the known original frame, all frame can be easily decrypted. In order to find out the permutation list, known-plaintext data with multiple length of the permutation list are needed. This is because some numbers may appear multiple times in the permutation list. Suppose a block of 64 bytes data is $(0, 0, 0, 14, ..., 5)$. After permutation, it changes to $(14, 0, 5, 0, ..., 0)$. Then there are three possible positions for 0's. With two more blocks (128 bytes) of known-plaintext data, it is quite easy to decide uniquely the positions of these 0's because we can anticipate different byte values at the same position based on the statistical analysis. The amount of needed known-plaintext data also depends on the length of the permutation list. However, notice that the unicity of MPEG data stream is about the same magnitude of one I frame, knowing one I frame is enough to decrypt the permutation list based on Shannon's Theorem.

3.3 Zig-Zag Permutation Algorithm

The basic idea of Zig-Zag Algorithm [80] is that, instead of mapping the 8x8 block to a 1×64 vector in "Zig-Zag" order, it uses a random permutation list (secret key) to map the individual 8x8 block to a 1×64 vector.

Following experiments are conducted by [80].

- DC coefficient is mapped to the first element in the 1x64 vector and the rest of the elements are permuted. Obscured image.
- DC coefficient of every bock is set to zero or a fixed value between 0 and 255 and rest of the elements are permuted. Obscured image.

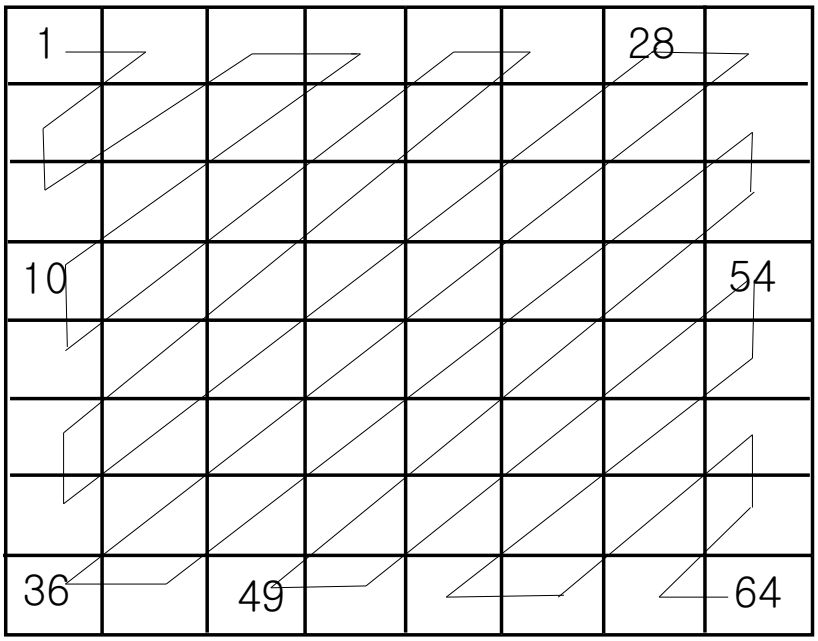


Fig. 3.1. Zig-Zag Algorithm of MPEG.

- DC coefficient is mapped to any other position other than the first position in the 1×64 vector, and the rest of the elements are randomly permuted. Incomprehensible image.
- AC_{63} coefficient is set to 0. Degradation is negligible.
- Split the DC coefficient into two parts, first part remain in the same position, the second part is substituted for AC_{63} and randomly permute the list. Incomprehensible image.

Since mapping Zig-Zag order and mapping according to the random permutation list have the same computational complexity, the encryption and decryption add very little overhead to the video compression and decompression processes. However

this method decrease the video compression rate because the random permutation distort the probability distribution of Discrete Cosine Transform (DCT) coefficients and make the Huffman table used less than optimal.

Also the basic Zig-Zag Permutation Algorithm is vulnerable to the ciphertext only attack. The attack is based on the fact that none-zero AC coefficients are gathered in the upper-left corner of the I block. Statistical analysis which count the number of non-zero DC and AC coefficients from all blocks in an I frame was conducted by [79]. The results show that:

- DC coefficients always have the highest frequency of non-zero occurrence.
- The frequency of AC_1 and AC_2 are among the top 6.
- The frequency of AC_3 to AC_5 are among top 10.

The second problem is more serious. The Zig-Zag permutation algorithm can not withstand the known-plaintext attack. Many video starts with standard header clips. Thus an adversary can easily figure out the key by simply comparing the known header clips with the encrypted header ones.

To solve this problem, author propose a method so-called binary coin flipping sequence together with two different permutation lists. For each 8×8 block, a coin is flipped. If it is a tail, the permutation list 1 (key1) is applied to the block; if it is a head, the permutation list 2 (key2) is applied to the block. Key1 and Key2 are the secret keys. This method is subject to known plain text attack. The idea is to select the key that has the tendency to gather AC coefficients in the upper left corner. However this scheme is also eligible for ciphertext only and known-plaintext attacks.

3.4 Video Encryption Algorithm

The Video Encryption Algorithm (VEA) developed by Qiao and Nahrstedt in [81] also fully encode video stream. However, VEA depends upon statistical properties of MPEG and uses a standard symmetric algorithm to reduce the amount of data that

is actually encrypted. This can yield an almost 50% gain in performance over the Naïve algorithm.

The chunk of I-frame, $(a_1, a_2, a_3, a_4, \cdots, a_{2n-1}, a_{2n})$, is divided into two data segments of odd and even list: $(a_1, a_3, a_5, \cdots, a_{2n-1})$ and $(a_2, a_4, a_6, \cdots, a_{2n})$. The encryption key consists of randomly generated 0 or 1 bit sequences with the equal number of 0's and 1's. The substream are then xored to obtain the ciphertext $(c_1, c_2, c_3, \cdots, c_n)$ which is concatenated to $E(a_2, a_4, a_6, \cdots, a_{2n})$, where E denotes an encryption function. If $(a_2, a_4, a_6, \cdots, a_{2n})$ is not a repeated pattern, the secrecy of this algorithm depends on the secrecy of the encryption function E because the even lists $(a_2, a_4, a_6, \cdots, a_{2n})$ can be considered an one-time pad. Thus we can say that this scheme is immune to known-plaintext attack. Also the key will be changed for each frame. The best way to perform ciphertext only attack is to use frequency analysis. The attack is based on the finding a pair (a, b) such that $a \bigoplus b = c$. From the statistical analysis since the highest pair frequency is 10^4 the chance of guessing these pairs is very remote and hence the ciphertext only attack is very difficult.

3.5 Selective Encryption Algorithm

Mobile multimedia applications, the focus of many forthcoming wireless services, increasingly demand low-power techniques implementing content protection and customer privacy. The encryption and decryption of video at rates of $30 - 60$ Mbps is not possible with the currently available encryption algorithms, which are originally developed for text data. In anticipation of the great need for a security mechanism for real-time video applications emerging with the implementation of high bandwidth networks, a new security mechanism is proposed. They use the features of MPEG layered structures. These algorithms all fall into the category of Selective Algorithm [82]. In order to efficiently utilize available bandwidth and storage capacity, it is expected that digitized video will be compressed using a standard compression algorithm.

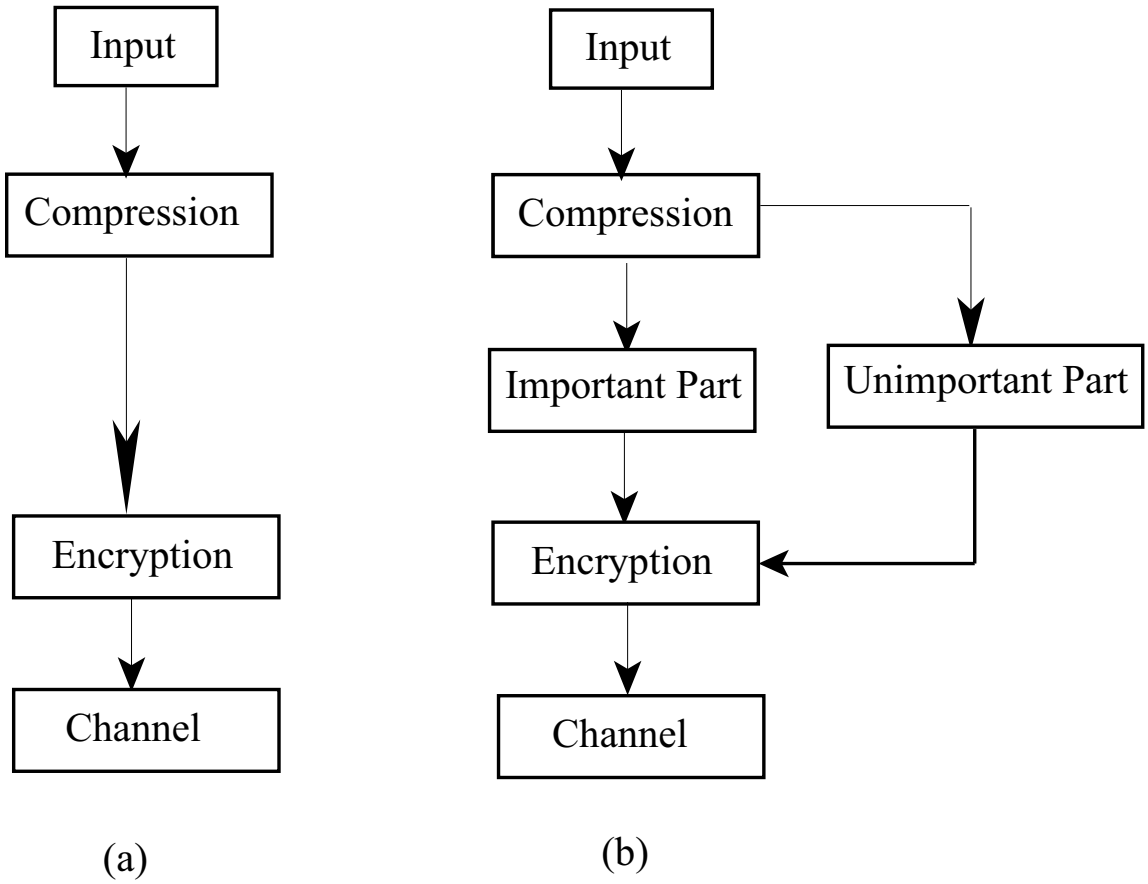


Fig. 3.2. Comparison of (a)the traditional approach to secure image and video communication and (b) the selective approach.

Partial encryption of the MPEG video stream provides some level of security because it presents two challenges to the ambitious network intruder. When portions of the MPEG stream are encrypted, the MPEG stream does not confirm to the standard MPEG stream layered structure, and consequently, it is impossible to identify frames, group of frames, or the encrypted I frames. The network intruder must first separate the encrypted portions of the stream, and then still faces the complexity of the encryption algorithm.

3.5.1 AEGIS: I Frame only

AEGIS [78] exploits the great sensitivity of compressed video. The approach used by Aegis is the encryption of I frames for all MPEG groups of frames in a MPEG video stream. The choice of encrypting I frames is based on the great significance of the intraframe in the decompression of a MPEG stream. B and P frames represent only translations of the picture information found in adjacent I frames; therefore, the encryption of I frames renders them useless. Further more, the intentional corruption of the stream has a serious impact on the outputs of the inverse DCT function during decoding. The recovery from such corruption is practically impossible. In addition to the encryption of I frames, Aegis also encrypts the MPEG video sequence header. The sequence header contains all of the decoding initialization parameters such as the picture width, height, frame rate, bit rate and buffer size. The encryption of the sequence header, also conceals the MPEG identity of the stream and makes the MPEG video stream unrecognizable. In order to further conceal the MPEG identity of the stream, the ISO end code (last 32 bits of MPEG) is also encrypted. Aegis uses DES for the encryption process. See appendix for empirical results of Aegis security scheme.

Agi and Gong [78] have shown that great portions of the video are visible partly because of inter-frame correlation and mainly from unencrypted I blocks in the P and B frames. Therefore, encrypting only I frames does not provide a satisfactory security level. In order to provide better security level, in addition to encrypting I frames, all I blocks in P and B frames must also be encrypted. Identifying the I-blocks will introduce overhead. Encrypting only I frames can save $30 - 50$ of encryption/decryption time. Size does not increase. Agi and Gong have also suggested to increase the frequency of I frames to enhance the security. It has the main drawback of increasing the length of string and consequentially the encryption time.

3.5.2 Sign-Bit of DCT Coefficients

Shi and Bharagava [83] uses a secret key to change the sign bits of the DCT coefficients of MPEG video data. The secret key $(k_1, k_2, k_3, \cdots, k_{2m})$ is randomly generated with length 2m, where the number of key and the length of key is not limited. If the sign bits of DC and AC coefficients are represented by S,$(s_1, s_2, s_3, \cdots, s_{2m})$, then the encrypted data is $E_k(S_i) = b_i \bigoplus s_i$ of length $2m$, where $\bigoplus$ is the binary xor operation. The encryption operation randomly changes the sign bits of DCT coefficients. The decryption function E_k^{-1} is the same as the encryption function since $E_k(E_k) = S$. For a key of length m an adversary needs to try 2m times in order to find a key.

In this algorithm, several keys can be used to enhance the security. For example, in the 2 keys scheme, one key is for Y blocks and the other for C_b and C_r blocks. In the 3 keys scheme, one for I frames, one for B frames, and one for P frames.

3.5.3 Headers

Lookabaugh [84] proposed the selective encryption of MPEG-2 video, which used in most contemporary digital television applications. They use the fact that the typical high-performance MPEG-2 encoded bitstreams only use a small portion of bits (around 1 percent) in important headers (video sequence, group of pictures, picture, and slice). It can be simple to obscure such headers because of a usual practice in encoding of aligning these headers and the multiplex (transport) level at which encryption is performed.

However, fields in such headers can be quite vulnerable to attack, even if obscured by selective encryption, for a variety of reasons: the fields are often static, they can be guessed from external information that is probably available to an attacker, they can be guessed from other information in the bitstream (e.g., picture type can be guessed from picture size, an example of the cryptanalytic technique of traffic analysis), or they can be ignored, albeit with nontrivial consequences for decoded image quality.

They evaluated each of these fields, and proposed and tested attacks. For example, they showed that a perceptual attack on the quantizer-scale-code syntactic element is feasible albeit with nontrivial picture degradation: in typical sequences there is a strongly peaked distribution for this code, and a perceptual attack would be to always use an expected value for this code in place of the correct value. It is clearly that the resulting reconstruction is distorted, but it is not obvious that it is sufficiently distorted to cause a pirate to pay for a clean version if the distorted version is available for free. A more encouraging example is the choice of the macroblock-type field that signals to a decoder the type of prediction used for each macroblock (16-pixel vertical by 16-pixel horizontal region) in a video frame. Although this field does not use a very large fraction of the bitstream (on the order of a couple of percent typically), if absent it is very difficult for a decoder to guess it and to decode remaining material correctly (since the macroblock type uses a Huffman code and, if incorrectly decoded, a decoder has a hard time resynchronizing) [32].

3.5.4 Byte-Encryption

Griwodz et al. [85,86] propose to randomly encrypt bytes in an MPEG stream for free distribution, while the original bytes at the corresponding positions are transferred in encrypted form to legitimate users. This is actually equivalent to encrypting byte at random positions. The authors find that encrypting 1% of the data is sufficient to make a video undecodable or at least invisible. However, the cryptanalysis given is entirely insufficient. Consider the worst case where only MPEG header data is encrypted by chance using this approach. It is well known that header data may be reconstructed easily provided the encoder in use is known. Additionally, no attack scenario is considered but only the case of playing the protected video in a standard decoder is covered. In order to guarantee a certain level of security, a higher amount of bytes need to be encrypted and care needs to be taken about which bytes are encrypted. Wen at al. [87] describe a more general approach as part of the

MPEG-4 IPMP standard, named *Syntax Unware Runlength-based Selective Encryption* (SURSLE). This algorithm encrypt X bits, the next Y bits are left in plain-text, the next Z bits encrypted again, and so on. In addition to the above mentioned security problems, both schemes partially destroy the MPEG bitstream syntax and potentially emulate important MPEG markers causing a decoder to crash [12].

3.6 Comparisons of MPEG Video Encryption Algorithms

In this chapter, we described currently known encryption algorithms for MPEG video streams and evaluated them with respect to three metrics: security level, encryption speed, and encrypted MPEG stream size. As our summary Table 3.1 shows, Naïve Algorithm and Video Encryption Algorithm (VEA) are the most secure algorithms, where Zig-Zag Permutation Algorithm has serious security flaws and cannot withhold the known plaintext attack nor the ciphertext only attack. With respect to encryption speed, Pure Permutation Algorithm and Zig-Zag Permutation Algorithm are very fast, and Naïve Algorithm is very slow due to applying DES on whole MPEG stream. When comparing the algorithms in terms of size metric, VEA, Pure Permutation Algorithm and Naïve Algorithm do not change their size, which is very much desirable. On the other hand, Zig-Zag Permutation Algorithm significantly increase the stream size which defeats the compression purpose of MPEG encoding. In summary, there are trade offs when applying different encryption algorithms to MPEG encoded video and its choice depends on the applications. We believe that VEA meets the requirements of most multimedia applications because it provides overall high security, size preservation, and relatively fast encryption. Any other algorithms suffers from either low security, or low speed, or stream size increases [79].

Table 3.1
Comparisons of Video Encryption Algorithms

Algorithm	Security	Speed	Size	Encryption Ratio
Naive	High	Slow	No change	100%
Selective	Moderate	Fast	Increase	1% - 100%
Zig-zag Permutation	Very low	Very Fast	Big increase	100%
VEA	High	Fast	No change	50%
Pure permutation	Low	Super Fast	No change	100%

3.7 Commercial Applications and Standards

3.7.1 JPEG-2000 Part 8 Security Standard (JPSEC)

JPEG-2000 (Joint Photographic Experts Group) is the latest standard for still image coding [88–90]. JPEG2000 is designed to supplement and enhance the existing JPEG standard for still image coding. It provides advanced features such as low bitrate compression, lossless and lossy coding, resolution and quality scalability, progressive transmission, region-of-interest (ROI) coding, error-resilience, and spatial random access in a unified framework [91–93].

The final JPEG2000 bitstream is organized as follows: A set of different main headers (including a main header (SIZ), a coding style header (COD), a quantization header (QCD), a comments header (COM), a start of a tile parts header (SOT)) is followed by packets of data which are all preceded by a packet header. In each packet appear the codewords of the code-blocks that belong to the same image resolution and layer, the header identifies the data. Depending on the arrangement of the packets, different progression orders may be specified [94].

The emerging JPEG-2000 Part 8 Security standard (JPSEC) is being defined to provide security services for JPEG-2000 images. JPSEC is a standard also known as ISO/IEC 15444-8. These include many aspects of security such as confidentiality [95],

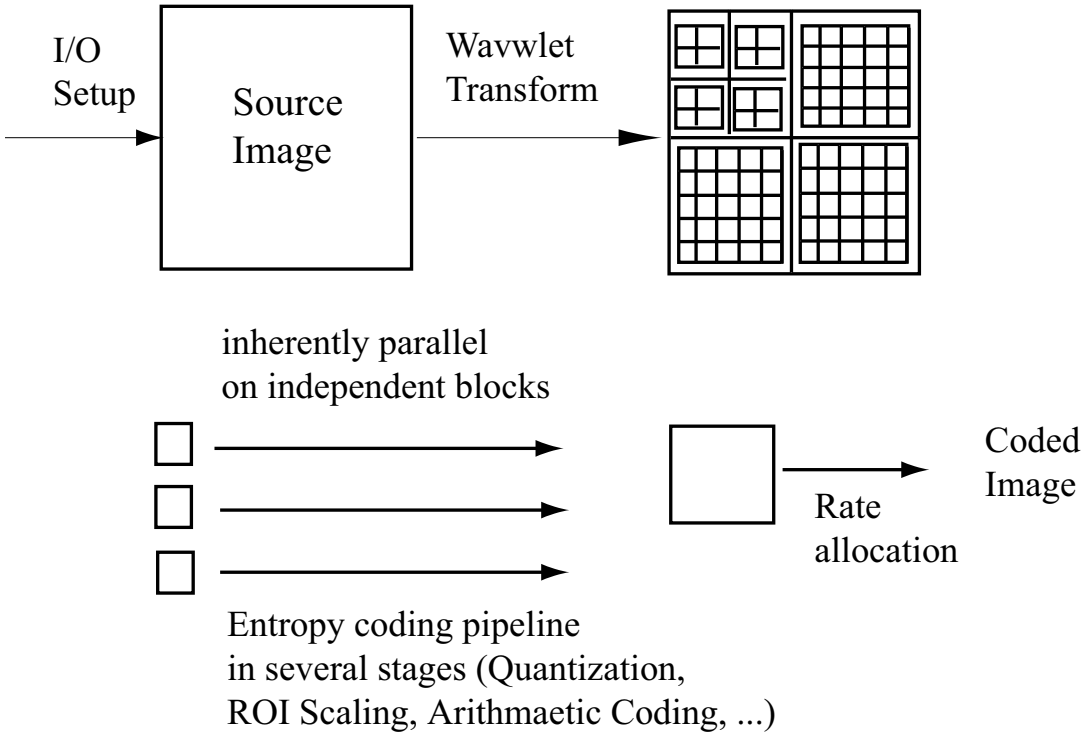


Fig. 3.3. Basic Structure of the JPSEC Coder.

authentication [96], integrity, conditional access [97], and ownership protection. These security services are achieved using techniques such as encryption, authentication, key generation and management, scrambling, and watermarking.

A JPEG-2000 codestream can be encrypted in a number of ways. Each encryption method has different implications on the privacy and transcoding flexibility of the protected codestream and on the complexity requirements of JPSEC creators, streamers, transcoders, and consumers. These requirements are especially critical for servers that adaptively stream and transcode large numbers of streams and thin clients that have limited device capabilities.

JPSEC defines an open and flexible framework for secure imaging as illustrated in Figure 3.4. A JPSEC protector application provides a number of security services (e.g. confidentiality, integrity verification, and source authentication). In order to

secure an image, it applies one or more JPSEC protection tools (e.g. encryption and digital signature). The resulting JPSEC codestream is generated by inserting in the stream the corresponding JPSEC syntax, signaling the JPSEC tools which have been used and how they have been applied to the image.

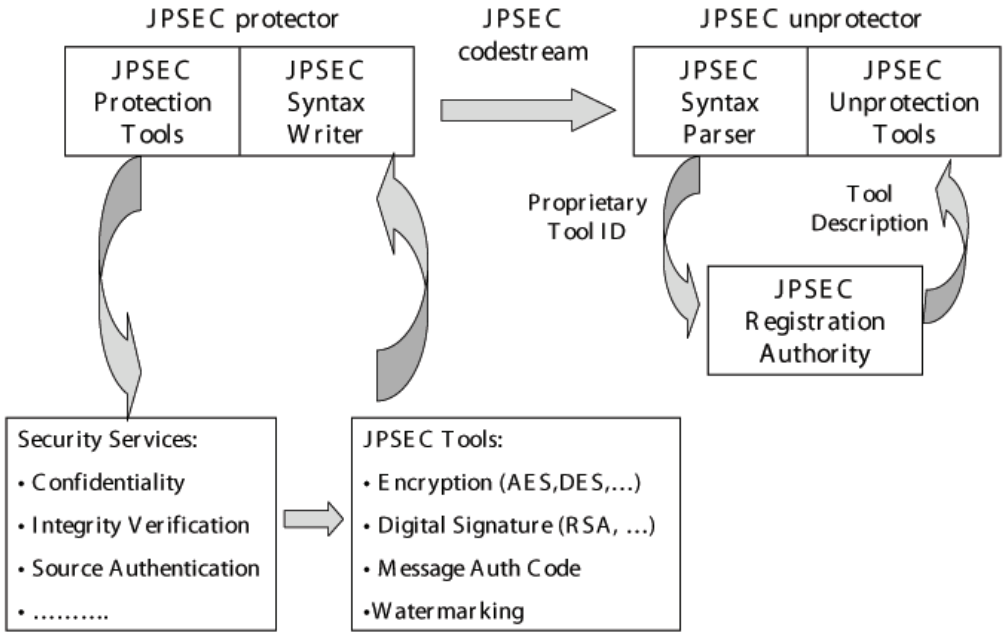


Fig. 3.4. JPSEC framework.

The JPSEC tools fall into two categories. The first category encompasses well-known cryptographic methods such as AES, DES, 3DES, RC4, RSA [98], MD5 [99, 100], and Secure Hash Algorithm (SHA-1) [101]. In this case, a number of templates are defined in order to specify method specific parameters. The tools in this category are therefore referred to as template protection tools. The syntax contains all the required information relative to the protection tool and how it has been applied. It

is therefore sufficient to enable a JPSEC application to unprotect the image data. The second category consists of proprietary tools. These tools have to be registered with the JPSEC Registration Authority (RA), they are then referred to as registration authority protection tools. Upon registration, a tool is assigned a unique identification number. In this case, the syntax contains the unique identification number along with private parameters. A JPSEC application may have to query the JPSEC RA in order to get a description of the tool and be able to unprotect the image data. With this registration process, provision is made for future tools to be identified and registered.

A JPSEC secure transcoding system is shown in Figure 3.5. An original image is encoded into a JPEG 2000 codestream. This is then locked with a key to form the JPSEC codestream. The JPSEC codestream can be unlocked back to a JPEG 2000 codestream by authorized entities using the appropriate key. This JPEG 2000 codestream can be decoded to reconstruct the image. The JPSEC codestream can also be securely transcended or scaled to a transcended JPSEC stream with a secure transcoding operation that does not require the key. The resulting transcended JPSEC codestream can be unlocked by authorized entities using the key to form a transcended JPEG 2000 codestream, which can then be decoded to reconstruct the image at a lower scale [102].

