DOMINIK ENGEL

MEDIA ENCRYPTION FOR STILL VISUAL DATA

MEDIA ENCRYPTION FOR STILL VISUAL DATA

AN ANALYSIS OF SELECTED TECHNIQUES FOR NATURAL IMAGES AND FINGERPRINT DATA IN THE SPATIAL AND WAVELET DOMAIN.

DOMINIK ENGEL

Dissertation

zur Erlangung des akademischen Grades
Doktor der technischen Wissenschaften
an der Naturwissenschaftlichen Fakultät der
Universität Salzburg

unter der Betreuung von
Ao.Univ.-Prof. Dr. Andreas Uhl

Juni 2008

TO LINDA

ABSTRACT

We propose, discuss, and evaluate a range of techniques for media encryption for still visual data. Two classes of visual data are used in the investigation: natural still images and fingerprint data. In terms of representational format, we focus on scalable representations of the source data.

We propose different security techniques for visual data coded with the wavelet-based scalable JPEG2000 standard, some of which are exclusive to JPEG2000, whereas others are more generally applicable. We discuss techniques for format-compliant JPEG2000 encryption and propose a bitstream-oriented packet header protection scheme that can overcome security flaws present in virtually all existing format-compliant encryption scheme for JPEG2000. We then turn to compression-integrated encryption of wavelet-coded visual data. We propose and evaluate the utility of two classes of key-dependent transform domains for encryption: parameterized wavelet filters and wavelet packets. We show that the former are insecure because they are vulnerable to a symbolic transform attack, whereas the latter can achieve high levels of security while retaining competitive compression performance, especially in the case of anisotropic wavelet packets.

A major focus of this thesis is the analysis and evaluation of the proposed schemes, but also of existing media encryption schemes. One area where the severity of the compromise in security that is introduced by design mistakes is especially evident is the protection of biometric data, where revocation is usually impossible. In this context we analyze an image-based selective encryption scheme that has recently been proposed in the context of fingerprint encryption. We discuss severe design mistakes and present an efficient attack.

PUBLICATIONS

Parts of this thesis have appeared previously in the following publications.

D. Engel and A. Uhl. Parameterized biorthogonal wavelet lifting for lightweight JPEG2000 transparent encryption. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '05*, pages 63–70, New York, NY, USA, Aug. 2005a. (Cited on pages 12 and 65.)

D. Engel and A. Uhl. Security enhancement for lightweight JPEG2000 transparent encryption. In *Proceedings of Fifth International Conference on Information, Communication and Signal Processing, ICICS '05*, pages 1102–1106, Bangkok, Thailand, Dec. 2005b. (Cited on pages 12 and 81.)

D. Engel and A. Uhl. Secret wavelet packet decompositions for JPEG2000 lightweight encryption. In *Proceedings of 31st International Conference on Acoustics, Speech, and Signal Processing, ICASSP '06*, volume V, pages 465–468, Toulouse, France, May 2006a. IEEE. (Cited on pages 12 and 152.)

D. Engel and A. Uhl. Lightweight JPEG2000 encryption with anisotropic wavelet packets. In *Proceedings of International Conference on Multimedia & Expo, ICME '06*, pages 2177–2180, Toronto, Canada, July 2006b. IEEE. (Cited on pages 12 and 113.)

D. Engel and A. Uhl. An evaluation of lightweight JPEG2000 encryption with anisotropic wavelet packets. In E. J. Delp and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, Proceedings of SPIE, pages 65051S1–65051S10, San Jose, CA, USA, Jan. 2007a. SPIE. (Cited on pages 12, 113, 119, and 155.)

D. Engel and A. Uhl. An attack against image-based selective bitplane encryption. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '07*, volume II, pages 141–144, San Antonio, TX, USA, Sept. 2007b. IEEE. (Cited on page 175.)

D. Engel, R. Kutil, and A. Uhl. A symbolic transform attack on lightweight encryption based on wavelet filter parameterization. In *Proceedings of ACM Mul-*

timedia and Security Workshop, MM-SEC '06, pages 202–207, Geneva, Switzerland, Sept. 2006. (Cited on page 91.)

D. Engel, T. Stütz, and A. Uhl. Format-compliant JPEG2000 encryption in JPSEC: Security, applicability and the impact of compression parameters. *EURASIP Journal on Information Security*, (Article ID 94565), 2007a. (Cited on pages 8, 26, and 31.)

D. Engel, T. Stütz, and A. Uhl. Format-compliant JPEG2000 encryption with combined packet header and packet body protection. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '07*, pages 87–95, New York, NY, USA, Sept. 2007b. ACM Press. (Cited on pages 31 and 49.)

D. Engel, T. Stütz, and A. Uhl. Efficient transparent JPEG2000 encryption with format-compliant header protection. In *Proceedings of IEEE International Conference on Signal Processing and Communications, ICSPC '07*, pages 1067–1070, Dubai, UAE, Nov. 2007c. (Cited on page 49.)

D. Engel, E. Pschernig, and A. Uhl. An analysis of lightweight encryption schemes for fingerprint images. *IEEE Transactions on Information Forensics and Security*, 3(2):173–182, June 2008a. (Cited on pages 175 and 176.)

D. Engel, T. Stütz, and A. Uhl. Efficient transparent JPEG2000 encryption. In C.-T. Li, editor, *Multimedia Forensics and Security*, chapter 16. IGI Global, Hershey, PA, USA, 2008b. to appear. (Cited on page 10.)

R. Kutil and D. Engel. Methods for the anisotropic wavelet packet transform. *Applied and Computational Harmonic Analysis*, 2008. to appear. (Cited on pages 113, 114, 119, and 121.)

ACKNOWLEDGMENTS

Many people have contributed in different ways to this thesis. I am very grateful to all of them and want to thank several of them in particular.

I want to sincerely thank my advisor Andreas Uhl for his great support and valuable advice that he provided throughout my PhD studies. It is rare to find an advisor who is so competent in both, the subject area and the social aspects of supervising a PhD thesis. I was lucky to receive excellent tuition with insightful discussions, well-founded and constructive critique, exciting projects and challenges, the encouragement to submit my work to international review and with the right balance between guidance and the freedom to pursue new ideas and directions.

During the time I wrote this thesis, I received funding from different sources. I would like to thank the Austrian Academy of Sciences (ÖAW), from which I received a DOC dissertation grant, the Austrian Science Fund (FWF), which supported my work under projects P151790 and P19159, the European Network of Excellence in Cryptology ECRYPT, which provided a research grant, and the University of Salzburg, which employed me as a research assistant.

The good atmosphere at the Department of Computer Sciences at the University of Salzburg in general and the stimulating environment of the Wavelab research group in particular contributed a lot to the pleasure of writing my thesis. With Thomas Stütz I enjoyed many discussions and collaborations, during which we dug deeper into JPEG2000 and its environment than either of us might have done alone. Rade Kutil took me on a highly interesting journey to the area of the anisotropic wavelet packet transform. As the Salzburg BOWS-2 team, Heinz Hofbauer, Peter Meerwald, Thomas Stütz and I not only had great fun in breaking watermarks together, but also discussed many new ideas. I would like to thank them and all my colleagues for providing an atmosphere that made working a pleasure.

Parts of this thesis have been presented at conferences or published in journals. A big thank-you to all the colleagues in the field who provided feedback either as reviewers or during discussions at conferences.

I want to thank my parents for the support and advice that they have provided me with continuously for over 30 years now. Many thanks also to my brother Christopher, my grandmother, and all the members of my family – especially the recent additions in form of the Stassak family – for encouraging me and showing so much interest in the progress of my work.

Sincere thanks to my friends, who provided the social counterbalance to writing my thesis during many social events and a change of scenery during many visits and travels.

Finally, I want to thank my wife Linda. She has been and is an invaluable pillar in my life. She helped me through the more tedious phases of writing and listened patiently whenever I needed listening. It is to her that I dedicate this thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROLOGUE

1 INTRODUCTION

The aim of this thesis is the development and evaluation of encryption techniques for two classes of visual data: natural images and fingerprint data. The main focus with respect to the representational format is on scalable representations. JPEG2000 has been established as the most recent and most advanced standard for scalable still image coding and a large part of this thesis will focus on security techniques for visual data coded with JPEG2000. Some of the discussed security techniques are exclusive to JPEG2000, others are more generally applicable also in other formats, and we only use JPEG2000 as an exemplary format in this case.

Apart from proposing a selection of new media protection methods, a major focus of this thesis is the analysis of security flaws in existing media encryption schemes. One area where the severity of the compromise in security that is introduced by design mistakes is especially evident is the protection of biometric data, where revocation of (biometric) key data is usually impossible. We will discuss this issue at the example of a lightweight encryption scheme recently proposed for fingerprint images.

The topic of this thesis is situated in the field of media encryption, which in turn is a subtopic of multimedia security. In this introductory chapter, we will first briefly introduce both topics before we then present outline and organization of this thesis.

1.1 MULTIMEDIA SECURITY

Multimedia security is a young research area, dealing with various fields that share the common goal for providing security techniques for multimedia data. The individual research areas are widespread, but typically the following fields are mentioned among the core areas.