3.7.2 Intellectual Property Management and Protection (IPMP)

Intellectual property management and protection (IPMP) is a standard within MPEG family which has been developed at first for MPEG-2 and MPEG-4. IPMP tries to create a way of interoperability for the deployment of content and applications and distinguish between 5 different communities: end-users or consumers, content providers, device manufacturers, service providers, and content authors. IPMP tries to meet the goals of all these groups by the creation of an extensive frame work. One important part of this framework is the concept of the IPMP tools. They are modules that perform one or more functions like authentication, decryption or watermarking

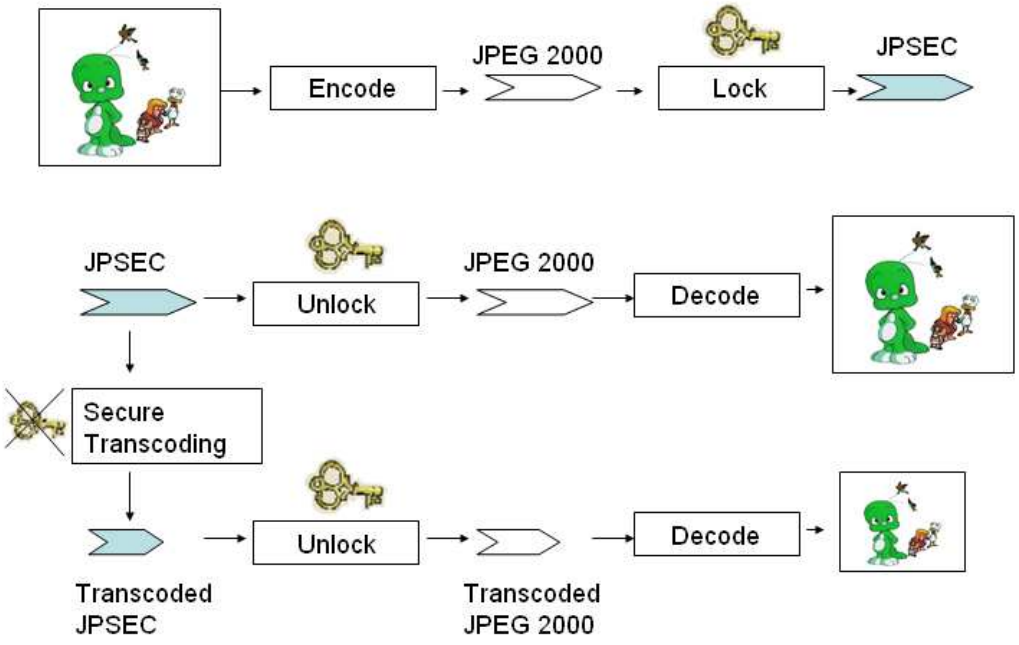


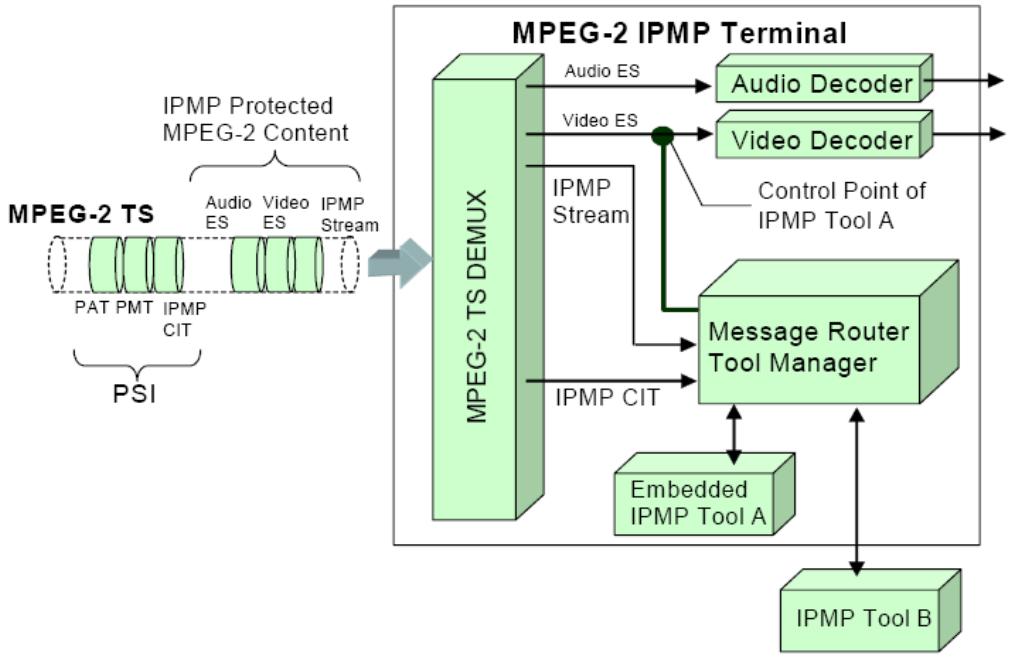
Fig. 3.5. Secure transcoding system diagram.

on an IPMP terminal, such modules are identified by an ID. They can be embedded in a bitstream, downloaded or acquired by other means [12]. The architecture and content of MPEG-2 IPMP are shown in Figure 3.6.

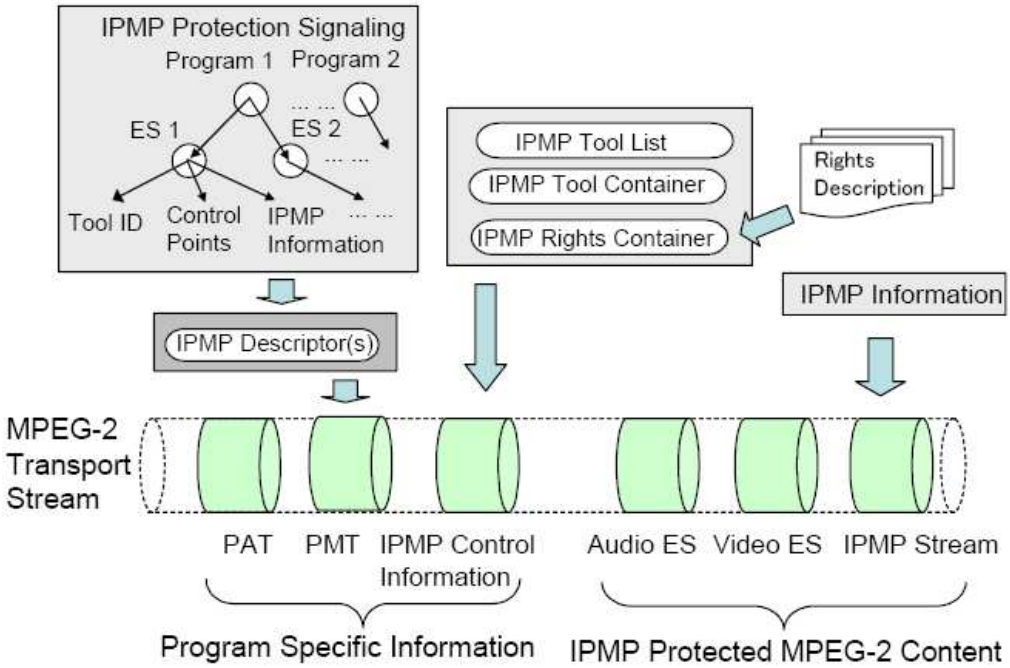
The Intellectual Property Management and Protection (IPMP) identifies carriers of creative works. The tool was developed as a complement of MPEG-4, the ISO compression standard for digital audio-visual material. Involved experts, notably those representing authors' societies, felt that MPEG-4 needed extra rules designed to protect intellectual property. To this end, IPMP was constructed as a supplementary layer on the standard [23].

3.8 Selective Bitstream Encryption Method

Selective encryption is a technique for encrypting parts of a compressed stream to minimize computational complexity [79]. Selective encryption is not a new idea. It



(a) Architecture



(b) Content

Fig. 3.6. IPMP-MPEG2.

has been proposed in several applications, especially in multimedia systems [32, 82]. Selective encryption can be used to reduce the power consumed by the encryption function for digital content when the content is protected by a digital rights management systems [8]. Since only parts of the bit stream are encrypted, selective encryption can also enable new system functionality such allowing previewing of content. For selective encryption to work, we need to rely not only on the beneficial effects of redundancy reduction described by Shannon [31], but also on a characteristics of the compression algorithm to concentrate important data relative to the original signal in a relatively small fraction of the compressed bitstream [32]. These important elements of the compressed data become candidates for selective encryption.

In our selective encryption, a bit stream is partially encrypted to minimize computational complexity or provide new functionalities for uses of the encrypted bit stream while at the same time providing "reasonable" security of the bit stream. One goal might be to provide additional error resilience in the case of a wireless network with packet losses and erasures.

The block diagram of our proposed selective encryption method for video compressed, using the DSC method described in the next section, is shown in Figure 3.7. The video sequence, X, is first compressed by the DSC encoder. The seed K' is used as the input to a pseudo-random generator (PRG), whose output is denoted by K. If K is truly random, then the PRG forms a stream cipher.

The output bitstream of the DSC encoder consists of several types of data: the video frames (pixels), the parity bits, intra frame, and feedback data. The encoded bit stream, W, is partially encrypted by forming the bitwise binary sum $Z = E[X] = W \oplus K$ of parts of the compressed bit stream. Then, Z is transmitted over the channel. The adversary is assumed to be able to eavesdrop on the ciphertext Z. We assume that the seed has been transmitted to the decoder through a secure channel. By implementing an identical PRG, the decoder also has access to K. Our goal is to decode X, using the fact that K is available at the decoder and the half frame of Y, of

the correlated video sequence, is available as side information. Because $Z = W \oplus K$, it follows that $W = Z \oplus K$.

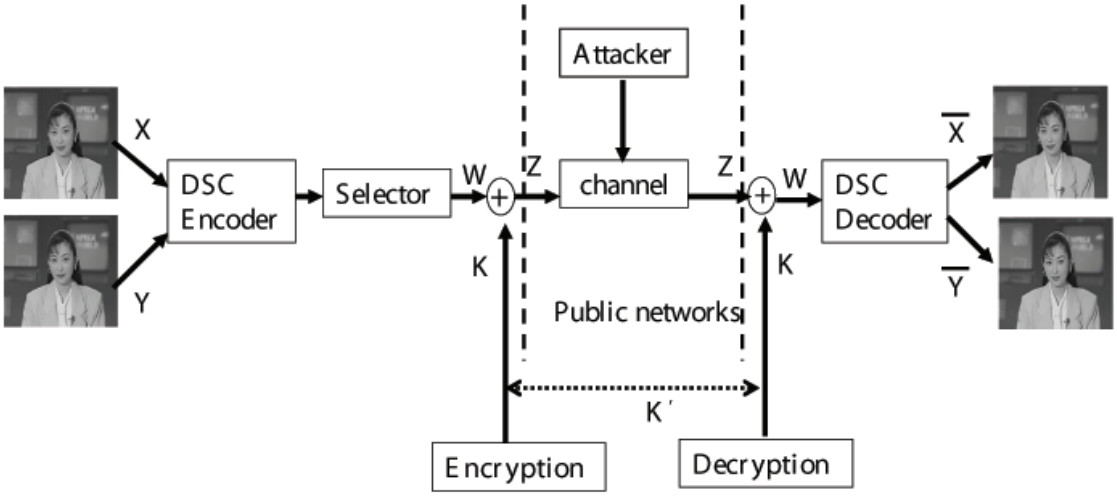


Fig. 3.7. Diagram of the proposed selective encryption method.

4. SELECTIVE ENCRYPTION OF THE DISTRIBUTED VIDEO CODED BITSTREAMS

The focus of this chapter is to examine the use of selective encryption on a compressed bit stream that has been encoded using distributed source coding. We studied how selective encryption can achieve a high level of effectiveness. By this, we mean a strategy in which even a small fraction of encrypted bits can cause a video sequence to become useless if an attacker attempts to decode it without decrypting the secured portions. In this study, we examined which types of bits are most effective for selective encryption. Instead of encrypting the entire video sequence bit by bit, we encrypted only these highly sensitive bits.

4.1 Distributed Video Coding Based on LDPC Codes

Low-density parity-check (LDPC) codes are a class of linear error-correcting codes [103]. Linear codes use a generator matrix $\mathbf{G}$ to map messages s to transmitted blocks $\mathbf{x}$, also known as codewords. They have an equivalent description in terms of a related parity-check matrix $\mathbf{H}$ with M rows and N columns. All codewords $\mathbf{x}$, of length N, satisfy $\mathbf{Hx} = 0$. Each row of $\mathbf{H}$ represents a parity check on a subset of the bits in $\mathbf{x}$; all these parity checks must be satisfied for $\mathbf{x}$ to be a codeword.

As their name suggests, low-density parity-check codes are defined in terms of parity-check matrices $\mathbf{H}$ that consist almost entirely of zeroes. Gallager [103] defined (n, p, q) LDPC codes to have a blocklength n and a parity-check matrix with exactly p ones per column and q ones per row, where $p \geq 3$. If all the rows are linearly independent then the rate of the code is $(q-p)/q$, otherwise the rate is $(n-p^{'})/n$ where $p^{'}$ is the dimension of the row space of $\mathbf{H}$.

We now define some basic notation we will use to describe low-density parity-check (LDPC) codes. LDPC codes are well represented by *bipartite graphs* in which one set of nodes, the *variable nodes*, corresponds to elements of a codeword (bits) and the other set of nodes, the *check nodes*, corresponds to the set of parity-check constraints which define the code. For a given length and a given degree distribution, we define an *ensemble* of codes by choosing edges, i.e., the connections between variable and check nodes, randomly. More precisely, we enumerate the edges emanating from the variable nodes in some arbitrary order and proceed in the same way with the edges emanating from the check nodes [104].

Definition 1: LDPC codes [103] are best described by their parity-check matrix **H** and the associated bipartite graph. The parity-check matrix **H** of a binary LDPC has a small number of ones. The way of spreading ones in **H** is described by the degree distribution polynomial $\lambda(x)$ and $\rho(x)$, which indicate the percentage of columns and rows of **H** respectively, with different Hamming weights. When both $\lambda(x)$ and $\rho(x)$ have only a single term, the LDPC code is *regular*, otherwise it is *irregular* [105].

Definition 2: The bipartite of an LDPC code is an equivalent representation of the parity-check matrix **H**. Each column is represented with a *variable node* and each row with a *check node*. The graph has an *edge* between variable node j and check-node i if $H(i,j) = 1$ [105, 106].

In this paper, we used a distributed source coding method based on non-uniform LDPC coding [65] at the symmetric rates (point C of Figure 2.1), i.e., both of the encoders are compressed at the same rate. That is, $R_x = R_y = \frac{H(X,Y)}{2} = \frac{1}{2} + \frac{H(p)}{2}$, where $H(p) = H(X|Y) = -plog(p) - (1-p)log(1-p)$ and p is the crossover probability of $P[X \neq Y|X] = p$. We shall refer to this as a symmetric LDPC code. Here X and Y are assumed independent, identically distributed binary sequences of length k. X and Y are statistically dependent to each other and the dependency can be described by the conditional mass function $P[X_1|X_2]$. The correlation between X and Y can be modeled as the input and output of a binary symmetric channel with crossover

probability of $P[X \neq Y|X] = p$. We assume that the length of the LDPC code of rate R is n.

4.1.1 LDPC Encoder

We use the DSC coder described in [65] and shown in Figure 4.1 for compressing a video sequence. To use this method to encode video we will let $\mathbf{X}$ be one frame of video and $\mathbf{Y}$ be the next frame of video in a sequence. We will transmit the upper half of $\mathbf{X}$ and the lower half of $\mathbf{Y}$ along with parity bits associated with each frame. We will use the lower half of $\mathbf{Y}$ to reconstruct the lower half of $\mathbf{X}$ using the parity bits for both $\mathbf{X}$ and $\mathbf{Y}$. We will then do a similar operation to reconstruct the upper half of $\mathbf{Y}$. Our goal is to investigate which parts of the bit stream of this encoder needs to protected. Using a linear binary (n,k) block code, such as a LDPC code, there are 2^{n-k} distinct syndromes, each indexing a set of 2^k binary words of length n. All sets are disjoint and in each set the Hamming distance properties of the original code are preserved, i.e., all codes have the same performance over the binary symmetric correlation channel. For the encoder, a sequence of input n bits is mapped into its corresponding syndrome $(n-k)$ bits. Thus, the compression ratio achieved with this scheme is n:(n-k) [107].

As shown in Figure 4.1, two half sets of each video frame are used to encode $\mathbf{X}$ and $\mathbf{Y}$, where $\mathbf{X}$ and $\mathbf{Y}$ represent even and odd number video frames respectively (the two video frames are correlated). The even video frame X is used an the input to a rate R_x systematic LDPC encoder. At the output of the encoder, the first half of the input video frame, x_1, and the corresponding parity check bits, p_1 are transmitted. This results in a source encoding rate of $R_{x1} = \frac{k/2+p_1}{k}$ bit per input bit. That is, the rate of the systematic LDPC code is equal to $R_x = \frac{k}{n} = \frac{1}{R_{x1}+1/2}$. A corresponding operation is performed on the odd video frame Y similar to that of the even video frame X. However, the second half of the video frame and the corresponding parity bits are used for this case. Since the compression rate of both video frames are the

same, the rate of the systematic LDPC codes are identical. Hence only a single LDPC code is needed [65].

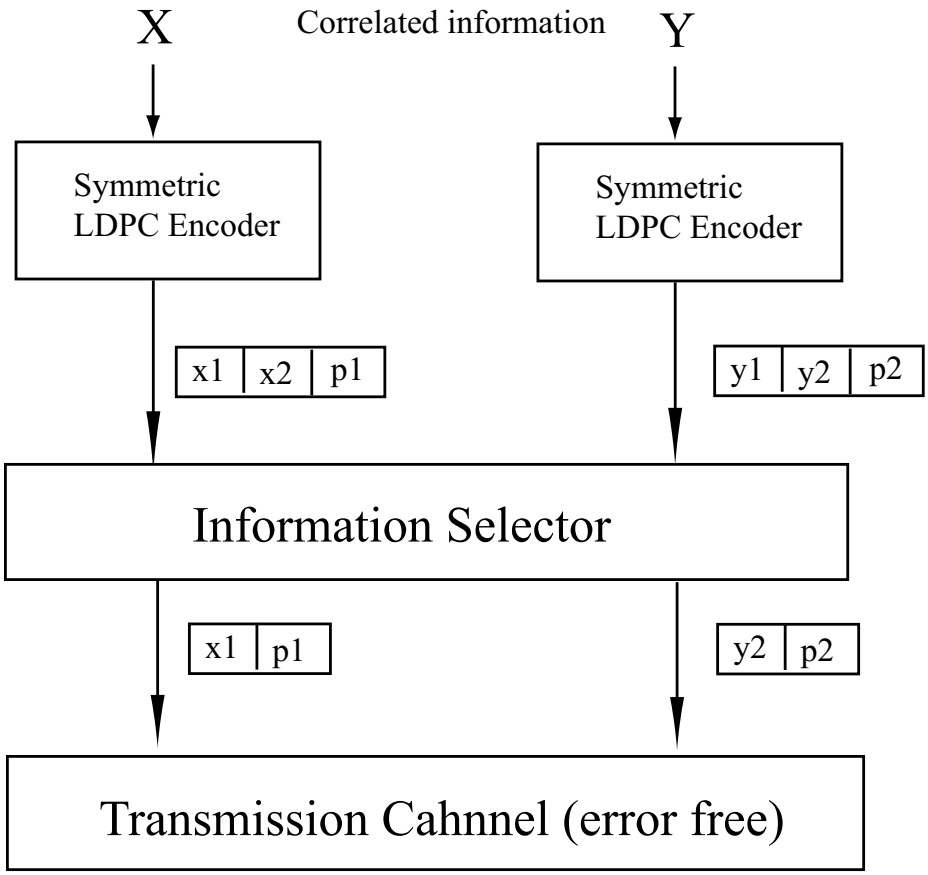


Fig. 4.1. Symmetric LDPC Encoding.

4.1.2 LDPC Decoder

The decoder estimate the n-length video frame **X** from its (n-k)-long syndromes and the side information, the half video frame of **Y**. The transmitted codewords are decoded from the received data, the two half video frames and the parity bits, using the likelihood of the possible codewords. The likelihood of the possible codewords is the probability of receiving the data that was actually received if the codewords is question were the one that was sent. For decoding purposes, the most important

issue is the relative likelihood for a bit to be 1 versus 0. This is captured by the likelihood ratio in favor of a 1, which is $P(data|bit = 1)/P(data|bit = 0)$.

Definition 3: For a Binary Symmetric Channel with error probability p, the likelihood ratio in favor of 1 bit is as follows: (a) if the received data was +1: $(1-p)/p$. (b) if the received data was -1: $p/(1-p)$.

The decoder of X has the first half of X perfectly, $x1$, (here we assume that the channel is error free). To construct the entire video frame for X, the decoder use the lower half of Y and the parity bits of X, p_1. The log likelihood ratios (LLRs) of all bits should be known in order to use the *message passing algorithm*, which will be described in the next section in detail. The LLRs of the parity bits and half video frames that passed through the channel are infinity. The lower half of Y is assumed to be the output of a binary symmetric channel (BSC) with cross over probability of p whose input is X. The LLRs of this fraction of the video frames are equal to $ln(\frac{1-p}{p})$. Then, by knowing the LLRs for all the bits, the message passing algorithm can decode the video frame X. The same process can be used to decode the video frame Y [65].

4.1.3 Implementation of Selective Video Encryption

In this scheme, the candidates for encryption are the half video frames (pixels) and the parity bits. Now we discuss the issue of the partial encryption for each candidate.

Parity Bits

Decoding can be done using only the parity check matrix defining the codewords, without reference to the generator matrix defining the mapping from the source messages to the codewords. Hence the parity bits must be encrypted. Otherwise an attacker can recover the original video frame directly. Below we describe how the parity bits are used by the decoder.

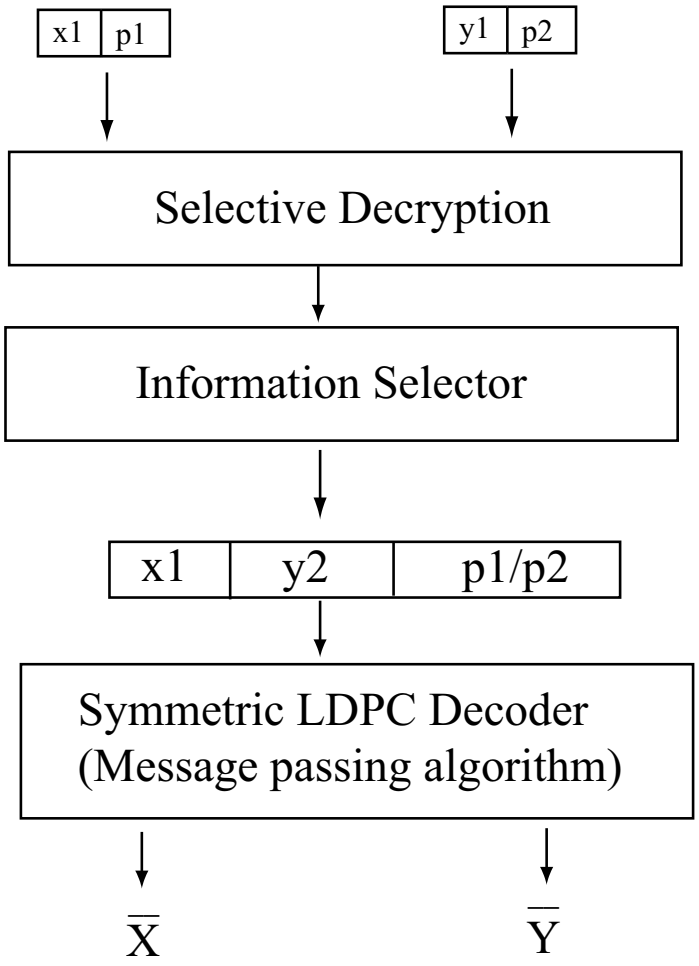


Fig. 4.2. Symmetric LDPC Decoding.

Assuming equal a priori probabilities for the codewords, the probability of correctly decoding an entire codeword is minimized by picking the codeword with the highest likelihood. One might instead wish to decode each bit to the value that is most probable. This minimizes the bit error rate, but is not in general guaranteed to lead a decoding for each block to the most probable complete codeword; indeed, the decoding may not be a codeword at all. Minimizing the bit error rate seems nevertheless to be the most sensible objective, unless block boundaries have some significance in a wider context. The begin, information about each bit of the codeword derived from the received data for that bit alone is expressed as a probability ratio,

the probability of the bit being 1 divided by the probability of the bit being 0. This probability ratio is equal to the likelihood ratio for that bit, since 0 and 1 are assumed to be equally likely a priori. As the decoding algorithm progresses, these probability ratios will be modified to take account of information obtained from other bits, in conjunction with the requirement that the parity checks be satisfied. To avoid double counting of information, for every bit, the algorithm maintains a separate probability ratio for each parity check that bit participates in, giving the probability for that bit to be 1 versus 0 based only on information derived from other parity checks, along with the data received for the bit.

As indicated above, the parity bits are used to decide what are the correct information bits in the decoding process. This means that using encrypted parity bits for source decoding would render the decoded video useless because the decoder would generate the wrong codewords.

The Half Video Frames

The half video frames must be encrypted because they reveals parts of the original video. We consider each 8 bit pixel of the frame in the form of 8 bitplanes. Our approach is to encrypt a subset of the bitplanes, starting with the bitplane containing the most significant bit (MSB) of the pixel and increasing to the least significant bit (LSB) of the pixel. By doing this the encrypted pixels are less likely to show any of the original gray scale information. The minimal percentage of pixels to be encrypted is 12.5% when encrypting 1 bit. We increase the percentage of the pixels encrypted in steps of 12.5%.

4.1.4 Simulation Results

Consider a single video frame (image) composed of $M \times N$ pixels (where M is the width and N the height of the image) where each image is in the YUV color space [108]. The YUV format is typically sub-sampled and for our work we will

use the 4:1:1 format. We will also use Peak-Signal-to-Noise-Ratio (PSNR) for our measure of image quality.

In our simulation, we used QCIF video sequences with 176×144 pixels as shown in Figure 4.3. The video is compressed using the LDPC coder discussed in the previous section. Our selective encryption algorithm encrypts the parity bits and a subset of the bitplanes for each pixel, starting with the most significant bit (MSB). The encrypted bitplanes are transmitted as plaintext.

Fig. 4.3. Original Video Sequence: YUV 4:1:1 sub-sampled with 176×144 pixels.

Figure 4.4 shows a reconstructed video frame after encrypting only the parity bits. We assumed that an attacker does not have access to the encryption key. Hence the attacker can access the unencrypted half video frames and reconstruct the video frames by combining the two half frames. The reconstructed video frames are very similar to the original frame. We note that the encryption of only the parity bits cannot guarantee the security of the video sequence.

Figure 4.5 shows four examples of reconstructed video frames after selectively encrypting MSBs of the pixels along with the parity bits. We encrypted the first MSB, the first two MSBs, the first four MSBs, and all 8 bits respectively.

In the case of encrypting the MSB and the parity bits, structural information is still visible, but the encryption of two or more bits and the parity bits reveals no

(a) The upper half frame of X.

(b) The lower half frame of Y.

(c) The reconstructed video frame.

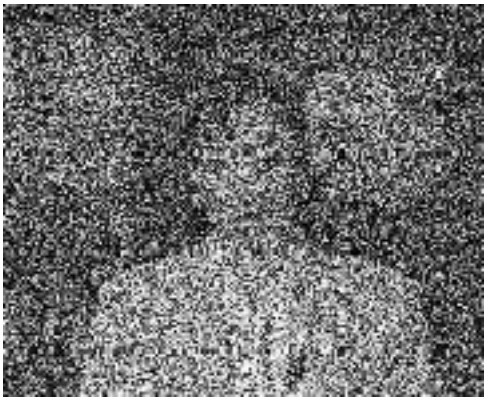
Fig. 4.4. Results when only the parity bits are encrypted.

useful information in the reconstructed frames. The PSNR decreases steadily from 18dB to 9dB as we encrypt more MSBs.

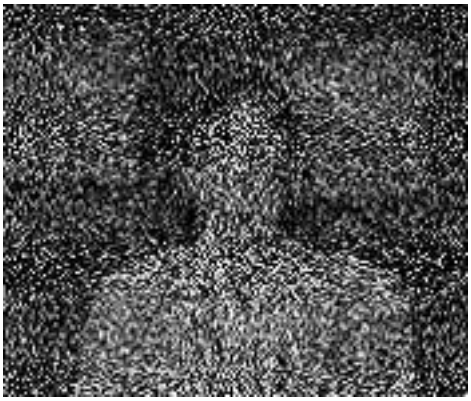
Figure 4.6 shows the case where the frame is reconstructed after the LSBs and the parity bits are encrypted. It shows that the PSNR decreases steadily from 117dB to 14dB for each additional bitplane encrypted and reaches 9dB when encrypting all bitplanes after all in case when the LSB bitplane is encrypted first.

As a consequence, the most secure way to perform selective bitplane encryption is to encrypt the MSB bitplane and subsequently additional bitplanes in order of decreasing significance with respect to their position in the binary representation.

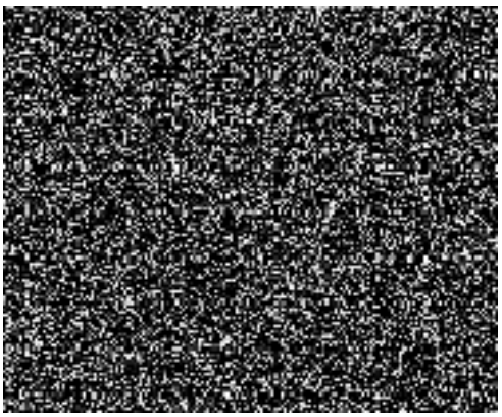
In these simulations, the rate of the systematic LDPC code is 1/2, $\frac{k}{n} = \frac{1}{2}$. We define the encryption ratio as the ratio of the number of encrypted bits $((i/8)\times(2/n)+(n-k))$ to the number of data bits $(2/n+(n-k))$. The percentage of encrypted bits are shown in Table 4.1.



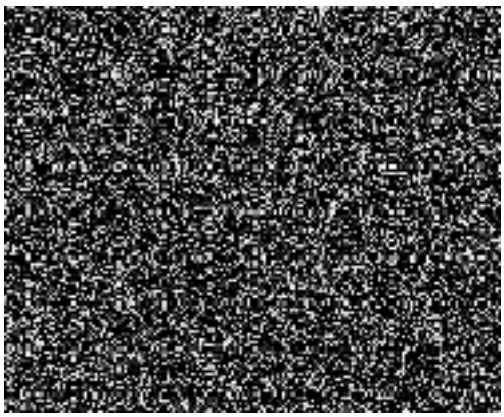
(a) 1MSB and parity bits are encrypted, PSNR=18.18dB.



(b) 2MSBs and parity bits are encrypted, PSNR=13.23dB.



(c) 4MSBs and parity bits are encrypted, PSNR=10.82dB.



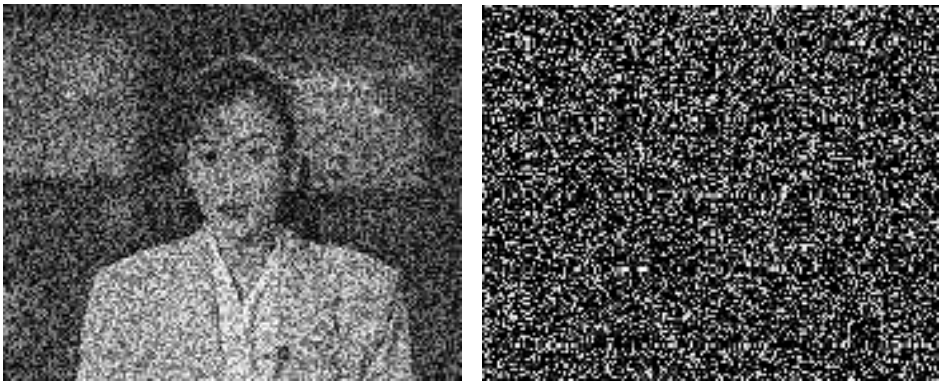
(d) 8MSBs and parity bits are encrypted, PSNR=9.94dB

Fig. 4.5. Visual examples of the selective encryption when MSBs and the parity bits are encrypted.

The experiments indicate that at least 63% of the bits need to be encrypted. This is larger than has been reported for selective encryption methods for MPEG-2 [32] where the syntax of the bit stream has been exploited. Our results are also due to the relative simple scheme we used for distributed source coding.

(a) 1LSB and parity bits are encrypted, PSNR=107.71dB.

(b) 2LSBs and parity bits are encrypted, PSNR=63.36dB.



(c) 4LSBs and parity bits are encrypted, PSNR=42.25dB.

(d) 8LSBs and parity bits are encrypted, PSNR=9.94dB

Fig. 4.6. Visual examples of the selective encryption when LSBs and the parity bits are encrypted.

4.2 Distributed Video Coding Based on Turbo Codes

In this section, we describe a DSC using Turbo code that is called Network Driven Motion Estimation (NDME) [52,53,59]. In standard Turbo code system [109], all systematic and parity bits are transmitted with equal power allocation. Turbo code is proposed as a way of dramatically reducing the errors in a forward error correction system. It can achieve near Shannon limit. This scheme combines the concepts of iterative decoding, soft-in/soft-out, recursive systematic convolutional (RSC) encoding

Table 4.1
Encryption ratios.

Encryption methods	Encryption ratio
parity bits only	50%
1 bit and parity bits	56%
2 bits and parity bits	63%
4 bits and parity bits	75%
8 bits and parity bits	100%

and interleaving. The Bahl-Jelineck algorithm, also known as the symbol-by-symbol MAP algorithm (MAP algorithm for short), is optimal for estimating the states or outputs of a Markov process observed in additive white Gaussian noise (AWGN). Two MAP decoders were used within the Turbo code decoder to produce the original results.

However, in NDME the motion estimation task is moved to the decoder. In network driven motion estimation, the motion search is performed at the decoder without accessing the current frame and the motion vectors are sent back to the encoder through a feedback channel. For a given frame, the system adaptively makes mode decisions from three modes: intra coding mode, conditional replenishment (CR) mode, and forward motion compensation mode. Intra coding mode is used to code the first frame of the sequence and to refresh the sequence. CR encodes the frame by sending the difference frame between the previous frame and the current frame. It is equivalent to forward motion compensation with no motion vectors sent back through the feedback channel. CR is chosen when the variance of the difference frame is comparably small. Through experiments, NDME [110] has shown to have a good rate distortion performance that is slightly worse than conventional encoder-based motion-compensated prediction (MCP) video coding, and much better than the coding scheme with only the CR and the intra coding modes.

4.2.1 Turbo Codec

The key idea behind our Wyner-Ziv approach with NDME is to inter-code those frames that were intra-coded in the original Wyner-Ziv coding structure. Motion vectors are estimated for these inter-coded frames at the decoder and sent back from the decoder to the encoder using a feedback channel. Video coding efficiency can thus be greatly improved while low complexity still remains at the encoder. Our system combines a "typical" Wyner-Ziv [68] with NDME that was developed for wireless systems [110]. The system diagram is shown in Figure 4.7. Three frame types are used in the system: intra-coded frame (I frame), network-driven frame (N frame) [110], and the Wyner-Ziv frame [68]. The Wyner-Ziv frames are coded using a Wyner-Ziv codec whose side information at the decoder is provided from the decoded I and N frames or a function of them. N frames are forward predictive frames coded in a similar manner as P frames [111] except using motion vectors estimated at the decoder. We may only encode the first frame as an I frame and alternatively encode the remaining frames as N frames and Wyner-Ziv frames. I frames have been intensively used by many Wyner-Ziv video coders, which consume considerable data rate. Our scheme is able to replace all I frames except for the very first one with N frames and hence a low complex but efficient video encoding approach is obtained. The details of our approach are described in the following sections.

4.2.2 Wyner-Ziv Video Codec

For Wyner-Ziv frames at the encoder, this scheme is operated in the pixel domain. The frames are encoded without the information of adjacent frames and decoded using the side information obtained from its neighboring frames. We refer to such a frame that is intra-coded and inter-decoded as a Wyner-Ziv frame. Every pixel is uniformly quantized to 2^M levels and represented by M bits. The resulting codewords are encoded by a Slepian-Wolf coder which is implemented by a Turbo code. We follow a previous method [71] to use rate compatible punctured Turbo codes (RCPT)

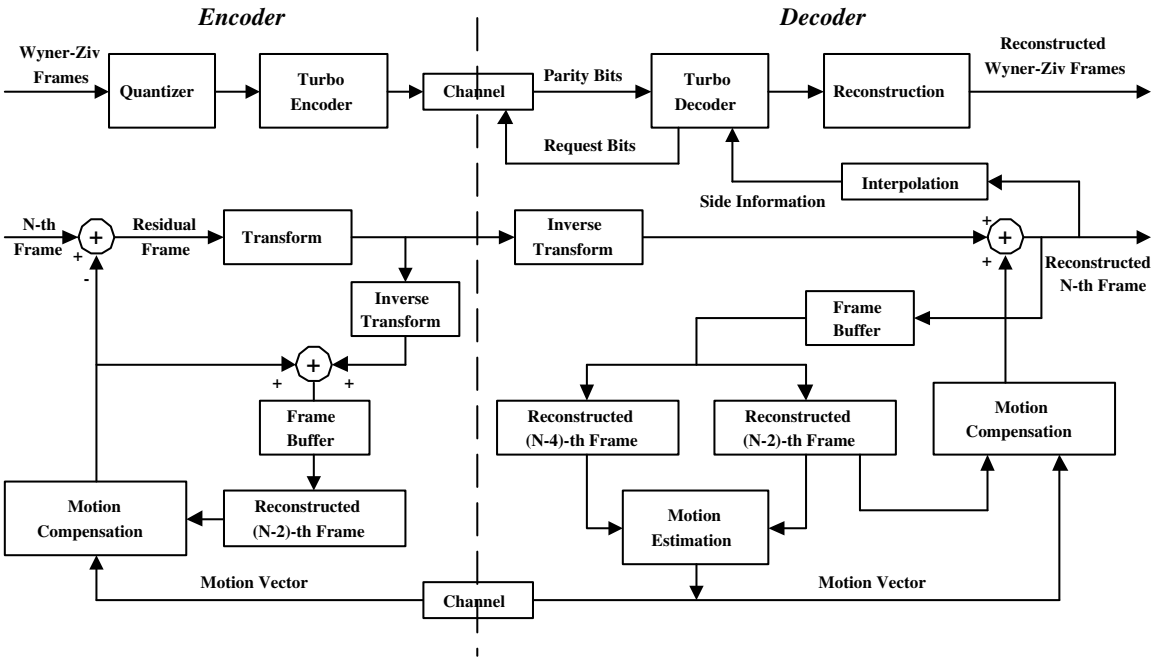


Fig. 4.7. Network-driven Wyner-Ziv video coding using forward prediction.

[109,112]. Two parallel recursive systematic convolutional (RSC) encoders constitute the Turbo encoder. The quantized symbols are directly sent to one RSC encoder and sent to the other encoder after a random interleaver. A portion of the parity bits is sent to the decoder through the channel, the systematic bits are completely discarded. At the decoder, a Turbo decoder decodes the parity bits with side information. The side information is an estimation of the current Wyner-Ziv frame obtained from the previously decoded frames. There are many ways to generate the side information. For example, motion information can be extracted from previously decoded frames and an estimation is obtained as the motion compensated prediction. One easy way is simply take the previous and the following decoded N frames as the reference and then the average two reference frames is used as the initial estimation of the current frame. This method is implemented in our experiments. To generate side information in this way, we note that the sequence has to be decoded in a B-frame-like manner, i.e., a Wyner-Ziv frame's previous and successive neighboring frames have

to be decoded prior to the decoding of the Wyner-Ziv frame. The Turbo decoder uses this initial estimation and incoming parity bits to decode the current frame. If the initial estimation is coincident with the parity bits, the Turbo decoder works in the normal way. Otherwise, the decoder disregards the received parity bits and uses the quantization bin which is nearest to the estimation. Under the worst conditions, the maximum distortion is proportional to the coarseness of the quantizer. Thus, the scheme prevents the decoder from having large errors.

The quality of the reference frames, or the initial estimation, is essential to improving the Wyner-Ziv frame's coding efficiency. A more accurate estimation requires less parity bits for the decoding of the current Wyner-Ziv frame to obtain the same quality. Many existing Wyner-Ziv video coders [71] code the reference frames in an intra-coded mode, whose poor rate distortion performance also affects the decoding of Wyner-Ziv frames. Under the same data rate constraint, the use of an I frame results in low decoded quality, which will further degrade the accuracy of the estimation (side information) for the decoding of the Wyner-Ziv frames. In our scheme, we use N frames instead of I frames, which not only greatly improves the rate-distortion performance of encoding the N frames themselves, but further provide references frames with much higher quality for the efficient decoding of the Wyner-Ziv frames.

4.2.3 Feedback Channel Motion Estimation: N Frames

Network-driven frames using the motion vector derived from the decoder are encoded, which is adopted from Rabiner's paper [110]. The difference relative to conventional encoder-based video coding is that motion search is implemented in the decoder. These frames are refereed as N frames.

As shown in figure 4.7, motion search is carried out between the previous two N frames. For each macroblock in the current frame, its co-located macroblock in the previous decoded N frame is used as a source and the second previous decoded N frame as the reference. Forward motion search is used to obtain the motion vector for the

co-located macroblock. This motion vector is regarded as a prediction of the motion vector for the current macroblock and sent back to the encoder through the feedback channel. In the encoder, the predicted motion vector is used, which is obtained from the previous N frame, and the current frame is encoded in a conventional MCP-based video coding manner.

4.2.4 Mode Selection

As discussed above, motion vectors of the N frames are derived at the decoder and then sent back to the encoder. The motion vectors of the already decoded N frames are used to interpolate/extrapolate the motion vectors for the current N frame. If we have several candidate motion vectors derived for the current N frame, we can then use mode selection to choose the best motion vectors at the encoder and further improve coding efficiency. Note that this will incur additional coding complexity at the encoder. However since we limit the number of available modes, generally 3 to 6, the extra coding complexity is very marginal.

Mode I: forward motion vector

The basic method to estimate the motion vector for encoding the N frames is shown in Figure 4.8 and referred to as Mode I. Reference A and reference B are the previous two decoded N frames stored in the frame buffer at the decoder. Let C denote the current N frame. The temporal distances between adjacent frames are denoted as TD_{AB} and TD_{BC} respectively. To find the motion vector in the current macroblock of the current frame, we use the motion information of the co-located macroblock in the previous frame by assuming that a constant translational motion velocity remains across frames. For each macroblock in current frame, we consider the co-located macroblock in B and search the best match in A to obtain the forward motion vector MV_F. The motion vector of current macroblock in type I mode can be obtained as:

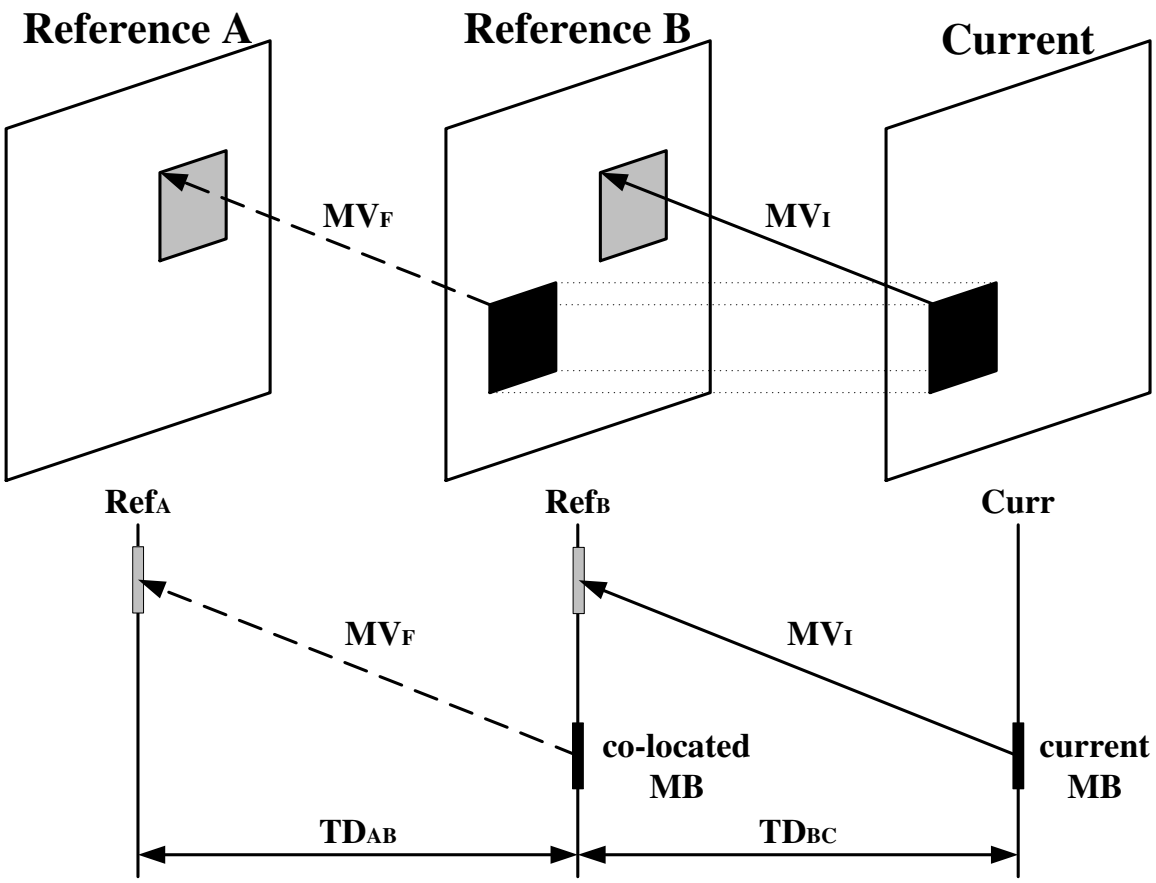


Fig. 4.8. Mode I: using forward motion vector.

$$MV_I = \frac{TD_{BC}}{TD_{AB}} MV_F \tag{4.1}$$

Since $TD_{AB} = TD_{BC} = 2$ and hence $MV_I = MV_F$. This mode is similar to that designed for encoding P frames [111] and we regard it as the basic mode.

Mode II: backward motion vector

The second mode is obtained based on the same assumption as Mode I but we consider the co-located macroblock in A as the source. Figure 4.9 illustrates Mode II using backward motion vectors of the co-located macroblock to extrapolate the

motion vectors for the current macroblock. We search the best matched macroblock in reference B. The motion vector obtained is referred to as a backward motion vector MV_B. Again assuming constant translational motion, the motion vector for current macroblock is

$$MV_{II} = -\frac{TD_{AC}}{TD_{AB}} MV_B \tag{4.2}$$

In our case, $TD_{AB} = 2$ and $TD_{AC} = 4$, and hence $MV_{II} = -2MV_B$.

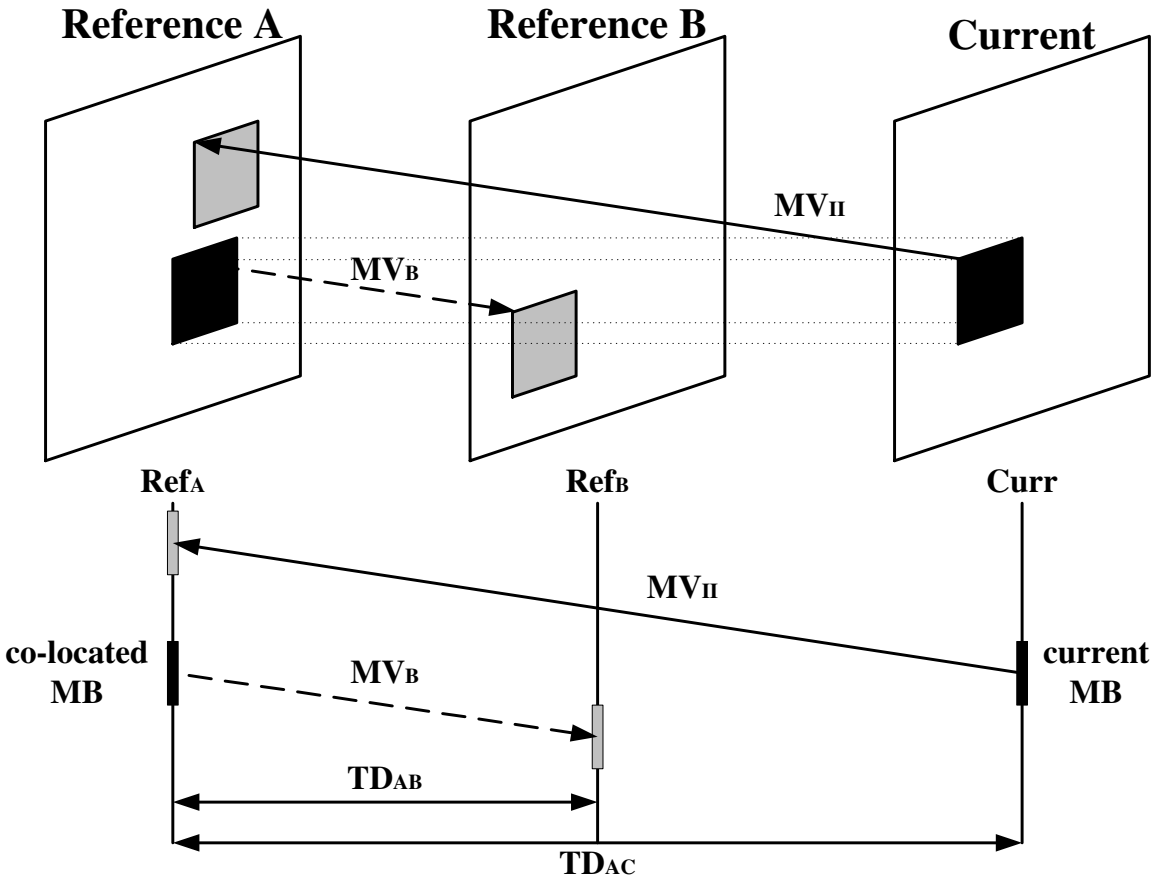


Fig. 4.9. Mode II: using backward motion vector.

Mode III: combination of Mode I and Mode II

In this mode, two motion vectors with different reference frames are first obtained using Mode I and Mode II. We then derive a weighted combination the two motion compensated frames using these two motion vectors and refer to it as Mode III. Suppose the motion compensated prediction frame obtained in Mode I is denoted as MC_I and that obtained in Mode II is denoted as MC_{II}. The motion compensated frame in Mode III is then obtained as:

$$MC_{III} = \frac{TD_{AC}}{TD_{AC} + TD_{BC}} MC_I + \frac{TD_{BC}}{TD_{AC} + TD_{BC}} MC_{II} \tag{4.3}$$

where TD_{AC} and TD_{BC} are the same as shown in Figure 4.8 and 4.9.

Mode Selection

We send the motion vectors obtained in Mode I and Mode II back to the encoder and obtain three candidate references motion compensated frames: MC_I, MC_{II}, and their weighted combination MC_{III}. We design the mode decision by choosing the best mode that minimizes the mean square error of the predicted error frame (PEF):

$$\text{Optimal Mode} = \arg \min_{k \in \{I,II,III\}} \frac{\sum_{(i,j)} (x(i,j) - \hat{x}^{(k)}(i,j))^2}{N \times N}, \tag{4.4}$$

where k denotes the index of the three modes, $x(i,j)$ denotes the original pixel value in the position (i,j), $\hat{x}^{(k)}(i,j)$ denotes the reconstructed pixel value using mode k, N represents the size of the macroblock, and the summation is over all the pixels of the current macroblock. According to this measurement of fidelity, we attain the optimal mode with highest peak signal-to-noise ratio (PSNR).