- *Media Encryption* provides the means for protecting multimedia files, e.g., for private communication. This is the subject area of the presented work, and we will introduce it in detail in the next section.
- *Watermarking* deals with embedding data into existing multimedia files. Different methods for watermarking are grouped into visible and invisible as well as

fragile and robust watermarking. In robust watermarking the watermark is designed to be detectable after modifications to the image that go beyond simple alterations like scaling or compression. An application for robust watermarking is the proof of ownership. In fragile watermarking, any modification destroys the watermark. An application for the fragile watermarking is tamper detection – the watermark serves as a digital seal.

- *Fingerprinting* uses invisible watermark embedding techniques to embed unique identifiers in a set of copies of multimedia data. An application of fingerprinting is the so-called *traitor tracing* that tries to detect the source of (e.g., stolen or illegally recorded) data.
- *Perceptual Hashing* maps a piece of multimedia data to a much smaller numerical set with the aim of giving a different hash to each different source content. Perceptual hashes differ from cryptographic hashes in that they only relate to the visual content – transcoding to a different representation format should not alter the hash. Two different kinds of hashes can be distinguished: authentication hashes that are used for example in proof of ownership and retrieval hashes that are used for classifying, indexing, and retrieving visual data, e.g., from a database.
- *Digital Forensics* develops techniques to expose forged material. It comprises techniques to verify or find the source that produced a piece of multimedia data, e.g., tests to determine with which camera model a digital image was taken. Another area is the detection of forgery in multimedia data, e.g., the use of photo editing programs to alter photographs.
- *Steganography* deals with methods to hide secret messages in inconspicuous covertexts. Other than watermarking, steganography can select or create the host material. The goal is to avoid detection and preserve the statistics of the original inconspicuous cover object. Steganalysis deals with the detection of secret messages in cover data.

There are many other other research topics related to multimedia security, but which are not subsumed under it. An example is the field of biometrics, which is often discussed as related to the field of multimedia security. This is not surprising as many of the objectives of biometrics can also be found in a similar way in multimedia security. Another point of overlap can be found in the representational formats that for prevalent biometric features, such as fingerprint and iris data are visual at least in one of the processing stages. In this respect, multimedia security and biometrics serve complementary objectives: while biometrics is concerned with the authentication part, methods from multimedia security can be used to secure the visual representation involved

in the authentication process. The discussion of security for fingerprint images reflects this overlap between the fields in this thesis.

Cultural and legal issues are related to multimedia security on a different level. As Delp (2005) argues in a pointed paper on the objectives and future of multimedia security, technical aspects are only one side of the issue and "new laws will need to be developed to combat this but also protect the rights of consumers" (p. 97). An example for the relation of these fields can be seen in the collection by Dittmann et al. (2006), which originated from a research program in the context of the EU-India Economic Cross Cultural Programmer. The papers in this collection discuss multimedia security as well as biometrics and also add a cultural perspective to the research.

There are several international conferences dedicated to multimedia security. Uhl and Pommer (2005) list the following conferences as the core forums: "ACM Multimedia & Security", which was first held in 1998, "Security, Steganography and Watermarking of Multimedia Content", a subconference of the SPIE Electronic Imaging symposium, and "Communications and Multimedia Security (CMS)", organized by IFIP. A comprehensive textbook with contributions of many researchers working in the field has been edited by Furht and Kirovski (2005). Dittmann et al. (2000) discuss approaches to multimedia and security and describe the security requirements that arise in modern multimedia systems.

Apart from a number of special issues that have been devoted to the field of multimedia security, a number of journals dedicated exclusively to the topics of multimedia security have been established recently. The IEEE *Transactions on Information Forensics and Security*[1] first appeared in 2006. The EURASIP open access *Journal on Information Security*[2] first appeared in 2007. IET *Information Security*[3] first appeared in 2005 (but before 2007 its name was "IEE Proceedings – Information Security").

1.2 MEDIA ENCRYPTION

Media encryption has as its goal the development of security schemes that differ from conventional encryption in that they are tailored to the representational format of the source media. Conventional encryption schemes are oblivious to what is being encrypted. Specific encryption schemes for media encryption take the representation into account and therefore can provide additional features. One feature could for example be the possibility to perform rate-adaption in the encrypted domain. In turn, media encryption scheme almost always sacrifice some of the security that can be obtained with full classical encryption.

[1] http://www.signalprocessingsociety.org/publications/periodicals/forensics/
[2] http://www.hindawi.com/journals/is/
[3] http://www.ietdl.org/IET-IFS

Several of the published review papers are located in the field of MPEG encryption, providing a more or less complete overview of the techniques proposed so far. Kunkelmann (1998) and Qiao and Nahrstedt (1998) provide overviews, comparisons, and assessments of classical encryption schemes for visual data with emphasis on MPEG proposed up to 1998. Bhargava et al. (2004) review four MPEG encryption algorithms published by the authors themselves. The survey by Lu and Eskicioglu (2003) focuses on shortcomings of current schemes and future issues, the one by But (2004) assesses the suitability of available MPEG-1 ciphers for streaming video, and Lookabaugh et al. (2003) focus on cryptanalysis of MPEG-2 ciphers.