4.2.5 Implementation of Selective Video Encryption

In this scheme, the candidates for encryption are the Wyner-Ziv frame (parity bits) and the motion vectors (N frames). Now we discuss the issue of the partial encryption for each candidate.

Parity Bits

In this scheme, a portion of the parity bits is sent to the decoder through the channel and the systematic bits are completely discarded. At the decoder, a Turbo decoder decodes the parity bits with side information. The side information is an estimation of the current Wyner-Ziv frame obtained from the previously decoded frames.

As indicated in the previous section, the parity bits are used to decide what are the correct information bits in the decoding process. This means that using encrypted parity bits for source decoding would render the decoded video useless because the decoder would generate the wrong codewords. Hence the parity bits must be encrypted. Otherwise an attacker can recover the original video frame.

N Frames

Compared to other Wyner-Ziv video coding methods, we replace the I-frames with N frames. Thus, we still exploit the temporal correlation of the video sequence while moving the motion estimation from the encoder to the decoder. In fact, we can regard a N frame as a pseudo P frame, the difference from a conventional P frame is that motion vectors are obtained without the use of a current frame.

Motion search is carried out between the previous two N frames as shown in Figure 4.7. For each macroblock in the current frame, its co-located macroblock in the previous decoded N frame is used as a source and the second previous decoded N frame as the reference. Forward motion search is used to obtain the motion vector for the

co-located macroblock. This motion vector is regarded as a prediction of the motion vector for the current macroblock and sent back to the encoder through the feedback channel. In the encoder, the predicted motion vector is used, which is obtained from the previous N frame, and the current frame is encoded in a conventional MCP-based video coding manner.

The motion vectors are used to decide what are the correct information bits in the decoding process along with the parity bits. This means that using encrypted N frames for source decoding would render the decoded video useless because the decoder would generate the wrong codewords.

Motion vectors of the N frames are derived at the decoder and then sent back to the encoder. The motion vectors of the already decoded N frames are used to interpolate/extrapolate the motion vectors for the current N frame. If several candidate motion vectors had derived for the current N frame, mode selection is used to choose the best motion vectors at the encoder and further improve coding efficiency.

The motion vectors are used to decide what are the correct information bits in the decoding process along with the parity bits and N frames. Hence the feedback data must be encrypted to so that an attacker can't recover the original video frame.

4.2.6 Simulation Results

In our simulation, we used QCIF video sequences with 176×144 pixels as shown in Figure 4.10. The video is compressed using the Turbo coder and the intraframe of H.264 discussed in the previous section. Our selective encryption algorithm encrypts a subset of the bitplanes of the parity bits and motion vector, respectively. The encrypted bitplanes are transmitted as plaintext. We assumed that an attacker does not have access to the encryption key.

Two parallel recursive systematic convolutional (RSC) encoders constitute the Turbo encoder. The quantized symbols are directly sent to one RSC encoder and sent to the other encoder after a random interleaver. We send a portion of the parity

Fig. 4.10. Original Video Sequence: YUV 4:1:1 sub-sampled with 176 × 144 pixels.

bits to the decoder through the channel, the systematic bits are completely discarded. At the decoder, a Turbo decoder decodes the parity bits with side information. The side information is an estimation of the current Wyner-Ziv frame obtained from the previously decoded frames.

Parity Bits Encryption

Turbo decoder uses this initial estimation and incoming parity bits to decode the current frame. If the initial estimation is coincident with the parity bits, the Turbo decoder works in the normal way. Otherwise, the decoder disregards the received parity bits and uses the quantization bin which is nearest to the estimation. Under the worst conditions, the maximum distortion is proportional to the coarseness of the quantizer. Thus, the scheme prevents the decoder from having large errors.

A rate compatible punctured turbo (RCPT) code is used and only a portion of the parity bits are sent to the Turbo decoder. At the decoder, the Turbo decoder reconstructed the frame with the side information derived from the N frames. If the reconstructed frame does not satisfy the requirement of the quality, the decoder requests more parity bits from the encoder. A conventional Wyner-Ziv video encoder encodes many frames as INTRA frames to guarantee low complexity encoding and accurate side information at the decoder. It does not make the best use of temporal

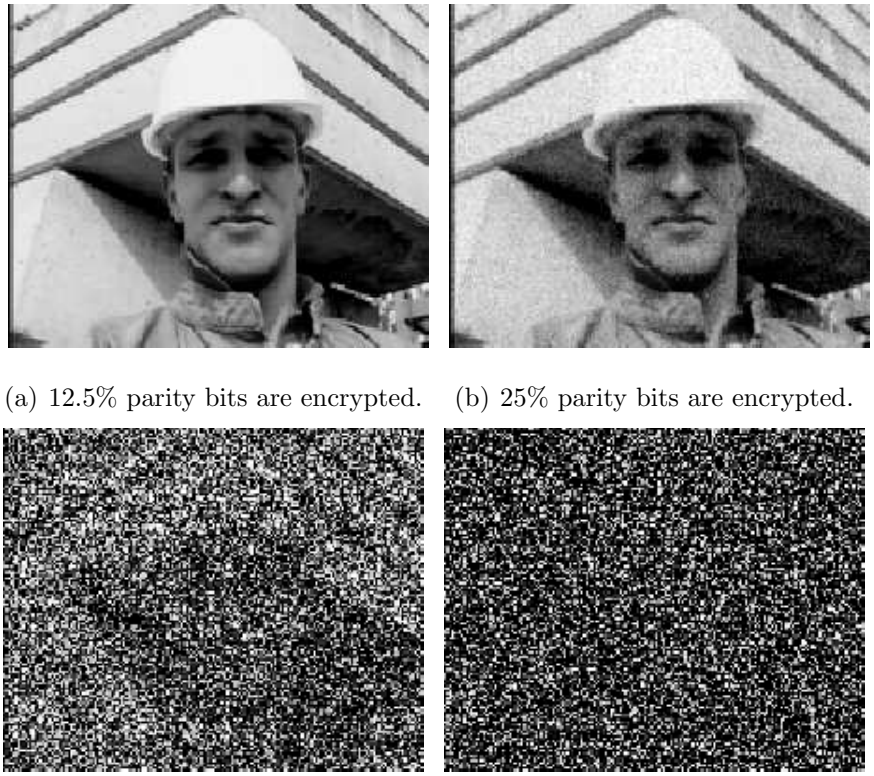
correlation across the frames, which is a major factor in conventional MCP-based video coding. We replace the I frames with N frames which can be regarded as pseudo P frames. Motion estimation is performed on the previous reconstructed frames at the decoder. The motion vectors for the current frame are predicted from the motion vectors of the previous frames and sent back to the encoder.

Table 4.2
Parity Bits Encryption.

Encryption Ratio	Total Bit Errors	PSNR(dB)
0%	378	31.6
12.5%	952	29.8
25%	1176	29.6
50%	16655	12.5
100%	16655	12.5

The simulation results of the parity bits encryption are shown in Table 4.2 and Figure 4.11. We also use Peak-Signal-to-Noise-Ratio (PSNR) for our measure of image quality. We assume that there is no channel error while sending the parity bits. In the case of encrypting the parity bits, the encryption of 50% or more bits reveals no useful information in the reconstructed frames. The PSNR decreases steadily from 32dB to 13dB as we encrypt more bits.

Figure 4.11 shows a reconstructed video video after encrypting only a part of the parity bits. When increasing the amount of encrypted packet data steadily, we finally result in 50% percent of the packet data encrypted where neither useful visual nor textual information in the image. In encrypting parity bitstream, the encrypted bitstream values are arithmetically decoded and the corresponding decoded bits depend on earlier results and corrupt the subsequently required decoding states. Therefore, the reconstruction video is a noise-like pattern.



(a) 12.5% parity bits are encrypted. (b) 25% parity bits are encrypted.

(c) 50% parity bits are encrypted. (d) 100% parity bits are encrypted

Fig. 4.11. Visual examples of the selective encryption when the parity bits are encrypted.

Motion Vector Encryption

We implemented the Huffman code and the Huffman-based symmetrical Reversible Variable Length Code (RVLC) to reduce the bandwidth of the backward channel. The Reversible Variable Length Code (RVLC) is utilized in the MPEG-4 error resilient tools for the recovery of the residual coefficients [113].

In order to decode the motion data, the encoder needs to have the codeword table. An entire sequence can use a static codeword table. However, experimental results show that updating the codeword table every frame improves the adaptivity and the coding efficiency. To reduce the data rate spent on the codeword table, we exploit an alternative approach by sending the cost table. The approach is based

Table 4.3
Motion Vector Encryption.

Encryption Ratio	Total Bit Errors	PSNR(dB)
0%	378	31.6
12.5%	887	30.5
25%	8515	11.4
50%	14351	9.12
100%	16477	3.68

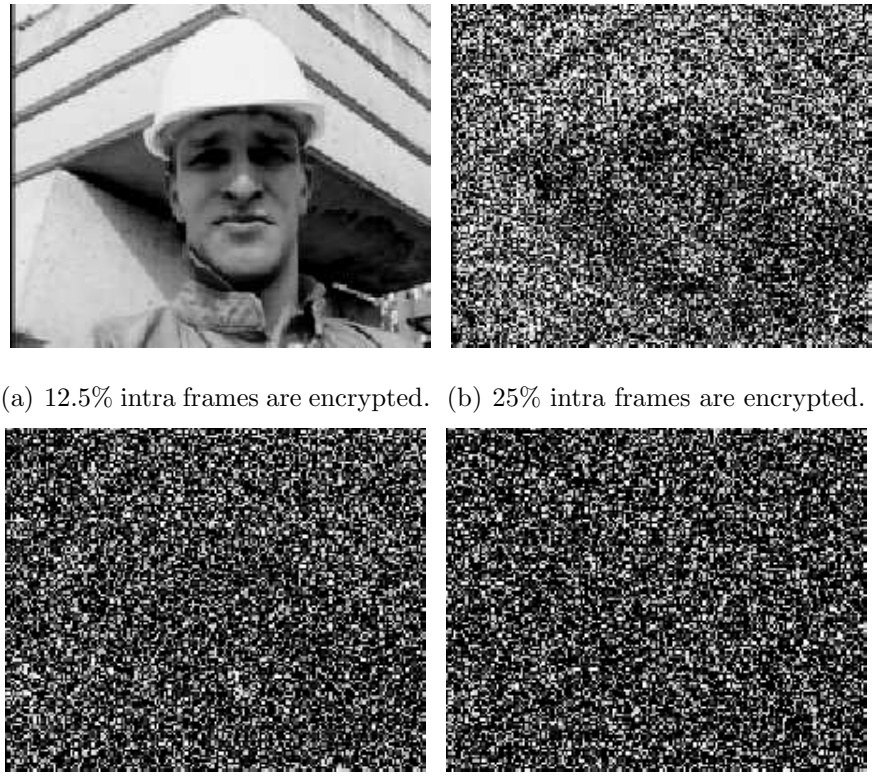
on the assumption that both the encoder and the decoder are capable of building identical binary trees for the codeword tables. To reduce the data rate further, we non-uniformly quantize the costs of the motion vectors to the power of 2 and send the logarithm of the costs.

In the case of encrypting the motion vector bits, the encryption of 25% or more bits reveals no useful information in the reconstructed frames. The PSNR decreases steadily from 32dB to 4dB as we encrypt more bits. Hence the motion vector is more critical data than the parity bits. The reason is that the conventional video coder is treated as the base layer and the bitstream from the Wyner-Ziv video coder is considered as the enhancement layer. That is, we can recover the reliable video frame in the decoder even though parity bits are not available with one reference frame.

The simulation results of the motion vector encryption are shown in Table 4.3 and Figure 4.12. Figure 4.12 shows a reconstructed frame video after encrypting only a portion of the intra frames.

4.3 Security Evaluation

The security of an entire image and video encryption approach has two aspects. First, the security of the cipher in use itself. Second, the importance and suitability



(a) 12.5% intra frames are encrypted. (b) 25% intra frames are encrypted.

(c) 50% intra frames are encrypted. (d) 100% intra frame are encrypted

Fig. 4.12. Visual examples of the selective encryption when the motion vectors (N frames) are encrypted.

of the data subject to encryption. The security is rated as low, medium, and high. It may have the additional property of being scalable if depending on the amount of data encrypted. In accordance to the two aspects of security, two entirely different types of attacks against image and video encryption are possible. On the other hand, the cipher in use is the target of an attack. In this case, common cryptanalytic results about the security of the specific ciphers in general apply. On the other hand, in the case of partial or selective encryption, it is possible to reconstruct the visual content without taking care of the encrypted parts. Depending on the importance of the encrypted data for visual perception, the result may range from entirely incomprehensible to just poor or reduced quality. In any case, when conducting the

direct reconstruction, the high frequency noise originating from the encrypted portions of the data is propagated into the reconstructed frame. In order to avoid this phenomenon, error-concealment attacks [87], perceptual attack [84], or replacement attacks [114, 115] have been proposed. These types of attacks either try to conceal the quality reduction caused by encryption by treating unbreakable data as lost and then trying to minimize the impact on quality as a result of loss (error-concealment attack) or simply replace the encrypted parts of the data by either artificial data mimicking simple non-structured content (replacement attack) or data minimizing the perceptual impact of the incorrect data (perceptual attacks) [12].

4.3.1 Replacement attack

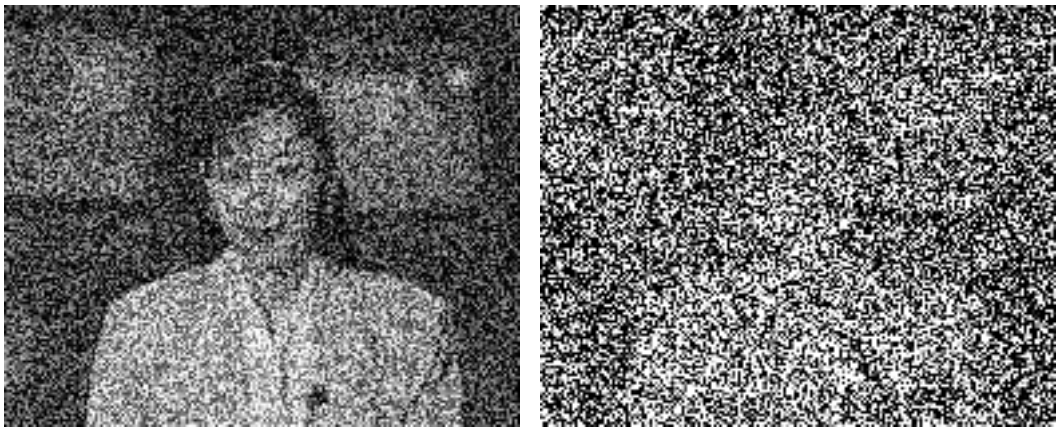
Decoding a partially encrypted image by treating the encrypted data as being unencrypted leads to images severely degraded be noise type patterns. Using these images to judge the security of the system leads to misinterpretations since a hostile attacker can de much better. In particular, an attacker could simply ignore the encrypted part, which can be easily identified by statistical means, or replace them by typical non-noise data [12, 87, 116].

Figures 4.5, 4.6, 4.11, and 4.12 clearly show that there can be still information left in the unencrypted parts of the data after selective encryption has been applied because in case of direct reconstruction this is hidden by the high frequency noise pattern. As a consequence, SE is evaluated after a replacement attack has been mounted.

Figures 4.5 shows 4 examples of directly reconstructed images after selectively encrypting 1, 2, 4, and 8 bitplanes. Whereas in case of encrypting the MSB only structural information is still visible, encrypting two or more bitplanes leaves no useful information in the reconstruction, at least when the parity bits are all encrypted and directly reconstructing the video data. In the following we access the security of selective bitplane encryption by considering two types of simple ciphertext only

attacks. A defect of many investigations of visual data encryption is the lack of qualifying the quality of the visual data that can be obtained by attacks against encryption. The reason is the poor correlation of PSNR and other simple quality measures and perceived quality especially for low-quality images. For the most simple attack we may even relate the visual examples to meaningful numerical values [12].

Assuming the cipher in use is unbreakable. We conduct the first attack by directly reconstructing the selectively encrypted video data. The encrypted parts introduce noise-type distortions. Therefore, we replace the encrypted parts by artificial data mimicking typical images. The encrypted bit-plane is replaced by a constant 0 bit-plane and the resulting decrease in average luminance is compensated by adding 64 to each pixel if only the MSB was encrypted, 96 if the MSB and next bitplane has been encrypted, and so on. Subsequently, reconstruction is performed as usual, treating the encrypted and replaced parts as being non-encrypted [12].



(a) 25% bits are encrypted and replacement attacks are mounted, PSNR=11.23dB.

(b) 50% bits are encrypted and replacement attacks are mounted, PSNR=10.12dB.

Fig. 4.13. Visual examples for the efficiency of the replacement attack.

Figures 4.13 shows two visual image reconstructions as obtained by the Replacement attack (2 and 4 bitplanes are encrypted). Whereas a direct reconstruction of an image with 2 bitplanes encrypted suggests this setting to be safe with 13.23dB

quality as shown in Figures 4.5, the replacement attack reveals that structural information is still present in the reconstructed image with 12.23dB. However, the visual information become severely estranged. Clearly, not only the visual appearance but also the numerical PSNR values have been significantly improved by the replacement attack. In any case, even if a replacement attack is mounted, encrypting 50% bits leads to perfectly satisfying results.

For the simple case of this encryption technique, we assume that the MBS biplane to be encrypted first. The idea of the replacement attack is to reconstruct the MSB data with the aid of the unencrypted remaining data. We exploit the well known property that most regions of natural images are covered by areas with smoothly changing gray values except edges. In areas of this type, the MSBs of all pixels tend to automatically detect such areas we define 2×2 pixels search window in which all 16 possible combinations of MSB configurations are tested. In this test, a certain set of differences among the 4 pixel values is computed for each of the 16 MSB configurations. The smallest difference is selected out of the set differences and the corresponding configuration of the MSB bits in the search window is defined to be the reconstruction. However the complexity of this attack increases significantly if more bitplanes are encrypted and also the reliability of the result is drastically reduced. Therefore it seems that a relatively high amount of data needs to be encrypted to realize reasonable security [12].

4.3.2 Security of the encryption scheme

In [31], Shannon provided the first rigorous statistical treatment of secrecy. The idea is that an eavesdropper will learn nothing at all about the plaintext if the encoded bitstream is statistically independent of the source messages. There are two basic approaches to discussing the security of a cryptosystem: computational security and unconditional security. The computational security concerns the computational effort to break a cryptosystem. We might define a cryptosystem to be computation-

ally secure if the best algorithm for breaking it requires at least N operations, where N is some specified, very large number. The problem is that no known practical cryptosystem can be proved to be secure under this definition. In practice, a cryptosystem is called if the best known method of breaking the system requires an unreasonably large amount of computing time. A cryptosystem is defined to be unconditionally secure if it cannot be broken, even with infinite computational resources [117, 118].

In our encryption scheme, a pseudo-random bit generator (PRBG) is used like one-time pad. A one-time pad is a very simple yet completely unbreakable symmetric cipher. Symmetric means it uses the same key for encryption as for decryption. One-time pad achieves the perfect secrecy, where the plaintext and the key are both bitstreams of a specific length, and the ciphertext is constructed by taking the bitwise exclusive-or of the plaintext and the key. The practical difficulty of the one-time pad is that the key, which must be randomly generated and communicated over a secure and tamper-proof channel, must be as long as the plaintext in order to ensure perfect secrecy. The seed functions as a key, and the PRBG can be thought as a keystream generator for a stream cipher.

If the key is truly random, an xor-based one-time pad is perfectly secure against ciphertext-only cryptanalysis. This means an attacker can't compute the plaintext from the ciphertext without knowledge of the key, even via a brute force search of the space of all keys. Trying all possible keys doesn't help you at all, because all possible plaintexts are equally likely decryptions of the ciphertext. This result is true regardless of how few bits the key has or how much you know about the structure of the plaintext.

5. SECURE GROUP KEY MANAGEMENT SCHEMES

This is the second part of thesis. Secure group key management schemes in wireless networks are discussed from Chapter 5 to Chapter 6.

5.1 Previous Works: Secure Scalable Multicast of Multimedia Data

Multicast protocols require an access control mechanism such that only the authorized members can access group communications. Access control is usually achieved by encrypting the content with an encryption key. This key is known as the session key (SK) that is shared by all valid group members. Access control typically employs a tree of encryption keys to update and maintain the SK. Tree-based schemes [33,34] have advantages that include computation, communication, and storage resources for the user and the group manager. In such schemes, the group key should be changed periodically or after a user leaves or joins the service to prevent the leaving/joining user from accessing the future /prior communication. This is known as "forward message secrecy" and "backward message secrecy." Also key management schemes in multicasting should be "scalable." By scalable we mean that the overhead involved in key exchange, updates, data transmission, and encryption must not be dependent on the size of the multicast group. Moreover, addition or removal of a host from the group should not affect the other members. This is known as the "1 affects n" scalability rule [119].

5.1.1 Non-scalable (Unicast) Group Key Management Protocols

The simplest scheme is to encrypt data with different secret keys for each member and distribute them via unicasting or multicasting [120]. A simple scheme for rekeying

a group with n members has the key distribution center (KDC) assigning a secret key to each member of the group. In order to distribute the group key, the KDC encrypts it with each members's secret key. This operation generates a message $O(N)$ long which is then transmitted to the whole group via multicast. On receiving the group key from the appropriate segment of the message using its own secret key. Since encryption overhead increases with the number of group members, this protocol isn't scalable.

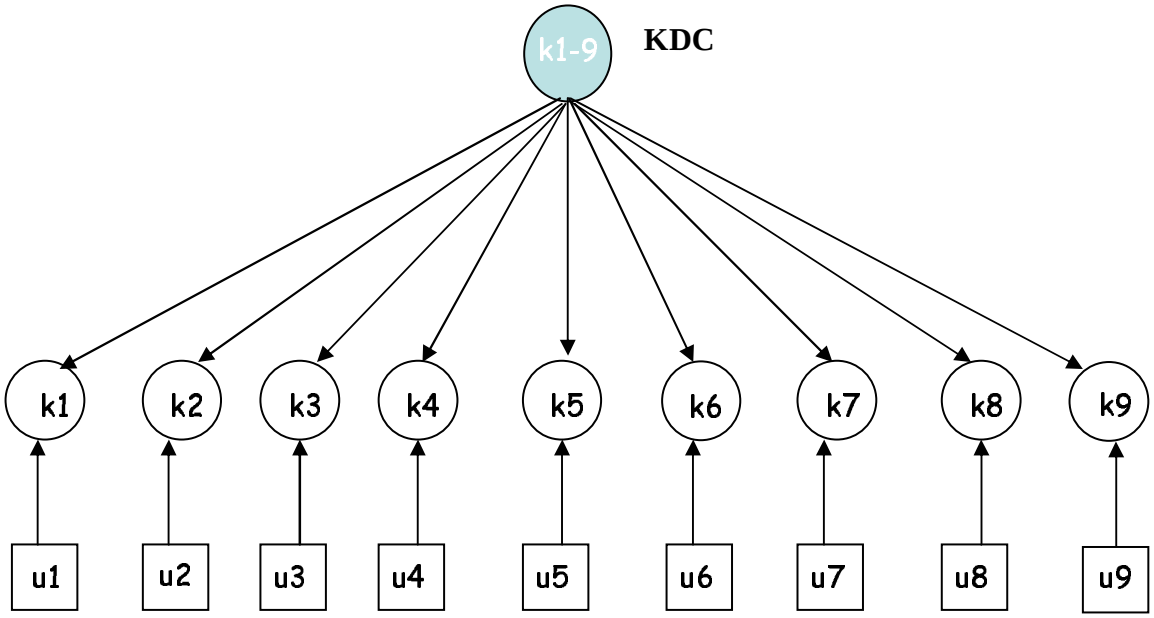


Fig. 5.1. Non-scalable Group Key Management.

5.1.2 Scalable Group Key Management Protocols

We can classify the scalable protocols into three main classes: centralized group key management protocols, decentralized architecture, and distributed key management protocols [41].

5.1.3 Centralized Group Key Management Protocols

A single entity is employed for controlling the whole group, hence a group key management protocol seeks to minimize storage requirements, computational power on both client and server sides, and bandwidth utilization. However, with only one managing entity, the central server may be a single point of failure. The entire group will be affected if there is a problem with the controller. The group privacy is dependent on the successful functioning of the single group controller [40]; when the controller is not working, the group becomes vulnerable because the keys, which are the base for the group privacy, are not being generated/regenerated and distributed. Furthermore, the group may become too large to be managed by a single group, thus raising the issue of scalability.

Several contributions propose the use of a Logical Key Hierarchy (LKH) [34, 121]. In this approach, a KDC maintains a tree of keys. The nodes of the tree hold key encryption keys (KEK). The leaves of the logical key tree correspond to group members and each leaf holds a KEK associated with that one member. Each member receives and maintains a copy of the KEK associated with its leaf and the KEKs corresponding to each node in the path from its parent leaf to the root. The key held by the root of the tree is the group key. For a balanced tree, each member stores at most $(log_2 N) + 1$ keys, where $(log_2 N)$ is the height of the tree and N is the group size.

For example, as shown in Figure 5.2, suppose u_9 is granted to join the upper key graph in the figure. The joining point is k-node k_{78} in the key graph, and the key of this k-node is changed to k_{78} in the new key graph below. Moreover, the group key at the root is changed from k_{1-8} to k_{1-9}. Users $u_1, \cdots, u_6$ only need the new group key k_{1-9}, while users u_7, u_8, and u_9 need new group key k_{1-9} as well as the new key k_{789} to be shared by them.