In this section we give something of a taxonomy of encryption schemes that exist in the area of media encryption. As the field is rather young, there is not a substantial number of comprehensive review books that specifically deal with media encryption. The book by Uhl and Pommer (2005) is one of the first books dedicated exclusively to the field. Another consequence of media encryption being a young field is that the nomenclature is not yet fully consistent. We will mention if different terms are used in parallel or if a term is used with different meanings in the following.

1.2.1 *Soft, Partial, and Lightweight Encryption*

One way to classify media encryption is by which cryptographic primitives are used for encryption.

Soft encryption versus hard encryption

In *soft encryption*, as opposed to *hard encryption*, cryptographic primitives are used that are less demanding than classical cryptographic primitives. These primitives lower the computational demands of the encryption, but of course they also lower security.

Partial / Selective Encryption

Another distinction is, if only some parts of the source data are encrypted and other parts are left in plaintext. In *partial* and *selective encryption* only some portions of the source data are encrypted. The classical motivation for this is to lower computational demands, but the reason to use partial / selective encryption could also be to leave some parts in plaintext (e.g., the base layer) that can be used to decode a preview image (i.e., provide transparent encryption, see below). Partial / selective encryption may also target only certain parts of the bitstream, e.g., packet bodies as opposed to packet headers, to facilitate functionality for the encrypted bitstream, e.g., format-compliance (see Section 1.2.3). The difference between partial and selective encryption is academic. Sometimes partial encryption is defined as comprising only schemes that

encrypt one contiguous part in the source data, whereas selective encryption might select different portions of the source data with plaintext data in between. However, mostly the two are used synonymously.

Said (2005) gives a critical view on partial / selective encryption schemes, especially under the consideration of the availability of side channels (such as thumbnail images).

In the area of JPEG2000, there is a number of approaches: for example, to employ a traditional cipher to encrypt different parts of the packet data has been proposed, e.g., by Grosbois et al. (2001); Wu and Deng (2004) and Norcen and Uhl (2003). Ou et al. (2007) propose two methods for region-based selective encryption in the context of medical imaging.

Lightweight encryption

The term *lightweight encryption* is used to denote encryption schemes which (deliberately) provide security on a much lower level than classical cryptography. In some settings, there might even exist relatively efficient attacks against these schemes. These schemes provide limited access control for a limited time and against attacks of limited computational resources. In turn they are usually computationally very cheap (sometimes they come nearly for free) and provide high levels of functionality. Note that lightweight encryption can theoretically be combined with partial encryption to denote an encryption scheme that provides only little security by processing only parts of the source data.

An example for lightweight encryption in JPEG2000 is the permutation of the order of wavelet coefficients during compression (Norcen and Uhl, 2004a). The computational demands of this approach, although of course higher than the operation of unencrypted JPEG2000, stay far below full AES encryption and can be tuned to preserve some scalability properties of the unencrypted JPEG2000 bitstream.

1.2.2 *Levels of Security*

Depending on the intended application scenario, encryption schemes may offer different levels of security.

Cryptographic security / Full confidentiality

Cryptographic security means that the security provided by the scheme is equivalent to security provided by a classical crpytographic cipher, e.g., AES. No information on the visual data may be leaked through the ciphertext. This high level of security can usually only be achieved if actually a cryptographically strong cipher is used to encrypt

the full bitstream. Other terms for cryptographic security are *full security*, *full privacy*, *full confidentiality*, and also *conventional encryption*.

The DVD encryption scheme based on CSS was meant to provide this property; however, a weak cipher turned the scheme unintentionally into soft encryption (see below). In multimedia applications a digression from full cryptographic security is often deliberately accepted, if it comes with an increase in functionality or greatly reduces computational demands.

Content security

Content security, a term introduced by Thomas Stütz, refers to security schemes in which the visual data is not accessible in any form. No visual information, like shape or color information, may be derived from the plaintext.

However, it may be possible to link ciphertext and plaintext. It could, for example, be possible to find the corresponding plaintext of a given ciphertext from a set of a million images. For example, in Engel et al. (2007a) we have shown that this is possible in the context of format-compliant encryption schemes for JPEG2000 by using header information that remains in plaintext. Of course, such a fingerprint in the ciphertext is not a property that is desirable in high security applications. In high security applications, no link between ciphertext and plaintext should exist (arguably apart from maybe the filesize). This is important for *deniability*: if Alice wants to keep a ciphertext for her archives after exchanging visual data with Bob, she does not want Bob to be able to later link the plaintext image with the ciphertext in her archive (which he could, even if Alice used a different encryption key for the archived ciphertext than for the communication with Bob).