After granting a leave request from user u, server s updates the key graph by deleting the u-node for user u and the k-node for its individual key from the key

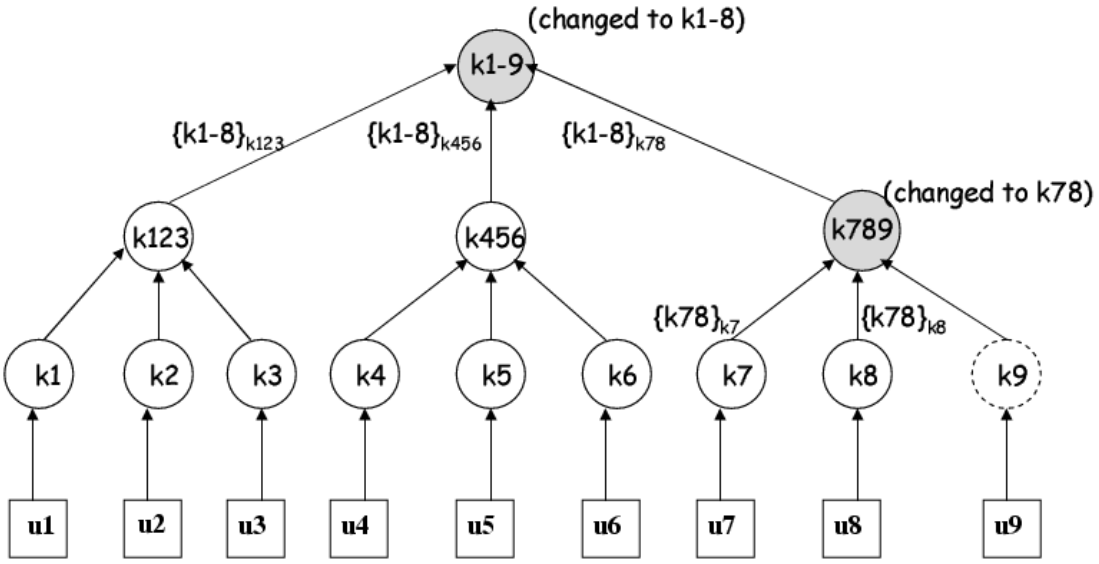


Fig. 5.2. An Example of Logical Key Hierarchy.

graph. The parent of the k-node for its individual key is called the leaving point. To prevent the leaving user from accessing future communications, all keys along the path from the leaving point to the root node need to be changed. After generating new keys for these k-nodes, server s needs to securely distribute them to the remaining users. For example, as shown in Figure 5.2, suppose u_9 is granted to leave the lower key graph in the figure. The leaving point is the k-node for k_{789} in the key graph, and the key of this k-node is changed to k_{78} in the new key graph above. Moreover, the group key is also changed from k_{1-9} to k_{1-8}. Users $u_1, \cdots, u_6$ only need the new group key k_{1-8}, while users u_7, u_8, and u_9 need new group key k_{1-8} as well as the new key k_{78} to be shared by them.

The algorithm proposed by Waldvogel [122] is different for joining operations. Instead of generating fresh keys and sending them to members already in the group, all keys affected by the membership change are passed through a one-way function. Every member that already knew the old key can calculate the new one. Hence, the new keys do not need to be sent and every member can calculate them locally [40].

The efficient large-group key (ELK) protocol is proposed by Perrig [123]. The ELK protocol uses a hierarchical tree and is similar to one-way function tree (OFM) [124] in the sense that a parent node key is generated from its children keys. ELK uses pseudo-random functions (PRFs) to build and manipulate the keys in the hierarchical tree. A PRF uses a key K on the input M of length m to generate output of length n. Using the PRF on a key, it is possible to drive four different keys to be used in the different contexts. ELK employs a timely rekey, which means that the key tree completely updated in each time of interval. ELK also introduce the idea of *hints*. A hint is a piece of information, which is smaller than a key update message, that can be used to recover possible lost rekey message updates. It is provided to improve the reliability of the rekey operation and it is conveyed in data messages [40].

5.1.4 Decentralized Group Key Management Protocols

In the decentralized subgroup approach [33, 44, 125], the large group is split into small subgroups. Different controllers are used to manage each subgroup, minimizing the problem of concentrating the work on a single place. In this approach, more entities are allowed to fail before the whole group is affected [40].

Mittra proposes a decentralized group key management scheme, Iolus [33], which is a framework with a hierarchy of agents that splits the large group into small subgroups. There are the group security controller which manages the top-level subgroup and the group security intermediaries (GSIs), one per subgroup, which manage each of the other subgroups. Generically they are called group security agents (GSAs). A Group Security Agent (GSA) manages each subgroup. The GSAs are also grouped in a top-level group that is managed by a Group Security Controller (GSC).

Iolus uses independent keys for each subgroup and the absence of a general group key means membership changes in a subgroup are treated locally. It means that any changes that affect a subgroup are not reflected in other subgroups. In addition, the

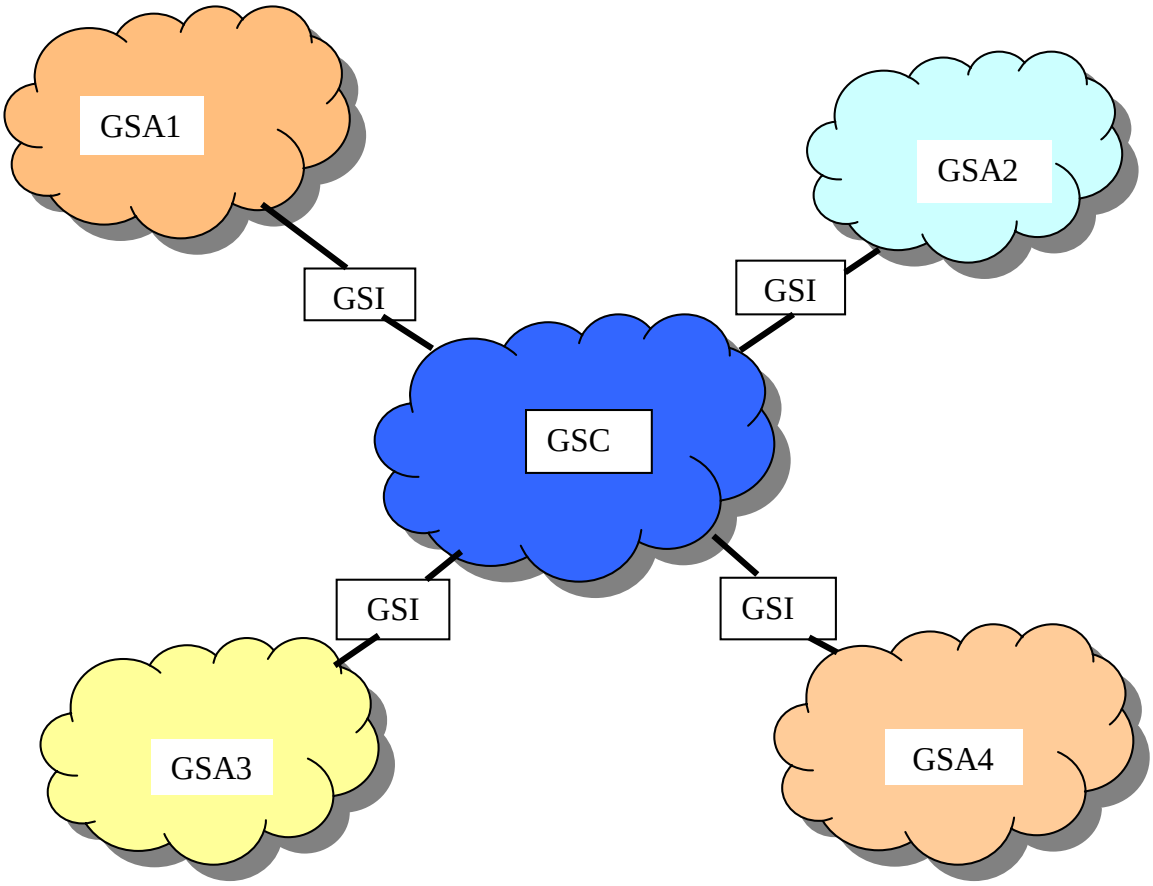


Fig. 5.3. Example of a Secure Distribution Tree.

absence of a central controller contributes to the fault-tolerance of the system. If a subgroup controller (namely GSA) fails, only its subgroup is affected.

Although Iolus is scalable, it has the drawback of affecting the data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup, and thereby one key, to another. This becomes even more problematic when it is taken into account that the GSA has to manage the subgroup and perform the translations needed. The GSA may thus become a bottleneck.

More specifically, the GSAs form a hierarchy of subgroups as shown in Figure 5.3. The GSC maintains control of the top-level subgroup at the root of the secure distribution tree. It is ultimately responsible for the security of the entire group. GSIs

are special trusted servers that are authorized to act as proxies of the GSC or their parent GSIs and control their local subgroup. The GSIs are grouped according to levels within the secure distribution tree. GSIs at a given level join the subgroups of the GSI at the next higher level or the subgroup of the GSC. They form a bridge between subgroups by receiving data multicast in their parent or child subgroups and re-multicasting to their child or parent subgroups respectively.

The dual encryption protocol (DEP) [41] also uses hierarchical subgrouping of multicast members to address scalability. Each group is managed by a subgroup manager (SGM) which assists in key distribution as well as group access control. It distinguishes between participants and members of the multicast group. Members of multicast group are leaf nodes and internal nodes (SGMs) in the key distribution tree, that are entitled to the multicast data. On the other hand, participants of the multicast group are SGMs that assist in enforcing the secure multicast protocol without having any access to the multicast data. The dual encryption scheme enables DEP to hide multicast data from the participants. It consists of two different keys to encrypt sensitive data. The top level key encrypting keys are used to securely propagate data encryption keys through the hierarchical subgroups. The other set of keys are local subgroup keys that are used by SGMs to distribute the encrypted data encryption keys to the corresponding subgroup members. Only the hosts with both the corresponding key encryption key and the local subgroup key can decrypt the data encryption keys.

Rafaeli proposed Hydra [125]. In Hydra, the large group is dived into smaller subgroups, and a server called the Hydra Server (HS) controls each subgroup. Hydra is a decentralized group key management scheme without a central subgroup controller. If a membership change takes place at HS_i, and a new key must be generated, it can generate the new group key and send this key to the other HS_j involved in that session. The case when one or more HS_s become unavailable will not cause a problem for the remaining HS_s. In order to have the group key distributed to all HS_s a synchronized group key distribution protocol (SGKDP) is employed. The SGKDP

protocol ensures that only a single valid HS is generating the new group key at every given time [40].

5.1.5 Distributed Group Key Management Protocols

In this scheme, there is no explicit KDC and the members themselves do the key generation. All members can perform access control and the generation of the key can be either contributory, meaning that all members contribute some information to generate the group key, or done by one of the members.

The distributed key management approach is characterized by having no group controller. The group key can be either generated in a contributory fashion, where all members contribute their own share to computation of the group key, or generated by one member. In the latter case, although it is fault-tolerant, it may not be safe to leave any member to generate new keys since key generation requires secure mechanisms, such as random number generators, that may not be available to all members. Moreover, in most contributory protocols (apart from tree-based approaches), processing time and communication requirements increase linearly in term of the number of members. Additionally, contributory protocols require each user to be aware of the group membership list to make sure that the protocols are robust [40, 126–131].

Diffie-Hellman (DH) group key exchange [132] is an extension for the DH key agreement protocol that supports group operations. The DH protocol is used for two parties to agree on a common key. In this protocol, instead of two entities, the group may have n members. The group agrees on a pair of primes, p and α, and starts calculating in a distributive fashion the intermediate values. The first member calculates the first value α^{x1} and passes it to the next member. Each subsequent member receives the set of intermediary values and raises them using its own secret number generating a new set. A set generated by the i^{th} member will have i intermediate values with $i-1$ exponents and a cardinal value containing all exponents.

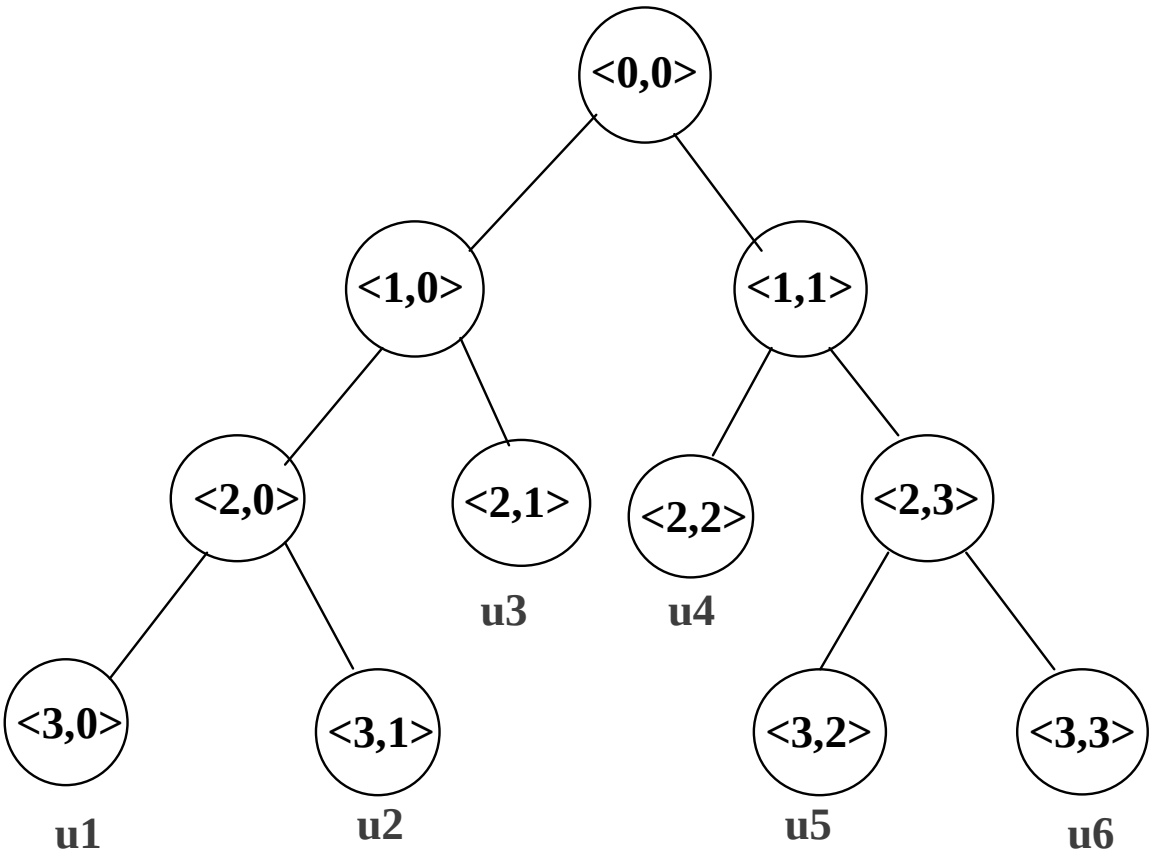


Fig. 5.4. A Key Tree of TGDH.

Tree-based Group Diffie-Hellman (TGDH) is shown in Figure 5.4 [133–135]. The root is located at level 0 and the lowest leaves are at level h. Since we use binary trees, 2 every node is either a leaf or a parent of two nodes. TGDH is an adaptation of key tree in the context of fully distributed, contributory group key agreement. TGDH computes a group key derived from the contribution of all group members using a binary tree. The tree is organized in the following manner: each node $< l, v >$ is associated with a key $K_{<l,v>}$ and the corresponding blinded key $BK_{<l,v>} = \alpha^{K<l,v>} mod(p)$. The key at the root node is the group key shared by all members, and a key at the leaf node is the random session contribution by a group member. Each member knows

all the keys on the path from its leaf node to the root as well as blinded keys on the key tree.

Assuming a leaf node $< l, v >$ hosts the member u_i, the node $< l, v >$ has $u_i^{'}s$ session random key $K_{<l,v>}$. Furthermore, the member u_i at node $< l, v >$ knows every key along the path from$< l, v >$ to $< 0, 0 >$, referred to as the key-path and denoted KEY_i^*. In Figure 5.4, if a member u_2 owns the tree T_2, then u_2 knows every key $\{K_{<3,1>}, K_{<2,0>}, K_{<1,0>}, K_{<0,0>}\}$ in KEY_i^*=$\{< 3,1 >, < 2,0 >, < 1,0 >, < 0,0 >\}$ and every blinded key BK_2^*=$\{BK_{<0,0>}, BK_{<1,0>}, BK_{<1,1>}, \cdots, BK_{<3,7>}\}$ on T_2. Every key $K_{<l,v>}$ is computed recursively as follows:

$$\begin{aligned} K_{<l,v>} &= (BK_{<l+1,2v+1>})^{K_{<l+1,2v>}} \mod (p) \\ &= (BK_{<l+1,2v>})^{K_{<l+1,2v+1>}} \mod (p) \\ &= \alpha^{K_{<l+1,2v>}K_{<l+1,2v+1>}} \mod (p) \end{aligned} \tag{5.1}$$

In other words, computing a key at $< l, v >$ requires the knowledge of the key of one of the two child nodes and the blinded key of the other child node. $K_{<0,0>}$ at the root node is the group secret shared by all members. We note that this value is never used as a cryptographic key for the purposes of encryption, authentication or integrity. Instead, such special purpose sub-keys are derived from the group secret, e.g., by setting $K_{group} = h(K_{<0,0>})$ where h is a cryptographically strong hash function. For example, in Figure 5.4, u_2 can compute $K_{<2,0>}, K_{<1,0>}$ and $K_{<0,0>}$ using $BK_{<3,0>}$, $BK_{<2,1>}$, $K_{<1,1>}$, and $BK_{<3,1>}$ [136].

The basic idea here is that every member can compute a group key when all blinded keys on the key tree are known. After any group membership event, every member unambiguously adds or removes some nodes related with the event, and invalidates all keys and blinded keys related with the affected nodes. A special group member, the *sponsor*, then takes on a role to compute keys and blinded keys and to broadcast the key tree to the group. If a sponsor could not compute the group key, then the next sponsor will compute comes into play. Eventually, some sponsor will

compute the group key and all blinded keys, and broadcast the entire key tree to facilitate the computation of the group key by the other members of the group.

Burmester et. al. introduced the first n-party key agreement protocol (BD) based on a tree [137]. They described protocols based on star, tree, broadcast and cyclic topologies. McGrew et. al. [124] used a one-way function tree (OFT), which is a binary tree, for the exchange of key information. Kim et. al. [136] investigated a number of different tree structures for group key agreement which are efficient with respect to a number of group operations such as member add, member delete, group merge and group partition. Becker et. al. [134] demonstrated 2 protocols, namely Hypercube and Octopus, which achieve the lower bounds on the round complexity and the number of exchanged messages respectively. These key agreement schemes are fully distributed and do not need any server or trusted third party (TTP).

5.2 Summary of Previous Works

We summarize and compare the properties of those protocols presented in Section 2. Using logical key trees reduces the complexity of group rekeying operation from $O(N)$ to $O(logN)$, where N is the group size. LKH [34] is a very efficient and hence scalable protocol for group rekeying when compared to a unicast-based näive approach. Let N be the group size, d be the degree of the key tree, then the communication cost for rekeying is $O(log_d N)$, whereas the näive approach requires a communication cost of $O(N)$. For a large group with very dynamic memberships, LKH may not perform well because it performs a group rekeying for every membership change.

Iolus [33] deals with the scalability issue by partitioning the group members into many subgroups, which are arranged in a hierarchy to create a single multicast group. Scalability is achieved by making each subgroup relatively independent and thus group membership changes can be confined to the respective subgroups. Another essential element that helps Iolus to achieve its scalability is the subgroup agents,

which assist in translating messages among subgroups using different subgroup keys. While improving scalability, this approach introduces extra propagation delays and requires full trust in each subgroup agent. In brief, having subgroup agents decrypt and re-encrypt the data packets is a drawback, both from a performance point of view and from a security point of view.

TGDH [136] do not have a leader during setup time and all members compute the intermediary values independently. At the final round, all members compute the same group key. Any member failure can be ignored, because it does not block the other members. That is, it provide contributory group key agreement based on different extensions of the two-party Diffie-Hellman key exchange. Moreover, they all support dynamic membership operations. TGDH contributory key agreement protocol is robust and efficient in the sense that it can deal with network partition and that the number of rounds for rekeying is limited by $O(logN)$ where N is the number of members currently in the group. However, with this protocol during the period of rekeying which occurs whenever member(s) join or leave the group, all group members stop data communication and wait until the new group key is formed in a distributed manner. In case a rekeying packet is delayed or lost, the intervals (latency) and frequencies of interruptions may become annoying.

6. MOBILITY IMPACT OF A GROUP KEY MANAGEMENT SCHEME

In wireless networks, secure multicast protocols are more difficult to implement efficiently due to the user mobility. Mobility is one of the most distinct features to be considered in wireless networks. Moving users onto the key tree causes extra key management resources even though they are still in service, which is called *handoff*. To deal with frequent handoff between base stations, it is necessary to reduce the number of rekeying messages and the size of the messages.

6.1 Handoff Schemes

There are 2 types of handoff: a hard handoff and a soft handoff, as shown in Figure 6.1. In the hard handoff, the connection to the current cell is broken, and the connection to the new cell is made. This is known as a "break-before-make" handoff. The soft handoff refers to the overlapping of Base Station (BS) coverage zones, so that every cell phone is always well within range of at least one base station. In some cases, mobile sets transmit signals to, and receive signals from, more than one BS at a time. This is known as a "make-before-break" handoff.

We describe a soft handoff scheme based on the location of a user instead of the use of the strength of a pilot signal from the user to the BS, as shown in Figure 6.2. There are two important parameters, L_ADD and L_DROP. L_ADD and L_DROP indicate the beginning of handoff and the termination of handoff based on the location of the user. In general, the system administrator decides the values of two parameters. In our simulation, 30% of soft handoff area is used. That is, the L_ADD is the boundary of overlapping area of two BSs and the L_DROP is the middle of two BSs as shown in Figure 3.2. In this example, a MS moves from A of BS1 to B of BS2.

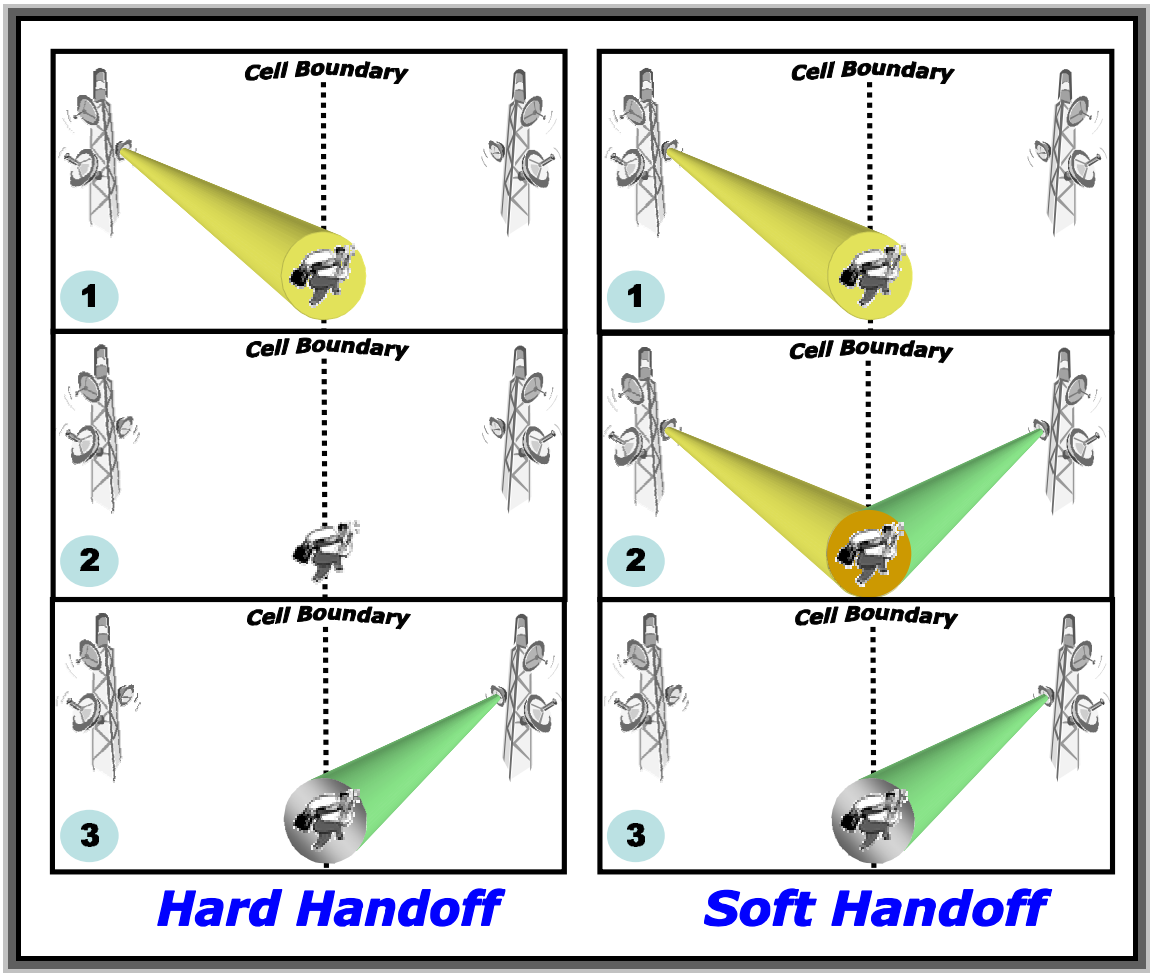


Fig. 6.1. Handoff Methods.

The Mobile Station (MS) requests a handoff to the neighboring BS when the location of neighboring BS exceeds the handoff threshold L_ADD. If the handoff request is accepted in the neighboring BS, BS2, the MS maintains two traffic channels assigned by the serving BS, BS1 and the neighboring BS. As the MS moves away from the serving BS and approaches the neighboring BS, the location of MS falls below the handoff drop threshold L_DROP for the servicing BS. If the location of the MS is close to the neighboring BS during the specific time interval, the traffic channel assigned by the serving BS is released, and the handoff is terminated.