Sufficient encryption

A scheme provides security of this level, if visual information can be decoded without key from the ciphertext, but the quality of the visual data gained in this way is too low for an enjoyable viewing experience. This type of security often is enough for media applications, in which the existence of a distorted version of the source media does not imply a security threat.

Apart from encryption schemes that are deliberately targeted at sufficient encryption, another group of algorithms can be categorized into this class: encryption algorithms, which were originally meant to provide cryptographic security, but for which security flaws have been found that allow to obtain visual information. Examples are many partial / selective encryption schemes that were proposed to provide security equivalent to full encryption but for which it turned out that discernible variants of

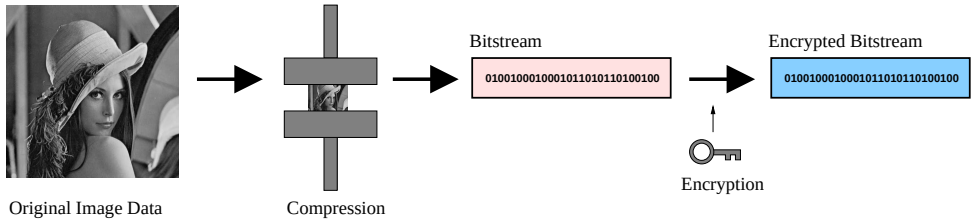
the visual data could be decoded from the portions of the data that were left in plaintext (and that had been meant to be useless without the encrypted parts).

Note that the distinction between sufficient and transparent encryption is slight, and lies in the minimum quality that can be decoded without the key. Some authors subsume sufficient encryption under transparent encryption.

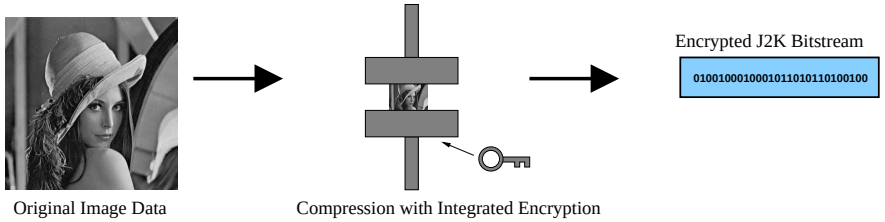
Transparent encryption

The term was introduced by Macq and Quisquater (1994, 1995) in the context of TV broadcasting and denotes encryption schemes for which public access is granted for a preview image, i.e., anyone can decode an image of reduced quality from the encrypted stream, even without the key data. The difference to sufficient encryption is that the preview image has to be of a (specified) minimum quality, i.e., apart from the *security requirement*, there is also a *quality requirement* (cf. Stütz and Uhl, 2007). Broadcasting applications, for example, can benefit from transparent encryption, as they, rather than preventing unauthorized viewers from receiving and watching their content completely, aim at promoting a contract with non-paying watchers, for whom the availability of a preview version (in lower quality) may serve as an incentive to pay for the full quality version.

Numerous contributions for transparent encryption of various formats exist. Droogenbroeck and Benedett (2002a) propose to encrypt the LSB bitplanes of the binary representation of raw image data. With respect to JPEG encoded images, the authors suggest to encrypt sign and magnitude bits of medium and high frequency DCT coefficients. Droogenbroeck (2004) extends the latter idea to "multiple encryption", where different sets of DCT coefficients are encrypted by different content owners, and "over encryption", where these sets do not have an empty intersection (i.e., coefficients are encrypted twice or even more often). In the context of scalable or embedded bitstreams, transparent encryption is achieved by simply encrypting the enhancement layer(s). This has been proposed by Kunkelmann (1998) and Kunkelmann and Horn (1998) who use a scalable video codec based on a spatial resolution pyramid. Dittmann and Steinmetz (1997a,b) propose using a SNR scalable MPEG-2 encoder/decoder. Yuan et al. (2003) propose to use MPEG-4 FGS for transparent encryption in the same manner. Stütz and Uhl (2006a) and Fisch et al. (2004) propose methods for progressive JPEG. For JPEG2000, Uhl and Obermair (2005) propose to encrypt packet data from the end of the bitstream, which corresponds to encryption of enhancement layers. Since the major part of the packet body data needs to be encrypted, this approach is demanding in terms of computational complexity. We will propose more efficient methods in Parts II and III. Stütz and Uhl (2007) discuss efficient methods for transparent encryption of JPEG2000. A comparison of classical



(a) Bitstream-oriented encryption



(b) Compression-integrated encryption

Figure 1.1: Bitstream-oriented versus compression-integrated encryption

bitstream-oriented approaches to the encryption of JPEG2000 to some of the methods proposed here can be found in Engel et al. (2008b).