In the case of hard handoff, MS requests a handoff to the neighboring BS immediately after exceeding the handoff threshold L_DROP. The moving MS does not maintain 2 traffic links in the handoff region.

The handoff-add threshold can be thought of as the "largest" distance between a MS and a BS such that the MS can reliably transmit information through the given BS. The handoff-drop threshold is the distance where the MS cannot communicate with the servicing BS any more. In general, the system administrator determines L_ADD and L_DROP to optimize wireless channel utilization. Each serving BS broadcasts this information.

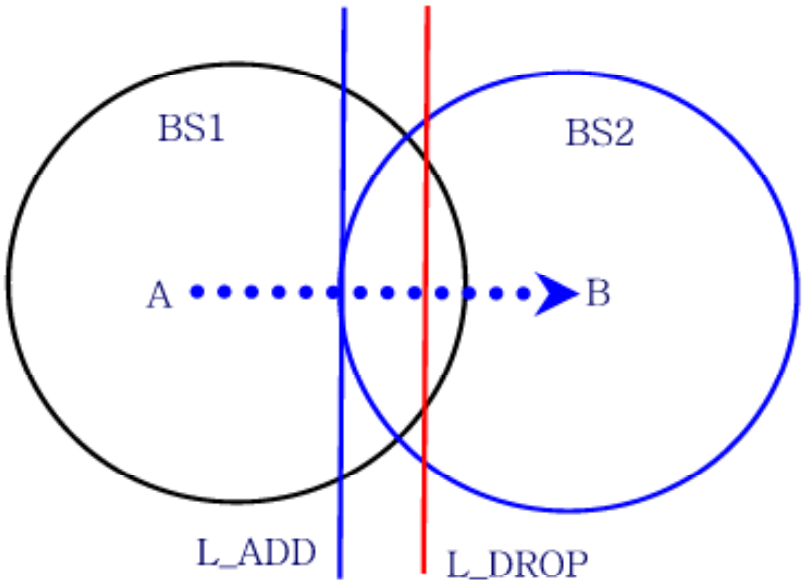


Fig. 6.2. An Example of L_DROP and L_ADD.

We propose a new handoff scheme to reduce the traffic of key updating during a handoff call. In the revised handoff scheme, two links are maintained during the handoff for the data transmission while the key update is only performed after completing the handoff. That is, the key updating does not occur when a call enters the handoff region. The connection to the new BS is just established without a key rekeying to prepare for the new connection. We can reduce the traffic of key update in handoff region. This is a variation of the soft handoff scheme.

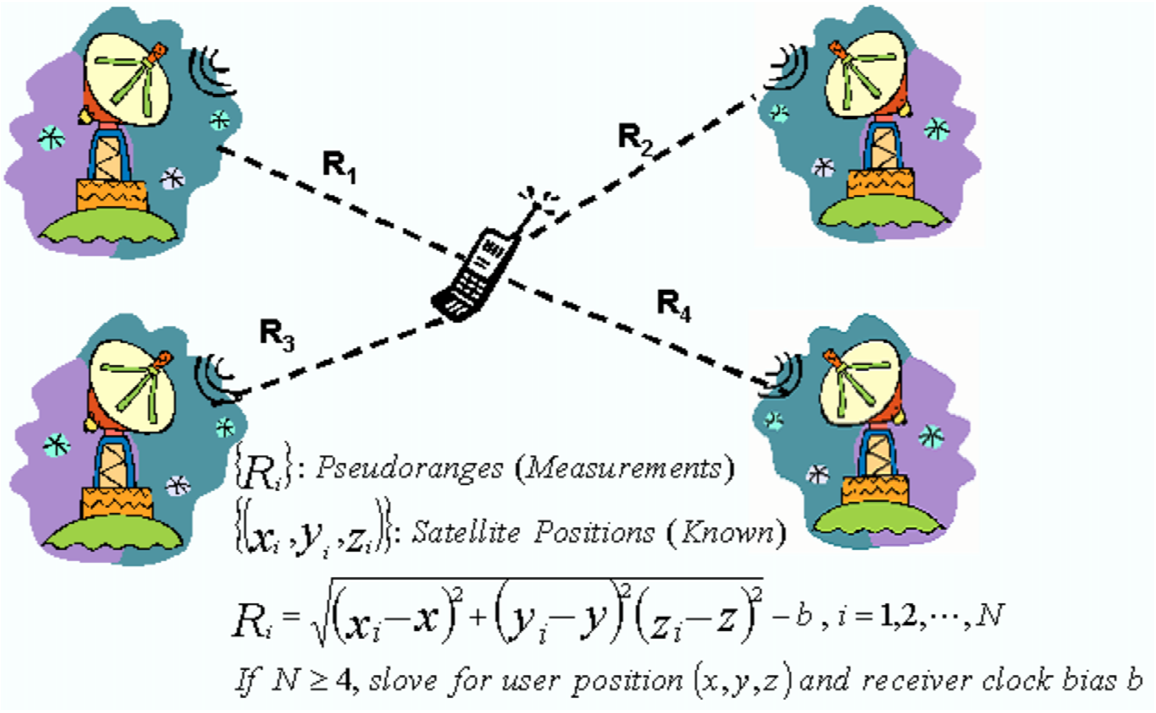


Fig. 6.3. The Principle of Location Tracking.

6.2 Location Tracking

We explain briefly about measuring the location of user in Code Division Multiple Access (CDMA) cellular system [2, 138]. The most widely known position location system is the Global Positioning System (GPS). The GPS is a satellite-based pseudo-ranging position location system that provides geolocation of user's with GPS receivers. The GPS is a proven technology that has found widespread use in military and navigation applications. It can reportedly provide position location accuracy's of less than 10 meters to military user's and 100 meters to commercial user's. A differential GPS (DGPS) has been developed that can improve the position location (PL) accuracy of the commercial operation. Global Positioning System (GPS) [139, 140] provides highly accurate positioning information. The idea behind GPS is that one's position (x,y,z) can be determined with the distance values from three different known

positions by the triangulation method. The distance is measured in terms of delay, where an accurate clock at the receiver measures the time delay between the signal leaving the satellite and arriving at the receiver. Four simultaneous delay measurements from four satellites are required to solve three unknowns and the user's clock offset as shown in Figure 6.3. Some proposals for positioning, using one or two satellites, were presented in [141,142] based on recently proposed mobile satellite systems.

CDMA2000 [2] is synchronized with the Universal Coordinated Time (UCT). The forward link transmission timing of all CDMA2000 base stations worldwide is synchronized within a few microseconds. Base station synchronization can be achieved through several techniques including self-synchronization, radio beep, or through satellite-based systems such as GPS, Galileo, or GLONASS. Reverse link timing is based on the received timing derived from the first multipath component used by the terminal.

6.3 Pre-positioned Secret Sharing (PSS)

We propose to use secret sharing techniques for the construction of the key trees. Secret sharing methods have been used for various security applications requiring users to share keys. We use the Pre-positioned Secret Sharing (PSS) scheme described in [143,144]. We already show in the previous work [145–147] that PSS based scheme is comparable to the tree-based schemes [33, 34] in the respect of communications cost, rekeying time cost, and memory cost in the wired network.

Shamir's secret sharing scheme [148] is a threshold scheme based on polynomial interpolation. It allows a dealer D to distribute a secret value s to n players, such that at least t players are required to reconstruct the secret. The protocol is information theoretically secure, i.e., any fewer than t players cannot gain any information about the secret by themselves.

Let's see how we can design an (n, t) secret sharing scheme. To make the presentation easy to understand, let's start with the design of an $(n, 2)$ scheme.

Let's say we want to share a secret s among n parties. We use some basic geometry (see Figure 6.4 below). Select the point $(0, s)$ on the Y axis that corresponds to the secret. Now, randomly draw a line that goes through this point. Pick n points on that line: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Each point that is picked represents a share. We claim that these n shares constitute an $(n, 2)$ sharing of s. Now we need to show that this scheme satisfies both the availability and confidentiality properties.

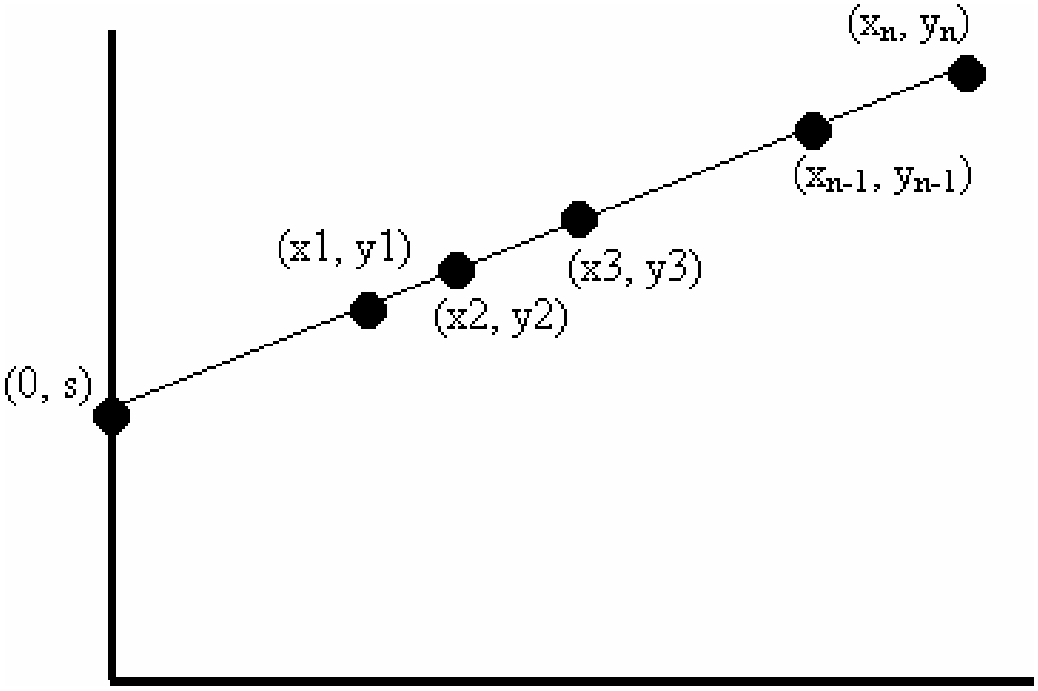


Fig. 6.4. (n,2) Secret Sharing Scheme.

To show availability, we need to prove that two parties can recover the secret. Two parties have two shares; that is two points. Given these two points, how can we recover the secret? Well, we know that two points determine a line, so we can figure out the line that goes through both points. Once we know the line, we know the intersection of the line with the y axis. Then, we get the secret. So, it only takes us two points (shares) to make the secret available.

What about confidentiality? We need to show that one share does not disclose any information about the secret. There are infinite possible lines that go through

this point, and these lines intersect with the y-axis at different points, all of which yield different "secrets". In fact, given any possible secret, we can draw a line that goes through the secret and the given share. This means that with one point, no information about the secret is exposed.

Using the same idea, can we design an $(n, 3)$ secret sharing scheme? Note that the key point in the $(n, 2)$ scheme is that a line is determined by two points, but not by 1. Now we need a curve that is determined by three points, but not 2. This curve happens to correspond to a quadratic function $y = a_2 * x_2 + a_1 * x + a_0$. Again, we find the point on the y-axis that corresponds to the secret, then we randomly select a curve corresponding to a quadratic function that goes through the point. Finally, we select n points on that curve as n shares to n parties (see Figure 6.5 below). Using a similar proof as in the $(n, 2)$ case, we can show that this is actually an $(n, 3)$ scheme that satisfies both availability and confidentiality [149].

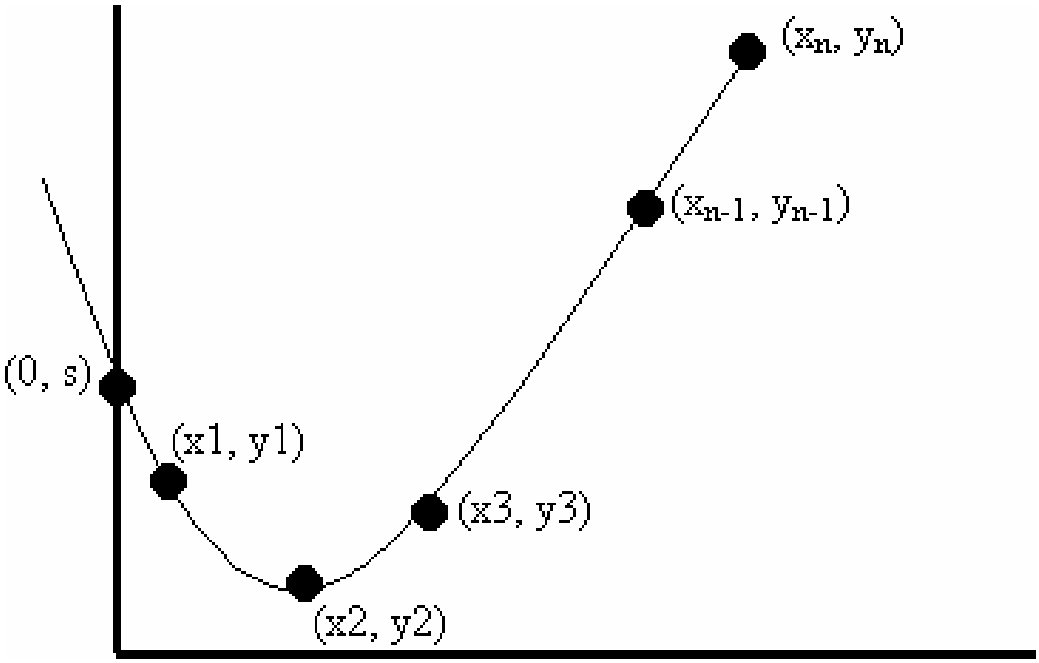


Fig. 6.5. (n,3) Secret Sharing Scheme.

To generalize the scheme even further, we have a construction of an (n, t) secret sharing scheme. Now we use the curve that corresponds to a $(t-1)$ degree polynomial:

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \mod (q) \tag{6.1}$$

We randomly select a curve corresponding to such a polynomial that goes through the secret on the y-axis. And then we select n points on the curve. Using the same arguments, we can show that this scheme satisfies both availability and confidentiality properties.

To reconstruct the secret from each subset of t shares out of n shares, we use interpolation property and Lagrange interpolation. Given distinct t pairs of $(i, f(i))$, there is a unique polynomial $f(x)$ of degree $t-1$, passing through all the points. This polynomial can be effectively computed from the pairs $(i, f(i))$. Without loss of generality we will mark this subset:$f(1), \cdots, f(t)$. We use Lagrange interpolation to find the unique polynomial $f(x)$ such that $degree f(x) < t$ and $f(j) = share_j(s)$ for $j = 1, 2, \cdots, t$, where $share_j(s) = (x_i, f(x_i))$, $i = 1, 2, \cdots, n$.

$$f(x) = \sum_{j=1}^{t} f(x_j) \times L_j(x), L_j(x) = \prod_{i \neq j, 1 \leq i \leq t} \frac{(x - x_i)}{(x_j - x_i)} \tag{6.2}$$

where, $L_j(x)$ is the Lagrange polynomial which has value 1 at x_i, and 0 at every other x_j. Then we can reconstruct the secret to be $f(0)$.

PSS uses a polynomial of order $(t-1)$ to generate shares. The shares will be used to generate the keys for the key tree. PSS is an interpolating scheme based on polynomial interpolation like Shamir's secret sharing scheme [148]. An $(t-1)$-degree polynomial over the finite field $GF(q)$

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \mod (q) \tag{6.3}$$

is constructed such that the coefficient a_0 is the secret and all other coefficients are random elements in the field. Each of the n shares is a point (x_i, y_i) on the curve defined by the polynomial, where x_i is not equal to 0. Given any m shares,

the polynomial is determined uniquely and hence the secret a_0 can be computed. However, given $t-1$ or fewer shares, the secret can be any element in the field. Therefore, PSS is a perfect secret sharing scheme. PSS uses a tree structure, which is composed of user nodes, subgroup-manager nodes, and the group-manager node in a bottom-up order. In the PSS, $(t-1)$ shares are assigned to each node while the t^{th} share is broadcasted as publication information. The $(t-1)$ shares of a node, which are secret, are referred to as the pre-positioned shares, while the broadcast share, is referred to as the activation share (AS). In PSS, the AS helps determine the symmetric keys for each node. Once a node obtains the AS, the original polynomial of order m can be reconstructed and hence the keys can be recovered, using the AS along with the private $(t-1)$ shares owned by the node.

6.4 Group Key Management

We design a key management tree such that the key tree matches the network topology. We localize the delivery of rekeying messages to small regions of network by transmitting the key update messages only to the users who need them. This lessens the amount of traffic in wireless and wired intervals.

We explain the group key management operations, join, leave and handoff, through the example as shown in Figure 6.6 and Figure 6.7. For each join, leave, and handoff, the shares will be changed to prevent the joining user from accessing past/future communications. After each join or leave, a new secure group is formed. The key server has to update the group's key graph by replacing the keys of some existing k-nodes, deleting some k-nodes and adding some k-nodes. Only one activating share is multicast by the key server, and it is used together with the pre-positioned information to generate three simultaneous keys.

In this example, 1 Group Manager (GM), 2 Subgroup Managers (SGM) and 6 users are considered. In Handoff operations, a 2 inter-BS handoff scheme is used for simplicity even though there are many handoff schemes [138].

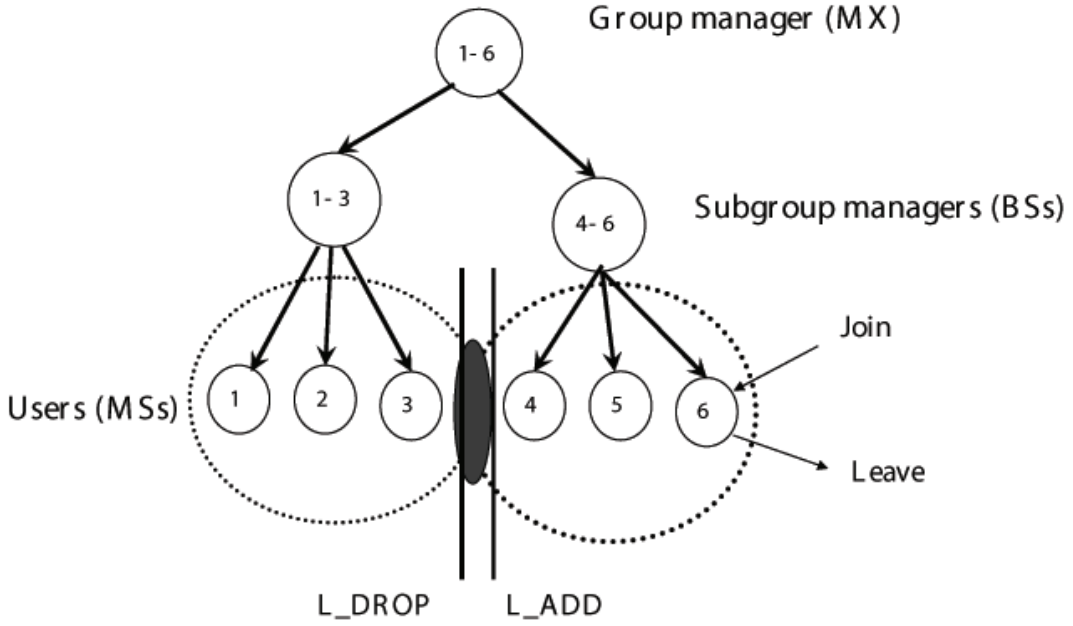


Fig. 6.6. Hierarchical Tree for Join/Leave.

6.4.1 Joining a Group via BS1

For example, as shown in Figure 6.6, suppose user 6 wants to join the secure group. To prevent the joining user from accessing past communications, all keys along the path from the joining point to the root node need to be changed.

User 6 sends a join request message to the key server. After granting the new user, the key server associates s_6 with the new member and creates a new node and a new set node. The key server attaches the set node to the existing joining point. After changing s_{1-5} to s_{1-6} and s_{4-5} to s_{4-6}, the key server constructs the following two messages:

1. AS,$\{s_{1-6}\}_{k1-5}$, $\{s_{4-6}\}_{k4-5}$

2. AS,$\{s_{1-6}, s_{4-6}\}_{k1-6}$

where AS is the activating share, the fresh keys k_{1-5}, k_{4-5} and k_6 are obtained by AS and the sets s_{1-5}, s_{4-5}, and s_6, respectively. The key server multicast the first message to the existing members, through $1-5$, while it unicast the second to the new member, 6. The members construct the new set of group keys, $k_{1-6}^{'}$, when the new AS is multicast with the encrypted content.

6.4.2 Leaving a Group via BS1

Now suppose user 6 wants to leave the secure group, as shown in Figure 6.6. To keep the leaving user from accessing future communications, all keys along the path from the leaving point to the root node need to be changed.

User 6 sends a leaving request message to the key server. After granting the leaving user, the key server deletes the member node and the set node from the key tree. The key server replaces s_{4-6} by s_{4-5} and s_{1-6} by s_{1-5}. Then it constructs the following messages and multicast to the remaining members:

1. $\{s_{1-5}\}_{k1-3}, \{s_{1-5}\}_{k4-5}$
2. $\{s_{4-5}\}_{k4}, \{s_{4-5}\}_{k5}$
3. AS

6.4.3 Handoff

As shown in Figure 6.7, user 4 is moving from BS2 to BS1 while the user is in the group service. The serving subgroup manager, BS2, requests a new connection to the neighboring BS, BS1, when the moving user exceeds the handoff add threshold, L_ADD. The key server associates s_4 with the new member of BS1, and creates a temporary node and a new set node. These sets are used within the handoff area. The key server attaches the set node to the existing joining point. After changing s_{1-3} to s_{1-4}, it constructs the following two messages:

1. AS, $\{s_{1-4}\}_{k1-3}$

2. AS, $\{s_{1-4}\}_{k1-6}$

The key server multicasts the first message to the existing member of BS1 while it unicasts the second message to the handoff member. Thus the handoff user keeps two links until it exceeds the handoff drop threshold, L_DROP. Immediately after the handoff user exceeds the L_DROP, the key server performs the leave procedure for BS2 and the add one for BS1.

The key server deletes the member node, here 4, and the set node from the key tree. The key server replaces s_{4-6} by s_{5-6}. Then it constructs the following messages and multicasts to the remaining members:

1. $\{s_{1-6}\}_{k1-4}$, $\{s_{1-6}\}_{k4-5}$

2. $\{s_{4-5}\}_{k4}$, $\{s_{4-5}\}_{k5}$

3. AS

In the case of hard handoff, the leave and join operations are taken immediately after the moving user exceeds the boundary of the serving BS. That is, we can consider the hard handoff user as a leaving and a joining user to the group service. In this case, the handoff user does not keep two links in the handoff region. This is the main difference between the soft handoff and the hard handoff operations.

Neither handoff schemes are practical for cellular networks with frequent handoffs because the extra communication cost is too high if the system does not limit the number of group members. Thus the system manager uses a resource management scheme, CAC function, in real system.

6.5 Simulations and Results

First, three measures are used to compare logical key hierarchical (LKH) based schemes [34, 121] and PSS: Storage cost, communication cost and computational cost [5, 145].

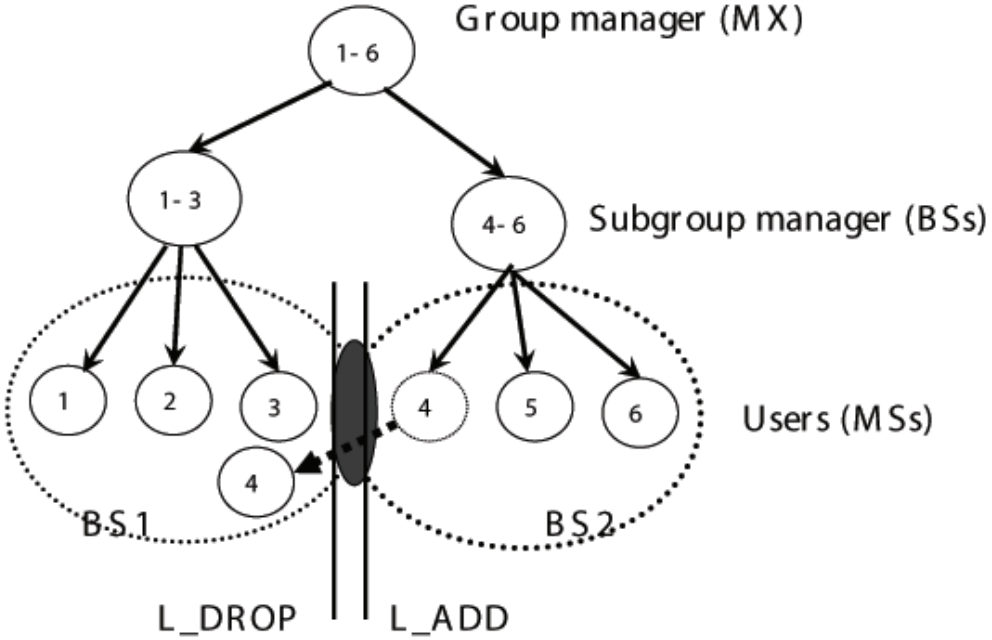


Fig. 6.7. Hierarchical Tree for Handoff.

6.5.1 Comparison of LKH and PSS schemes

The group key tree is assumed full and balanced. The height h of the tree is the length of the longest directed path in the tree, and the degree d of the tree is the maximum number of incoming edges of a node in the tree. The observations are summarized in the following Tables.

The number of encryptions and decryptions required by join/leave operations are the same in both schemes. In the PSS scheme, however, neither the server nor the members need to store the node keys generated after each rekeying. They can be deleted as soon as they are used in the decryption process. The sets (both the group set and the auxiliary ones), however, need to be kept until they are replaced. There

Table 6.1
Comparison of LKH and PSS schemes: Storage Cost

	LKH	PSS
# of keys held by server	$dn/(n-1)$	-
# of keys held by each member	h	-
# of share sets held by server	-	$dn/(n-1)$
# of share sets held by each member	-	h

is a 1-1 correspondence between the number of keys generated for each member and the number of sets held by each member.

Table 6.2
Comparison of LKH and PSS schemes: Communication Cost

	LKH	PSS
Join	$O(log_d(n))$	$O(log_d(n))$ and $O(1)$
Leave	$O(dlog_d(n))$	$O(dlog_d(n))$ and $O(1)$
Periodic rekeying	$O(d)$	$O(1)$

The size of the messages sent for join/leave operations are the same in both schemes. An additional communication cost in the PSS scheme for join/leave operations is the delivery of the activating share. The two schemes have different requirements in periodic rekeying. The communication cost for the PSS scheme is the delivery of the activating share and the communication cost for the LKH scheme is the delivery of d encrypted messages.

An additional computational cost in the PSS scheme for join/leave operations is the processing needed for the construction of the polynomials. There is a 1-1 correspondence between the number of polynomials constructed by the server and the

Table 6.3
LKH Computation Cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	0	$d/(d-1)$	$d(h-1)$
Periodic	d	1	

Table 6.4
PSS Computation Cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	$d(h-1)$	0	$d/(d-1)$
Periodic	0	0	

number of encryptions performed by the server. There is also a 1-1 correspondence between the number of polynomials constructed by each member and the number of decryptions performed by each member. The two schemes have different computational requirements to recover the group key in periodic rekeying. The PSS scheme needs one polynomial construction for the server and one polynomial construction for each member whereas the LKH scheme needs d encryptions for the server and one decryption for each member.

6.5.2 Simulation Parameters

Now we test the group key management scheme based on the pre-positioned secret sharing in the wireless cellular network. We employ a wireless cellular network that consists of 16 concatenated cells with 1 Mobile switching eXchanger (MX). We use

Table 6.5
Polynomial Construction Cost

	Server	Requesting member	Non-requesting member
Join	$2(h-1)$	$h-1$	$d/(d-1)$
Leave	$d(h-1)$	0	$d/(d-1)$
Periodic	1	1	

4 mobility models: $0 \sim 1$ km/hr for walking, $2 \sim 5$ km/hr for running, $6 \sim 25$ km/hr for low speed vehicle, and $26 \sim 100$ km/hr for high speed vehicle. The Poisson distribution with rate λ is used to model the number of calls occurring within a given time interval where λ is the shape parameter which indicates the average number of events in the given time interval. Exponential distribution with mean $1/\mu$ is used for the call duration. The close connection between the Poisson arrival process and the exponential interarrival time can be exploited immediately in properties of the exponential service time distribution. Table 6.6 shows the range of values and the constants for the parameters.

We showed in [5, 6] that our scheme is comparable to the Logical Key Hierarchy (LKH) schemes [33, 34] in the respect of communications cost, rekeying time cost, and memory cost in the wired network. With a the revised handoff scheme and a call admission control function, the number of handoff transactions per call was reduced by almost 20% compared to that of the soft handoff. In this paper, we will add more simulation results with respect to key update and handoff cost.

In a cellular system, a call originated in a cell gets a channel and holds it until that call is completed in the cell or the MS moves out of the cell. The channel holding time is either the call duration time or the time for which MS resides in the cell. This is a function of parameters such as the cell radius R(km), the MS speed V(km/hr), the direction of MS, etc.

Table 6.6
Simulation Parameters

Parameter	Value
# of MX	1
# of BS	16
# of MS	Up to 100 per BS
Call generation	Poisson with λ (calls/sec)
Call duration	Exponential with $1/\mu$ (1/sec)
User mobility	0-1 km/h (walking) 2-5 km/h (running) 6-25 km/h (low speed vehicle) 26-100 km/h (high speed vehicle)
Cell radius	1Km
Service	Voice, Data, Video
L_ADD	30% of BS coverage area
L_DROP	Boundary of BS

6.5.3 Key Update Costs in Wireless and Wireline Intervals

We set parameters to measure the number of transactions in wireless and wireline intervals such that μ=1/60 (/sec), λ=100 (calls/sec), V=50 (km/h), R=2(km), and simulation time= 5 minute. The cost represents the key updates transactions. That is, a new call arrival and a call termination mean 1 key update respectively. The wireless cost and the wireline cost of the location matching trees (our scheme) and the Logical tree are shown for different quantities of participating BSs. We observed that the location matching trees have both smaller wireless cost and smaller wireline costs than the logical trees when the number of BSs is equal or greater than 2, and the advantages of the matching trees are more significant when the system contains

more BSs. In this system, the communication cost of the matching trees can save as low as 20% of the communication cost of the independent trees as shown in Figure 6.8 and Figure 6.9.

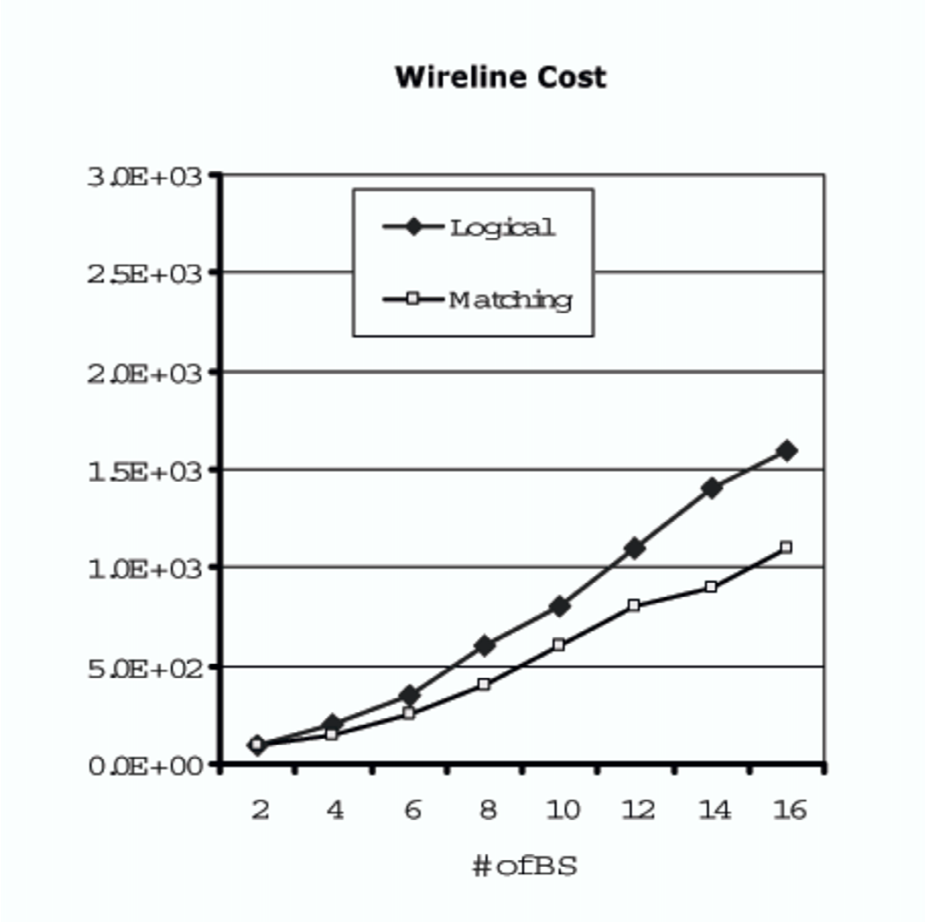


Fig. 6.8. Key Update Costs in Wireline Intervals

6.5.4 Handoff Cost

We set parameters to measure the number of handoff attempts for each user group such that μ=1/180 (/sec), λ=20 (calls/sec), R=2(km), and simulation time= 10 minutes. We observe that each user group undergoes 3 $\sim$ 8 handoffs during the call duration. Moving users onto the key tree causes some extra key management

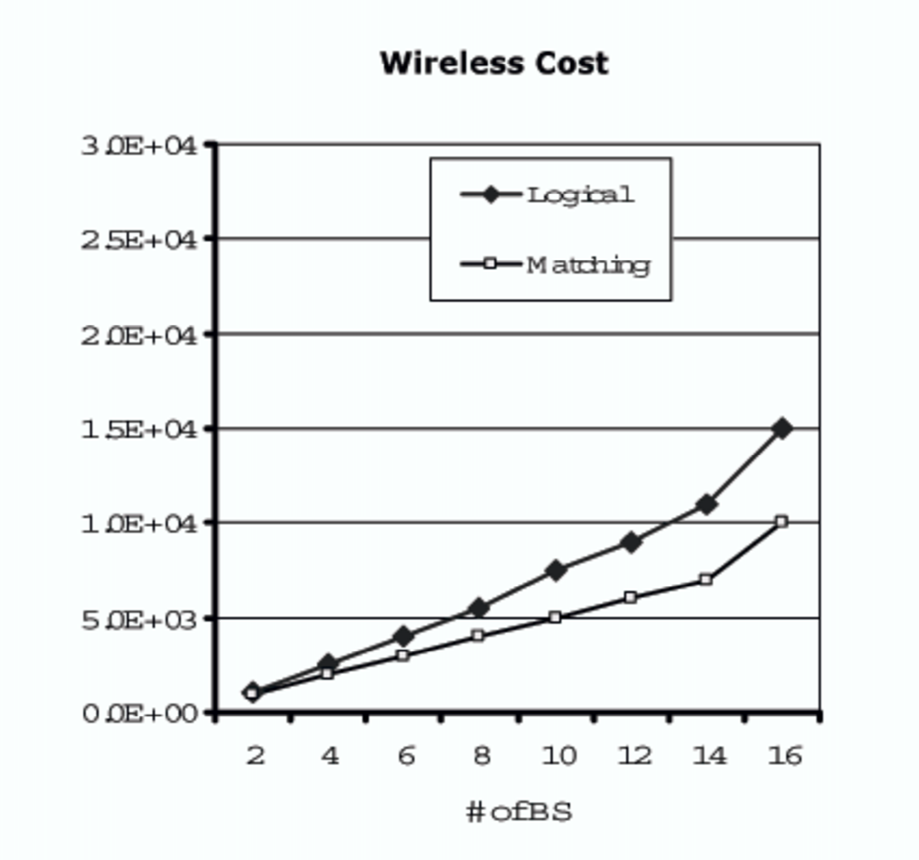


Fig. 6.9. Key Update Costs in Wireless Intervals

resources even though they are still in service because of handoff. To take care of frequent handoff between wireless access networks, we proposed a new revised handoff schemes. This new handoff scheme can reduce some key update costs in wireless and wireline intervals because it only updates the keys after completion handoff.

A call can have 3 key transactions during the call duration: call generation, handoffs, and call termination. A handoff call requires 2 key update transactions: (1) adding a new channel when a call enters handoff region and (2) deleting a serving channel after completing handoff.

Thus the number of key transactions during call duration, N, equals to

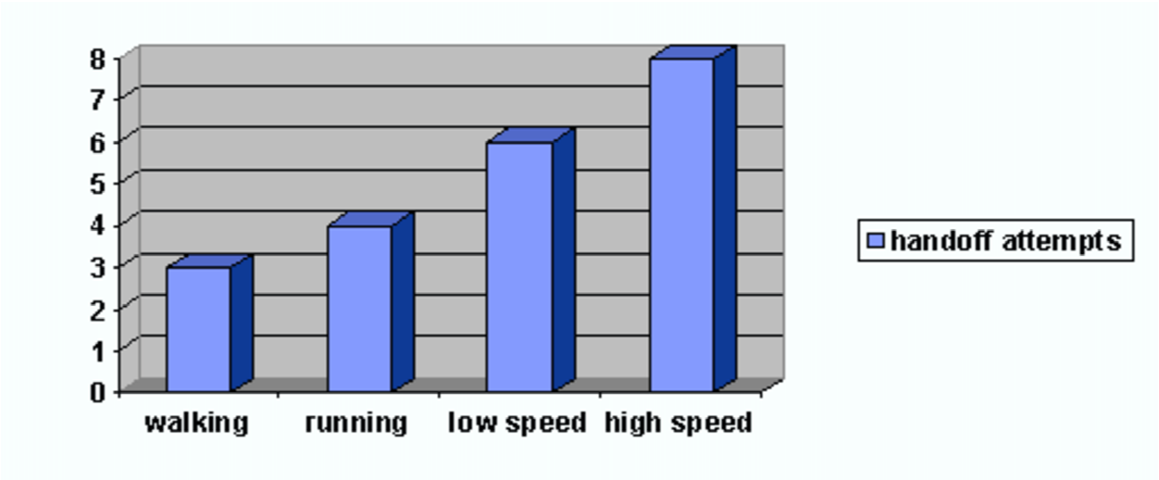


Fig. 6.10. Handoff Attempts for Each User Type

$$N = 1 \times (callgeneration) + 1 \times (calltermination) + 2 \times (\#ofHandoff) \tag{6.4}$$

We run our simulation 10 times and calculate the average handoff attempts per user according to the mobility models. Each call has $3 \sim 8$ handoffs during the call service time. We show the result in Figure 6.10.

In Figure 6.11 and Figure 6.12, we plot the number of handoff attempts as a function of the number of new calls. The number of handoff attempts increases linearly as the number of new calls increases. With the hard handoff schemes, the number of handoffs per call is reduced by about 20% comparing to the results of Figure 6.11. It's very expected result because the hard handoff scheme requires less key update transactions in handoff region.

Now we find that the handoff part can be the largest inefficiency in wireless cellular networks. To reduce the number of handoffs, we can increase the radius of cell. However, as the radius of cell increases, the system capacity decreases. That is, the total number of users in a system will be decreased if the radius of cell is increased. So we need an alternative method.

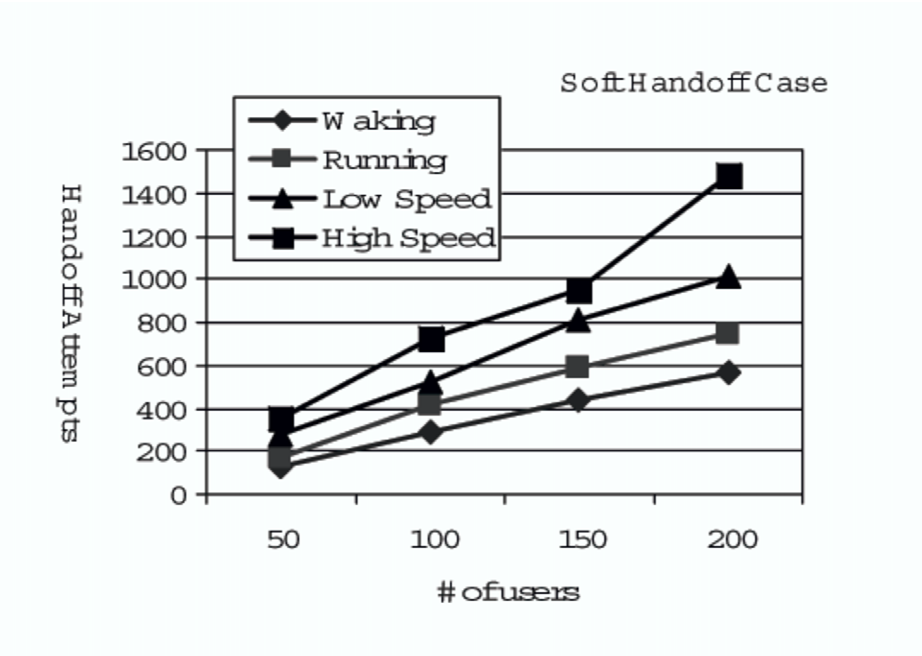


Fig. 6.11. The Number of Handoff Attempts in Soft Handoff Case

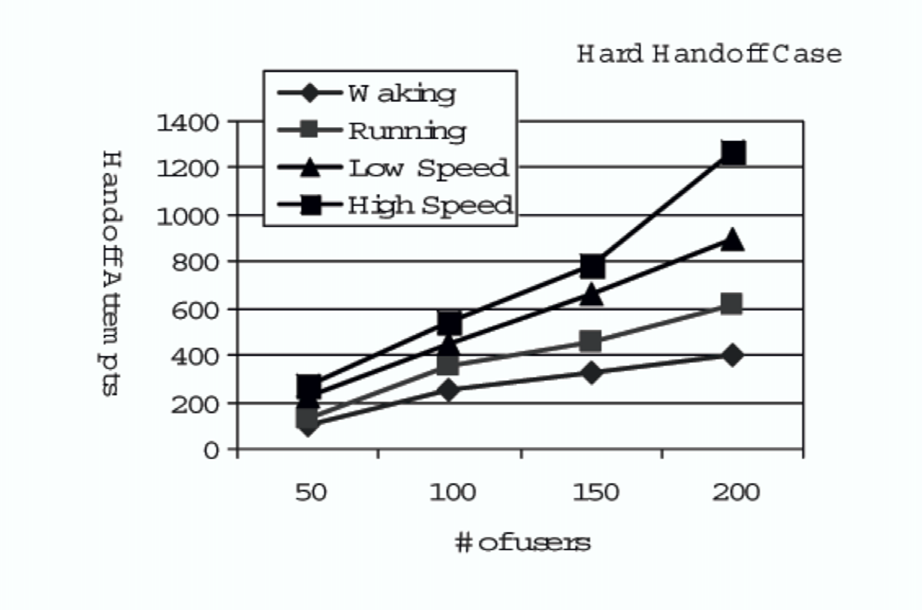


Fig. 6.12. The Number of Handoff Attempts in Hard Handoff Case

We don't take into account a call admission control (CAC) so far. That is, we don't restrict the number of users for each BS. The CAC function determines whether

to accept a new call and a handoff call [150]. With the CAC and the revised handoff schemes, the number of handoffs per call is reduced by almost 20% comparing to the results of Figure 6.12 until the threshold of CAC, here 100 users per BS. After the threshold, the handoff attempts stay to a certain level since the CAC limited the number of new calls. In Figure 6.13, we plot the number of the handoff attempts as a function of the number of new calls with a CAC and a revised handoff scheme. We find that the number of handoff attempts don't increase after 100 users. Because of the CAC, only 100 users are accepted in each BS.

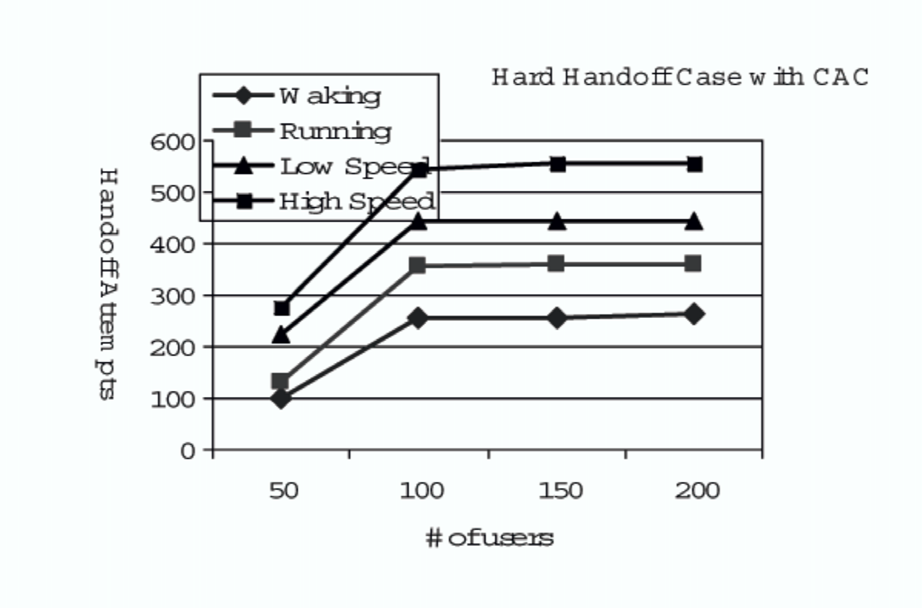


Fig. 6.13. The Number of Handoff Attempts vs. The Number of New Calls with a CAC.

7. CONCLUSIONS

In this dissertation, video encryption and multicast security problems in wireless networks were examined.

7.1 Contributions of this Dissertation

- **Selective video encryption for a LDPC and Turbo codes based distributed source coding method**

  A framework for implementing selective video encryption for a LDPC and Turbo codes based distributed source coding method is proposed. Secrecy results from a tradeoff between processing power and speed, but real-time processing is achievable. We showed that the encryption of 50% or more bits reveals no useful information in the reconstructed video. Hence the proposed method has some advantages over conventional full data encryption with regard to complexity. We are investigating how the compressed bit stream can be exploited by imposing a syntax on the output of the DSC coder.

- **Secure group key management algorithm**

  We designed a group key management tree such that the neighbors on the key tree are also physical neighbors on the cellular network. The group key management scheme uses the pre-positioned secret sharing scheme. By tracking the user location, we localized the delivery of rekeying messages to the nodes that need them. This lessens the amount of traffic in the cellular network. We find that each call undergoes an average of $3 \sim 8$ handoffs during a call duration according to the user mobility model. We proposed a new handoff scheme to minimize the key updating transactions. This new handoff scheme reduces one

of the two key update transactions in the handoff region - adding a new channel when a call enters the handoff region. In the handoff area, only a new traffic channel is added to minimize the interruption time of the data transmission. With a the revised handoff scheme, the number of handoffs per call is reduced by almost 20% compared to that of the soft handoff. Also a simple CAC function is used to maintain key updating transactions to a level defined by the system manager.

7.2 Future Work

- **Selective video encryption for distributed source coding**

  It is a future work to propose a generalized scheme to selectively encrypt video sequences, which can offer several advantages: flexibility, multiplicity, spatial selectiveness and format compliance. Also the integration and interoperability of different multimedia security techniques (e.g. encryption and robust watermarking or encryption and fragile watermarking) is another interesting research topic.

- **Secure group key management algorithm**

  As the future work, it remains as an open problem to efficiently implement and integrate the proposed protocols. The other problem is to support the additional functions such as authentication in lower complexity of communication and computation.

LIST OF REFERENCES

LIST OF REFERENCES

[1] 3GPP Organizational Partners, "http://portal.etsi.org/dvbanca/3gpp/3gppspecs.asp," 2000.

[2] CDMA Technology Group, "http://www.cdg.org/technology/3g/," 2001.

[3] Wireless LAN Association, "http:http://www.wlana.org/," 2005.

[4] IEEE 802.16e Task Group , "http://www.ieee802.org/16/tge/," 2005.

[5] H. Um and E. J. Delp, "A secure group key management scheme for wireless cellular networks," in *Proceedings of the International Conference on Information Technology: New Generations (ITNG'06)*, (Las Vegas, Nevada), April 10-12 2006.

[6] H. Um and E. J. Delp, "A new secure group key management scheme for multicast over wireless cellular networks," in *Proceedings of the International Performance, Computing, and Communication Conference (IPCCC'06)*, (Phoenix, Arizona), April 10-12 2006.

[7] H. Um and E. J. Delp, "Selective video encryption of a distributed coded bitstream using LDPC codes," in *Proceedings of IS-T/SPIE Symposium on Electronic Imaging on Security, Steganography, and Watermarking of Multimedia Contents VIII (EI'06)*, (San Jose, Califonia), January 15-19 2006.

[8] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp, "Advances in digital video content protection," *Proceedings of the IEEE*, vol. 93, pp. 171–183, 2005.

[9] N. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations," in *Proceedings of the ASIACRYPT'02*, pp. 267–287, 2002.

[10] "Data Encryption Standard(DES), National Bureau of Standards FIPS Publication 46," 1977.

[11] "Data Encryption Standard(DES), National Bureau of Standards FIPS Publication 81," 1980.

[12] A. Uhl and A. Pommer, *Image and Video Encryption: From digital rights managements to secured personal communication.* 233, Spring Street, New York, NY10013: Springer, 2005.

[13] "ITU-T Recommendation H.261, Video codec for audiovisual services at p*64 kb/s," 1990.

[14] "ITU-T Recommendation H.263 Version (H.263), Video codec for low bitrate communications," March 1996.

[15] "ITU-T Recommendation H.263 Draft Version 2 (H.263+), Video codec for low bitrate communications," September 1997.

[16] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 Advanced Video Coding)." Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-E022, September 2002.

[17] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)." Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, March 2003.

[18] I. E. Richardson, *H.264 and MPEG-4 Video Compression*. Wiley, 2003.

[19] "ISO/IEC DIS 13 818 (MPEG-2), Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s," 1993.

[20] "ISO/IEC IS 11172 (MPEG-1), Generic coding of moving pictures and associated audio information," 1994.

[21] "ISO/IEC DIS 14 496 (MPEG-4), Very-low bitrate audio-visual coding," 1998.

[22] "Coding of audio-visual objects, Part-2 Visual, Amendment 4: Streaming video profile." ISO/IEC 14496-2/FPDAM4, July 2000.

[23] "ISO/IEC DIS 14996-1:2000/AMD3 (MPEG-4 IPMP), MPEG4 IPMP (Intellectual Property Management and Ptotection) Final Proposed Draft Amendment," 2002.

[24] J. Vass, S. Zhuang, and X.Zhuang, "Scalable, error-resilient, and high-performance video communications in mobile wireless environments," *IEEE Transactions Circuits Systems and Video Technology*, vol. 11, no. 7, pp. 833–847, 2001.

[25] S. Wee and J. Apostolopoulos, "Secure scalable video streaming for wireless networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2049–2052, May 2001.

[26] D. Wu, T. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broadband wireless networks," *Proceedings of the IEEE, Special Issue on Multi-Dimensional Broadband Wireless Technologies and Applications*, vol. 89, pp. 6–20, January 2001.

[27] B. Sklar, "Rayleigh fading channels in mobile digital communication systems.part I: Characterization," *IEEE Communication Magagin*, vol. 35, pp. 90–100, July 1997.

[28] J. Villasenor, Y.-Q. Zhang, and J. Wen, "Robust video coding algorithms and systems," *Proceedings of IEEE*, vol. 87, pp. 1724–1733, October 1999.