1.2.3 *Format-compliant Encryption*

A format-compliantly encrypted bitstream still complies to the syntactical rules of the original visual format. The encrypted bitstream can be decoded with a standard decoder (but of course the resulting reconstruction will not be the original media plaintext). For scalable formats, for example, this means that tasks like rate adaptation can be performed directly on the encrypted media without the need for decoding.

In the context of JPEG2000 encryption, there is a number of approaches that propose format-compliant encryption: Grosbois et al. (2001); Wu and Mao (2002); Sadourny and Conan (2003); Kiya et al. (2003); Dufaux and Ebrahimi (2003); Norcen and Uhl (2003); Wu and Deng (2004); Wu and Ma (2004); Conan et al. (2005); Deng et al. (2005); Liu (2006); Stütz and Uhl (2006b); Grangetto et al. (2006); Yang et al. (2007). Segment-based encryption as proposed by Wee and Apostolopoulos (2001a,b) does not fulfill the requirement for full format-compliance in the strict sense, as in this case rate-adaption can only be performed by a JPSEC-compliant decoder. In Part I, we will turn to format-compliant encryption of JPEG2000 in more detail.

1.2.4 *Compression-oriented versus Bitstream-oriented Encryption*

Depending on whether encryption is applied during compression or operates on the compressed codestream[4] after compression, two classes of media encryption schemes can be distinguished, as illustrated in Figure 1.1.

Bitstream-oriented Encryption

Bitstream-oriented techniques only operate on the finished codestream. Although they may parse the codestream and for example use meta-information from the codestream, they do not access the encoding (or decoding) pipeline. Classical methods for encryption fall into this category, and also many selective / partial encryption that only encrypt parts of the bitstream.

Header encryption, which has the goal of securing meta-information in the codestream, is also classified as bitstream-oriented encryption. We will present a method for header encryption in the first part of this thesis.

Compression-integrated Encryption

Compression-integrated techniques apply encryption as part of the compression step, sometimes going so far that part of the compression actually is the encryption. One possibility is to apply classical encryption after the transform step (which in most

[4]Note that to be exact, one should not use the terms "codestream" and "bitstream" as synonyms. The meaning of "bitstream" is simply an arbitrary concatenation of bits. A "codestream" adds syntactical constraints to the organization of a bitstream. In the context of JPEG2000, "bitstream" has an alternative definition:

> The actual sequence of bits resulting from the coding of a sequence of symbols. It does not include the markers or marker segments in the main and tile-part headers or the EOC marker. It does include any packet headers and in stream markers and marker segments not found within the main or tile-part headers.
> (ISO/IEC 15444-1, 2000, p. 2)

Whereas "codestream" is defined as

> A collection of one or more bit streams and the main header, tile-part headers, and the EOC required for their decoding and expansion into image data. This is the image data in a compressed form with all of the signalling needed to decode.
> (ISO/IEC 15444-1, 2000, p. 3)

We do not strictly adhere to these definitions throughout this thesis (as is evident in the present section on "bitstream-oriented encryption" which clearly should better be termed "codestream-oriented encryption" according to the general meanings). It should be clear from the context if we refer to a bitstream with or without syntactical constraints.

cases inevitably destroys compression performance). For other approaches the transform step is also the encryption step at the same time. Norcen and Uhl (2004a) propose compression-integrated encryption by permutation of zerotrees. An example of compression-integrated encryption for JPEG2000 are methods that change the mechanics of the arithmetic coder (Grangetto et al., 2006; Liu, 2006).

Another possibility to achieve compression-integrated encryption is by selecting the transform domain to be used for encoding based on a key. The principal idea of such schemes is that without the key the transform coefficients cannot be interpreted or decoded and therefore no access to the source material is possible (or only a construction of limited quality is possible if sufficient or transparent encryption is the goal).

Fridrich et al. (1998) introduce the concept of key-dependent basis functions to protect a watermark from hostile attacks. This approach suffers from significant computational complexity. Fridrich (1999) develops the idea further and proposes a faster method for the generation of key-dependent orthogonal patterns. Cancellaro et al. (2007) propose a technique for data hiding using a key-dependent basis function in the tree structured Haar transform domain. There are also some propositions that use secret Fourier transforms: The embedding of watermarks in an unknown domain is discussed by Djurovic et al. (2001), and Unnikrishnan and Singh (2000) suggest to use this technique for encryption of visual data. Vorwerk et al. (2000) propose the encryption of the filter choice used for a wavelet decomposition. However, this suggestion remains vague and is not supported by any experiments. Pommer and Uhl (2001) propose encrypting the filterbanks used for an NSMRA decomposition. The use of parameterized wavelet filters for lightweight encryption is proposed by Uhl and Pommer (2004), the use of key-dependent wavelet packet decompositions is proposed by Pommer and Uhl (2002, 2003).

In the context of JPEG2000, the degrees of freedom in the wavelet transform are a prime candidate for constructing a secret transform domain. Parameterized wavelet filters have been employed for JPEG2000 lightweight security by Köckerbauer et al. (2004) and Engel and Uhl (2005a,b). Key-dependent wavelet packet structures in JPEG2000 have been discussed for the same purpose by Engel and Uhl (2006a,b, 2007a). The proposed schemes can be seen as a form of header encryption, as only the information pertaining to the frequency domain needs to be encrypted, the rest of the data remains in plaintext. This approach has the advantage that only the parameters of the secret transform domain needs to be kept secret, so the demands for the encryption stage are minimal. Another advantage is that these approaches are well suited for signal processing in the encrypted domain, a research area that has gained a lot of attention recently, simply because the encrypted domain *is* a transform domain. We will propose, discuss and evaluate compression-integrated encryption techniques that use key-dependent transform domains in Parts II and III.

1.2.5 *On-line versus Off-line encryption*

In applications where the data is acquired before being further processed the plain image data may be accessed directly for encryption after being captured by a digitizer. Following Uhl and Pommer (2005), we denote such applications as *on-line*. Examples for this scenario are video conferencing and on-line surveillance. On the other hand, as soon as visual data has been stored or transmitted once, it has usually been compressed in some way. Applications where compressed bitstreams are handled or encrypted are denoted as *off-line*. Examples are video on demand and retrieval of medical images from a database.

1.2.6 *Other Functionality*

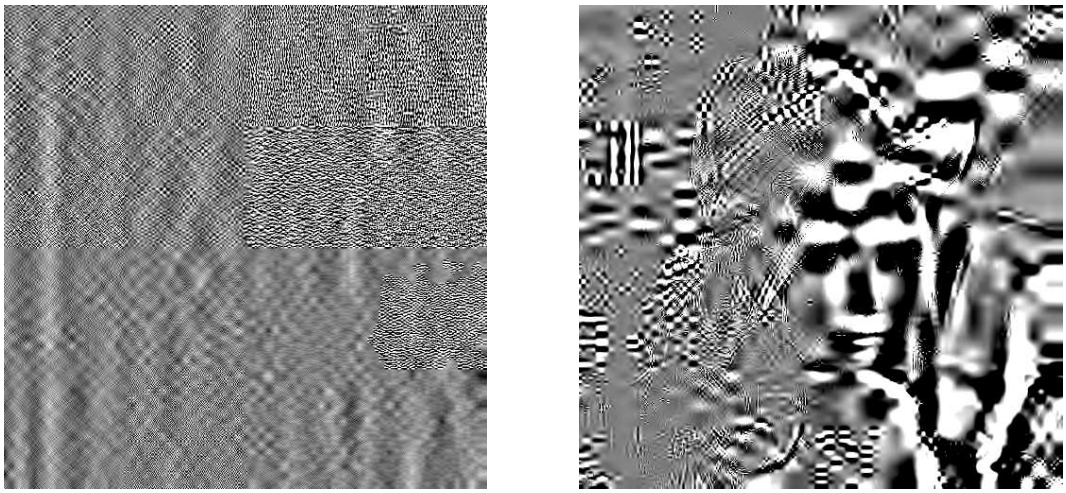
We have already mentioned format-compliant encryption and transparent encryption as features that might be traded in for security in (lightweight) encryption schemes. Other functionality might include the reduction in computational demands or the possibility to do signal processing in the encrypted domain.

Reduced computational demands for encryption is especially interesting in mobile environments, where at least one party in the communication uses processors of low processing capabilities. In order to still give a certain level of security in these environments, encryption schemes that significantly lower computational demands (e.g., through selective encryption) are desired. In the vast majority of cases this implies a reduction in security.

Signal processing in the encrypted domain is a functionality that has recently received a lot of interest. For example, a special issue of the EURASIP Journal on Information Security edited by Piva and Katzenbeisser was dedicated to the topic in 2007. The possibility of performing signal processing in the encrypted domain allows to apply a (restricted) set of operations on the encrypted bitstream, with the same effect as if the set (or an equivalent set) of operations had been applied to the plaintext bitstream. This allows untrusted nodes in networked communication, for example, to perform adaptations without the need of a key.