[29] X. Lai and J. L. Massey, "A proposal for a new block encryption standard," in *Proceedings of the EUROCRYPT'90*, pp. 389–404, 1990.

[30] X. Lai, J. L. Massey, and S. Murphy, "Markov ciphers and differential cryptanalysis," in *Proceedings of the EUROCRYPT'91*, pp. 17–38, 1991.

[31] C. Shannon, "Coomunication theory of secrecy systems," *Bell Systems Techical Journal*, vol. 28, pp. 656–715, October 1949.

[32] T. Lookabaugh and D. C. Sicker, "Selective encrytion for consumer applications," *IEEE Communications Magazine*, pp. 124–129, May 2004.

[33] S. Mitta, "Iolus: A framework for the scalable secure multicasting," in *Proceedings of the ACM SIGCOMM'97*, pp. 277–288, September 1997.

[34] C. Wong, M.Gouda, and S. Lam, "Secure group communication using key graphs," *IEEE/ACM Transaction on Networking*, vol. 8, pp. 16–30, Febuary 2000.

[35] R. Molva and A. Pannetrat, "Scalable multicast security in dynamic groups," in *Proceedings of the 6th ACM conference on Computer and communications security*, (Kent Ridge Digital Labs, Singapore), pp. 101–112, 1999.

[36] R. Molva and A. Pannetrat, "Scalable multicast security with dynamic recipient groups," *ACM Transactions on Information and System Security*, vol. 3, pp. 136–160, August 2000.

[37] S. Kumar and P. Radoslavov, "The MASC/BGMP architecture for inter-domain multicast routing," in *Proceedings of the ACM SIGCOMM'98*, (Vancouver, Canada), pp. 93–104, August 1998.

[38] C. Shields and J. J. Garcia-Luna-Aceves, "KHIP a scalable protocol for secure multicast routing," in *Proceedings of the ACM SIGCOMM'99*, (Cambridge, MA), pp. 53–63, August 1999.

[39] R. Song and L. Korba, "Cryptanalysis of scalable multicast security protocol," *IEEE COMMUNICATIONS LETTERS*, vol. 7, pp. 561–563, NOVEMBER 2003.

[40] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Computing Survey*, vol. 35, pp. 309–329, Semtember 2003.

[41] L. Dondeti, S. Mukherjee, and A. Samal, "Comparison of hierarchical key distribution schemes," in *Proceedings of the IEEE Globecom Global Internet Symposium*, (Rio de Janerio, Brazil), December 1999.

[42] R. Canetti, J. Garayt, G. Itkid, D. Micciancios, M. Naore, and B. Pinkasll, "Multicast security: A taxonomy and some efficient constructions," in *Proceedings of the IEEE INFOCOM '99*, vol. 2, (Cambridge, MA), pp. 708–716, March 21-25 1999.

[43] Y. Sun, W. Trappe, and K. J. R. Liu, "An efficient key management scheme for secure wireless multicast," in *Proceedings of the IEEE International Conference on Communication (ICC'02)*, pp. 1236–1240, 2002.

[44] D. BT *et al.*, "Secure group communications for wireless networks," in *Proceedings of the IEEE MILCOM 2001*, (Mclean, VA), October 2001.

[45] A. Wyner and J. Zip, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 1–10, January 1976.

[46] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471–480, Semtember 1973.

[47] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE: Special Issue on Advances in Video Coding and Delivery*, vol. 93, pp. 71– 83, January 2005.

[48] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," in *Proceedings of the IEEE Data Compression Conference (DCC'99)*, March 1999.

[49] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, pp. 626–643, March 2003.

[50] R. Puri and K. Ramchandran, "PRISM: a reversed multimedia coding paradigm," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, vol. 1, pp. 617–620, Semtember 2003.

[51] R. Puri and K. Ramchandran, "PRISM: a video coding paradigm based on motion-compensated prediction at the decoder." submitted.

[52] L. Liu, Y. Liu, and E. J. Delp, "Network-driven Wyner-Ziv video coding using forward prediction," in *Proceedings of the SPIE International Conference on Visual Communications and Image Processing*, (San Jose, CA), pp. 641–651, Januaray 2005.

[53] L. Liu, P. Sabria, L. Torres, and E. J. Delp, "Error resilience in network-driven wyner-ziv video coding," in *Proceedings of the SPIE International Conference on Image and Video Communications and Processing, accepted*, (San Jose, CA), January 15-19 2006.

[54] Z. Li and E. Delp, "Wyner-ziv video side estimator: Conventional motion search methods revisited," in *Proceedings of the IEEE International Conference on Image Processing*, (Genova, Italy), pp. 825–828, September 11-15 2005.

[55] Z. Li, L. Liu, and E. J. Delp, "Wyner-ziv video coding: A motion estimation perspective," in *Proceedings of the SPIE International Conference on Image and Video Communications and Processing, accepted*, (San Jose, CA), January 15-19 2006.

[56] Z. Li, L. Liu, and E. J. Delp, "Wyner-ziv video coding with universal prediction," in *Proceedings of the SPIE International Conference on Image and Video Communications and Processing, accepted*, (San Jose, CA), January 15-19 2006.

[57] S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, pp. 1181 – 1203, May 2003.

[58] M. Costa, "Writing on dirty paper," *IEEE Transactions on Information Theory*, vol. 29, pp. 439– 441, May 1983.

[59] L. Liu, *New Approaches For Low Complexity Vodeo Coding and Reliable Transmission*. preliminary report, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, December 2005.

[60] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, pp. 6– 17, Jan. 2005.

[61] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. IT-8, pp. 21–28, January 1962.

[62] D. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, pp. 399–431, March 1999.

[63] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*. New York: John Wiley & Sons, April 2002.

[64] S. Shamai, S. Verdu, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 564–579, March 1998.

[65] M. Sartipi and F. Feriki, "Distributed source coding in wireless sensor networks using LDPC coding: The entire Slepian-Wolf Rate Region," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2005)*, pp. 1939–1944, 2005.

[66] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on Information Theory*, vol. 47, pp. 585–598, February 2001.

[67] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA), pp. 240–244, November 2002.

[68] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of the IEEE Data Compression Conference (DCC)*, (Snowbird, UT), pp. 252–261, April 2-4 2002.

[69] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," in *Proceeding of IEEE International Conference on Image Processing, ICIP-2003*, (Barcelona, Spain), pp. 869–872, September 14-17 2003.

[70] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-ziv coding for video: applications to compression and error resilience," in *Proceedings of the IEEE Data Compression Conference (DCC 2003)*, pp. 93–102, March 2003.

[71] A. Aaron, S. Rane, D. Rebollo-Monedero, and B. Girod, "Systematic lossy forward error protection for video waveforms," in *Proceeding of IEEE International Conference on Image Processing, ICIP-2003*, (Barcelona, Spain), pp. 609–612, September 14-17 2003.

[72] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2004)*, (Singapore), October 2004.

[73] A. Aaron, S. Rane, , E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proceedings of the SPIE: Visual Communications and Image Processing (VCIP 2004)*, (San Jose, CA), Januaray 2004.

[74] Q. Xu and Z. Xiong, "Layered wyner-ziv video coding," in *Proceedings of the IEEE Visual Communications and Image Processing (VCIP 2004): Special Session on Multimedia Technologies for Embedded Systems*, (San Jose, CA), January 2004.

[75] A. Sehgal, A. Jagmohan, and N. Ahuja, "A state-free causal video encoding paradigm," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, vol. 1, pp. 605–608, Semtember 2003.

[76] MPEG Technology Group, "http://www.chiariglione.org/mpeg/," 1998.

[77] J. Daemen and V. Rijmen, *The Design of Rijndael:AES-The Advanced Encryption Standard*. ISBN 3540425802: Springer-Verlag, 2002.

[78] I. Agi and L. Gong, "An emprical study of MPEG video transmissions," in *Proceedings of The Internet Society Symposium on Network and Distributed System Security*, (San Diego, CA), pp. 137–144, Febuary 1996.

[79] L. Qiao and K. Nahrstedt, "Comparison of MPEG encryption algorithms," *Proceedings of the International Journal of Computers and Graphics, special issue: Data Security in Image Communication and Network*, vol. 28, January 1998.

[80] L. Tang, "For encrypting and decrypting MPEG video data efficiently," in *Proceedings of The Fourth ACM International Multimedia Conference (ACM Multimedia'96)*, (Bosten, MA), pp. 219–230, November 1996.

[81] L. Qiao and K. Nahrstedt, "A new algorithm for MPEG video encryption," in *Proceedings of The First International Conference on Imaging Science, Systems, and Technology (CISST'97)*, (Las Vegas, Nevada), pp. 21–29, July 1997.

[82] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on signal processing*, vol. 48, pp. 2439–2451, August 2000.

[83] C. Shi and B. Bharagava, "A fast MPEG video encryption algorihtm," in *Proceedings of the 6th International Multimedia Conference*, (Bristol, UK), Setember 1998.

[84] T. Lookabaugh *et al.*, "Selective encryption of MPEG-2 video," in *Proceedings of the SPIE Multimedia Systems and Applications VI*, (Orlando, FL), September 2003.

[85] C. Griwotz, "Video protection by partial content corruption," in *Proceedings of Multimedia and Security Workshop at the 6th ACM International Multimedia Conference*, (Bristol, England), pp. 37–39, 1998.

[86] C. Griwotz, O. Merkel, J. Dittmann, and R. Steinmetz, "Protecting vod the easier way," in *Proceedings of Multimedia and Security Workshop at the 6th ACM International Multimedia Conference*, (Bristol, England), pp. 21–28, 1998.

[87] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin, "A format compliant configurable encryption framework for access control of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 545–557, June 2002.

[88] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, pp. 36–58, September 2001.

[89] D. Taubman and M. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[90] "Information technology−JPEG 2000 image coding system−Part 1: Core coding system." Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 15444-1:2000 and ITU-T T.800.

[91] R. Norcen and A. Uhl, "Robust authentication of the JPEG2000 bitstream," in *Proceedings of the IEEE Nordic Signal Processing Symposium (NORSIG 2004)*, (Espoo, Finland), June 2004.

[92] D. Engel and A. Uhl, "Security enhancement for lightweight JPEG 2000 transparent encryption," in *Proceedings of the Fifth International Conference on Information, Communication and Signal Processing (ICICS '05)*, (Bangkok, Thailand), December 2005.

[93] M. Rhepp, H. Stogner, and A. Uhl, "Comparison of JPEG and JPEG 2000 in low-power confidential image transmission," in *Proceedings of the European Signal Processing Conference (EUSIPCO'04)*, (Vienna, Austria), September 2004.

[94] A. Uhl and C. Obermair, "Transparent encryption of JPEG2000 bitstreams," in *Proceedings of the Fifth EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, (Smolenice, Slovak Republic), 2005.

[95] Y. Sadourny and V. Conan, "A proposal for supporting selective encryption in JPSEC," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 846–849, November 2003.

[96] Z. Zhang, Q. Sun, G. Qiu, Y. Shi, and Z. Ni, "A unified authentication framework for JPEG2000," in *Proceedings of the IEEE ICME 2004*, 2004.

[97] Y.Wu, D. Ma, and R. Deng, "Progressive protection of JPEG2000 codestreams," in *Proceedings of the IEEE ICIP 2004*, 2004.

[98] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 49, pp. 120–126, February 1978.

[99] R. Rivest, "The MD4 message digest algorithm," in *Advances in Cryptology – CRYPTO '90 Proceedings*, pp. 303–311, 1991.

[100] H. Dobbertin, "Cryptanalysis of MD4," pp. 53–69, 1996.

[101] "Secure Hash Standard: Federal Information Processing Standards Publication 180-1," 1995.

[102] S. Wee and J. Apostolopoulos, "Secure transcoding with JPSEC confidentiality authentication," in *Proceedings of the IEEE ICIP 2004*, 2004.

[103] R. G. Gallager, "Low density parity check codes," *MIT Press*, 1963.

[104] T. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on information theory*, vol. 47, pp. 619–637, Febuary 2001.

[105] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Proceedings of the SPIE: International Conference on Image and Video Communications and Processing(IVCP)*, (San Jose, CA), January 18-22 2004.

[106] S. Babvey, A. Bourgeois, and S. W. McLaughlin, "An Efficient R-Mesh Implementation of LDPC Codes Message-Passing Decoder," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005.

[107] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communication Letter*, vol. 6, pp. 440–442, October 2002.

[108] K. Jack, *Video Demystified*. San Diego, CA: HighText Interactive, Inc., 1996.

[109] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding:turbo-codes," in *International Conference on Communications*, (Geneva, Switzerland), pp. 1064–1070, May 23-26 1993.

[110] W. B. Rabiner and A. P. Chandrakasan, "Network-driven motion estimation for wireless video terminals," *IEEE Transactions on Circuits and Systems for Video Technology,*, vol. 7, pp. 644–653, August 1997.

[111] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC)," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, JVT-G050, 2003.

[112] C. Berrou and A. Glavieux, "Near optimium error correcting coding and decoding: Turbo codes," *IEEE Transactions on Communications*, vol. 344, pp. 1261–1271, October 1996.

[113] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, vol. 36, pp. 112–119, June 1998.

[114] R. Norcen, M. Podesser, A. Pommer, H. P. Schmidt, and A. Uhl, "Confidential storage and transmission of medical image data," *Computers in Biology and Medicine*, vol. 33, no. 3, pp. 277–292, 2003.

[115] R. Norcen and A. Uhl, "Selective encryption of the jpeg2000 bitstream," in *Proceedings of IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security (CMS'03)*, (Turin, Italy), pp. 194–204, October 2003.

[116] A. Pommer and A. Uhl, "Selective encryption of wavelet packet subband structures for secure transmission of visual data," in *ACM Multimedia: Multimedia and Security Workshop*, (Juan-les-Pins, France), pp. 67–70, December 2002.

[117] D. R. Stinson, *Cryptography*. Boca Raton, Florida: CRC Press LLC, 1995.

[118] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Transaction on signal processing*, vol. 52, pp. 2992–3006, October 2004.

[119] M. Moyer, J. Rao, and P. Rohatgi, "Survey of security issues in multicast communications," *IEEE Network*, vol. 13, pp. 12–23, November-December 2003.

[120] L. Gong and N. Scacham, "Elements of trusted multicasting," in *Proceedings of the IEEE International Conference on Network Protocols*, (Boston, MA), pp. 23–30, October 1994.

[121] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architecture. RFC2627," 1999.

[122] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The versakey framework: Versatile group key management," *IEEE Journal on Selected Areas in Communications (Special Issue on Middleware)*, vol. 17, pp. 1614–1631, August 1999.

[123] A. Perrig, D. Song, and J. D. Tygar, "Elk: A new protocol for efficient large-group key distribution," in *Proceedings of the IEEE Symposium on Security and Privacy*, (Oakland, CA), May 2001.

[124] D. A. McGrew and A. T. Sherman, "Key etablishment in large dynamic groups using one-way function trees," May 1998.

[125] S. Rafaeli and D. Hutchison, "Hysdra: A decentralised group key management," in *Proceedings of the 11th IEEE international WETICE: Enterprise Security Workshop*, 2002.

[126] G. Resta and P. Santi, "An analysis of the node spatial distribution of the random waypoint model for ad hoc networks," in *Proceedings of the ACM Workshop on Principles of Mobile Computing (POMC'2002)*, pp. 44–50, 2002.

[127] C. Bettstetter and C. Wagner, "The spatial node distribution of the random waypoint mobility model," in *Proceedings of the German Workshop on Mobile Ad Hoc Networks (WMAN)*, pp. 41–58, 2002.

[128] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz, "Secure multicast groups on ad hoc networks," in *Proceedings of the 1st 1st ACM workshop on Security of ad hoc and sensor*, pp. 94–102, 2003.

[129] T.-C. Chiang and Y.-M. Huang, "Group keys and the multicast security in ad hoc networks," in *Proceedings of the International Conference on Parallel Processing Workshops*, pp. 385–390, October 6-9 2003.

[130] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "Gkmpan: An efficient group rekeying scheme for secure multicast in ad-hoc networks," in *Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004)*, pp. 42–51, 2004.

[131] F. Dai and J. Wu, "Constructing k-connected k-dominating set in wireless networks," in *Proceedings of the 19th International Parallel and Distributed Processing Symposium*, April 2005.

[132] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 31–37, March 1996.

[133] A. Perrig, "Efficient collaborative key management protocols for secure autonomous group communication," in *Proceedings of the international Workshop on Cryptographic Techniques and E-commerce (CryptTEC'99)*, pp. 192–202, 1999.

[134] C. Becker and U. Wille, "Communication complexity of group key distribution," in *Proceedings of the 5th ACM conference on computer and communications security*, November 1998.

[135] C. Boyd, "On key agreement and conference key agreement," in *Proceedings of the information security and privacy: Austalasian conference*, pp. 294–302, 1997.

[136] Y. Kim, A. Perrig, and G. Tsudik, "Communication-efficient group key agreement," in *Proceedings of the 17th International Information Security Conference IFIP SEC'01*, 2001.

[137] M. Burmester and Y. Desmedt, "secure and efficient conference key distribution system," in *Proceedings of the EUROCRYPT'94*, pp. 275–286, 1994.

[138] "TIA/EIA interim standard (IS-95), mobile station - base station compatibility standards for dual mode wideband spread spectrum cellular system," July 1993.

[139] P. Enge and P. Misra, "Special issues on global positioning system," *Proceedings of the IEEE*, vol. 87, January 1999.

[140] T. S. Rappaport, J. H. Reed, and B. D. Woerner, "Position location using wireless communications on highways of the future," *IEEE Communications Magazine*, October 1996.

[141] N. Levanon, "Quick positioning determination using 1 or 2 LEO satellites," *IEEE Transactions On Aerospace and Electronic systems*, vol. 34, July 1998.

[142] K. Narenthiran, R. Tafazolli, and B. G. Evans, "Simple positioning method for location tracking in mobile satellite communications," in *Proceedings of the 18th AIAA International Communication Satellite Systems Conference*, (Oakland USA), April 2000.

[143] G. J. Simmons, "How to (really) share a secret," *Proceedings of the Advances in Cryptology - CRYPTO'88 , Springer-Verlag*, pp. 390–448, 1990.

[144] G. J. Simmons, "Prepositioned shared secret and/or shared control schemes," *Proceedings of the Advances in Cryptology - EUROCRYPT'89, Springer-Verlag*, pp. 436–467, 1990.

[145] A. M. Eskicioglu and M. R. Eskicioglu, "Multicast security using key graphs and secret sharing," in *Proceedings of the Joint International Conference on Wireless LANs and Home Networks ICWLHN 2002 and Networking ICN 2002*, (Atlanta, GA), pp. 228–241, August 26-29 2002.

[146] A. M. Eskicioglu, "Multimedia security in group communications: Recent progress in key management, authentication, and watermarking," *ACM Multimedia Systems Journal, Special Issue on Multimedia Security*, pp. 239–248, 2003.

[147] A. M. Eskicioglu, S. Dexter, and E. J. Delp, "Protection of multicast scalable video by secret sharing: Simulation results," in *Proceedings of the IEEE MILCOM 2003*, (Santa Clara, CA), January 2003.

[148] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 11, pp. 612–613, November 1979.

[149] L. Zhou and M. Frei, "http://www.cs.cornell.edu/courses/cs513/2000sp/ secretsharing.html," 2000.

[150] H. Um, "Access control schemes for DS-CDMA cellular system supporting an integrated voice/data traffic," in *Proceedings of the SBT/IEEE International Telecommunications Symposium 1998*, pp. 72–77, August 1998.

VITA

VITA

Hwayoung Um was born in Jeonju, Korea. He received B.S. and M.S. in Department of Electrical and Computer Engineering at Hanyang University in 1991 and 1993, respectively. He received Ph.D. degree from School of Electrical and Computer Engineering at Purdue University in 2006. His advisor is Professor Edward J. Delp.

Before came to Purdue, he worked at Electronics and Telecommunication Research Institute (ETRI), Korea, from 1993 to 2000. After joining to Purdue, he was a graduate research assistant at Video and Image Processing Laboratory (VIPER).

He is a student member of the IEEE. His research interests include video communication, sensor network, security, wireless networks, and low complexity video and image processing.

PUBLISHED PAPERS:

1. Hwayoung Um and Edward J. Delp, "Selective Video Encryption Of A Distributed Coded Bitstreams," submitted to the Journal of Electronic Imaging.
2. Hwayoung Um and Edward J. Delp, "A New Secure Group Key Management Scheme for Multicast over Wireless Cellular Networks," submitted to the International Journal of Network Security (IJNS).
3. Hwayoung Um and Edward J. Delp, "Partial Video Encryption Of A Distributed Coded Bitstream Using Turbo Code," submitted to the IS-T/SPIE Symposium on Electronic Imaging on Security, Steganography, and Watermarking of Multimedia Contents, January, 2007, San Jose, Califonia, USA.
4. Hwayoung Um and Edward J. Delp, "Selective Video Encryption Of A Distributed Coded Bitstream Using LDPC Codes," in Proceedings of the IS-T/SPIE Symposium on Electronic Imaging on Security, Steganography, and Watermarking of Multimedia Contents VIII, January 15-19, 2006, San Jose, Califonia, USA.

5. Hwayoung Um and Edward J. Delp, "A New Secure Group Key Management Scheme for Multicast over Wireless Cellular Networks," in Proceedings of the IEEE International Performance Computing and Communications Conference (IPCCC 2006), April 10-12, 2006, Phoenix, Arizona, USA.

6. Hwayoung Um and Edward J. Delp, "Secret Sharing Based Group Key Management Scheme for Wireless Cellular Networks," in Proceedings of the IEEE International Conference on Information Technology (ITNG 2006), April 10-12, 2006, Las Vegas, NV, USA.

PUBLISHED PAPERS with ETRI:

1. H.Y. Um, B.H. Ryu, J.H. Ahn, "Call Handling Capacity of base Station for Mobile Originated Calls in the CDMA Mobile System," CDMA International Conference (CIC'99), pp.449-453, 1999, Seoul, Korea.

2. R.H. Ryu, Y.O. Park, and H.Y. Um, "A Call Control Method for Multiple Traffic Classes in CDMA System," ICT'99, pp. 304-308, 1999.

3. H.Y. Um and B.H. Ryu, "Processing Capacity and Transmission Delay for Voice Traffic in IMT-2000 Networks," APCC'99, pp. 545-549, 1999.

4. H.Y. Um, "Access control schemes of integrated voice/data traffic in CDMA cellular system," IUCPC'98, pp. 307-310, 1998.

5. G.Z Ko, H.Y. Um, and K.S. Kim, "Throughput and Delay Analysis of a Multimedia DS-CDMA Unslotted ALOHA system with Imperfect Power Control," JCCI'98, Vol. II, pp. 1002-1006, April 1998.

6. H.Y Um and S.Y. Lim, and J.H. Ahn, "Evaluation of Optimal Wireless ATM Cell Structure," CIC'97, pp. 560-564, 1997.

7. H.Y Um and S.Y. Lim, "FEC Schemes for Wireless ATM Cell Configuration," MoMuC'97, pp. 108-111, 1997.

8. H.Y Um and S.Y. Lim, "Call Waiting Time Reduction Schemes for Dynamic TDMA Protocol," ICUPC'97, pp. 64-68, 1997.

9. H.Y. Um, Y.O. Park, W.G. Park, and S.S. Song, "Performance Analysis of Hybrid ARQ for Broadband Wireless ATM System," ICT'97, pp. 1073-1077, 1997.

10. W.G. Park, H.Y. Um, S.Y. Lim, S.J. Lim, and S.H. Lee, "The Model for Traffic Control Functions Based Wireless ATM Network," IEEE ATM'97 Workshop, pp. 410-415, January 1997.

11. W.G. Park, H.Y. Um, J.H. Ahn, and S.H. Lee, "Performance Analysis on Traffic Load Shedding Schemes for Mobile Communication System," ICUPC'97, pp. 306-310, January 1997.

12. K.H. Doo, H.J. Lee, H.Y. Um, and S.S. Song, "FEC/SR-ARQ Error Correction Schemes for Wireless ATM Cell Structure," JCCI'97, pp. 955-959, April 1997.

13. H.Y. Um, "Reverse Link Site Diversity for Micro-Cellular DS-CDMA System," ITS'96, pp. 225-228, September 1996.

14. S.J. Lee, W.B. Lyu, and H.Y. Um, "Anti-Jamming Algorithm For Traffic Load Shedding in DS-CDMA Cellular System," ICECS'96, pp.1056-1059, August 1996.

15. W.G. Park, H.Y. Um, and J.H. Ahn, "A Highly Reliable Maintenance and Administration SW in the BS for CMS: Design and Implementation," ICEIC'95, Vol. II, pp. 198-201, April 1995.

REGISTERED PATENTS with ETRI

1. H.Y. Um, J.H. Ahn, and K.S. Kim, "Data Transfer Method for Improving Capacity of DS-CDMA Mobile System," Registration Number 0250450, January 2000

2. H.Y. Um, J.H. Ahn, and K.S. Kim, "A MAC Method for CDMA Network," Registration Number 0240634, December 1999

3. H.Y. Um, "New Method for Determination of Request Slot in TDMA," Registration Number 0212463, May 1999

4. S.J. Lee, H.Y. Um, S.Y. Lim, and W.G. Park, "Anti-Jamming Algorithm For Traffic Load Shedding in DS-CDMA Cellular System," Registration Number 0211031, April 1999

5. H.Y. Um and S.J. Lee, "A Traffic Capacity Control Method of Base Station in CDMA Mobil System," Registration Number 194602, February 1999

6. S.J. Lee, H.Y. Um, and S.Y. Lim, "Dynamic Traffic Control Method for DS/CDMA Cellular System," Registration Number 170187, October 1998

7. W.G. Park and H.Y. Um, "Process Loading Data Control Method Using Operator Commands in CDMA Mobile System," Registration Number 146552, May 1998

8. H.Y. Um, S.J. Lee, S.Y. Lim, and W.G. Park, "Method for Loading of Common Data Initial in CDMA Mobile System," Registration Number 116141, June 1997