1.3 QUALITY METRICS

A lot of quality metrics for assessing visual quality have been proposed in recent years. The high number of proposals is due to the fact that so far no metric has been found to be reliable in all application contexts. The peak-signal-to-noise-ratio (PSNR) is the most widespread quality metric. However, the PSNR is inadequate under a number



(a) PSNR 9.2 dB (b) PSNR 9.2 dB

Figure 1.2: An example for PSNR being unsuited for assessing images of extremely low quality

of conditions (e.g., shifting or shearing), among them assessing the quality of reconstructed or attacked images with high distortion levels.

Because in media encryption, images with very high distortion levels are typically obtained by attacks, the PSNR is a very mediocre measure in this case. We have discussed full confidentiality and content security above. In schemes that aim at providing full confidentiality or content security, two attacked images of very low quality might have the same PSNR, but in one image some contours are preserved whereas in the other image, no visual content can be discerned. Even though these two images have the same PSNR, one of them invalidates the associated encryption scheme, that strived for content security or even full confidentiality. The situation is illustrated in Figure 1.2 for attacked encrypted Lena images: both images have exactly the same PSNR when compared to the unencrypted source image, but whereas for Image 1.2(a) no visual information can be discerned, Image 1.2(b) reveals definite facial features (the examples were created with the JPEG2000 header protection scheme discussed in Part I). In this section, we give a brief overview of some metrics for visual quality that have been suggested as alternatives to PSNR.

In the context of evaluating watermarking systems, Kutter and Petitcolas (1999) discuss a variety of measures for visual quality. In a section on pixel-based metrics they present a list of *difference distortion metrics*, *correlation distortion metrics*, and other pixel-based metrics. They also give a brief discussion of perceptual quality metrics. Under difference distortion metrics, Kutter and Petitcolas (1999) subsume Maximum

Difference, (Normalized) Average Absolute Difference, (Normalized) Mean Square Error, L^p-Norm, Laplacien Mean Square Error, Signal to Noise Ratio, Peak Signal to Noise Ratio, and Image Fidelity. The authors identify this group of measures as the ones most often used in visual information processing, with the PSNR being the most popular. An assessment of difference distortion metrics can be found in Eskicioglu et al. (1993).

Kutter and Petitcolas (1999) argue that while sophisticated watermarking method exploit the human visual systems (HVS) in one way or the other, this is not reflected in the difference-based metrics. Therefore, for a fair benchmark of watermarking systems, they make the case for perceptual quality metrics which are adapted to the HVS. This statement need not be limited to watermarking systems. Generally, the measures that are mostly used to assess the quality of visual data, like the Peak Signal to Noise Ratio (PSNR), are not well correlated with findings that relate to the working of the human visual system. Therefore, most "objective" measures recently suggested for the assessment of visual quality incorporate HVS-models to some degree. The degree varies to which the HVS-model is used as reference and reflected in the quality measure. Many of the suggested approaches also include some reference to subjective evaluation: the success of many of the proposed metrics is evaluated by a comparison to subjective quality tests. Some approaches also incorporate data from subjective tests to fine-tune the proposed quality metrics.

Lambrecht and Verscheure (1996) suggest a metric based on a multi-channel spatio-temporal model of the HVS, that has been parameterized by psychophysical experiments. Winkler (1998) proposes a perceptual distortion metric (PDM) for color images based on a model of the human visual system that incorporates color perception. In Winkler (2000) the suggested PDM is evaluated in more detail. The author reports results from a subjective evaluation and reports good correlation of the presented measure with subjective data for natural images. In Winkler et al. (2001) a perceptual distortion metric is discussed that accommodates mechanisms of both metrics: color perception, temporal and spatial mechanisms in multiple channels, contrast sensitivity, and pattern masking in each channel.

Wang and Bovik (2002) argue that as the main function of the human eyes is to extract structural information from the visual input, a good measurement for quality should regard structural distortion. They propose a simple quality index which models distortions as a combination of three factors: loss of correlation, mean luminance distortion and variance distortion. This quality index is applied to the whole image with a sliding window approach, leading to a quality map of the image. The quality index of the image is defined as the average of the quality map. The resulting measure is called *structural similarity index (SSIM)*.

There is an abundant amount of other approaches to quality measures of visual data. Some of them lean more towards the difference-error-based approaches, for example,

Caramma et al. (2000) propose a measure for video quality similar to PSNR that is weighted by motion vectors and evaluated in subjective tests. On the other side of the spectrum there are measures that try to incorporate a model of the HVS as closely as possible. For example, Carnec et al. (2003) propose a "perceptual subband decomposition" that aims at simulating the working of the perceptual system and incorporate neurocognitive pathways in their measure.

The quality metrics mentioned so far are targeted towards images of sufficient quality, and are therefore of limited use to assess the quality of attacked images in media encryption. Mao and Wu (2004) propose an alternative measure for low quality images that separates evaluation of luminance and edge information into a *luminance similarity score* (LSS) and an *edge similarity score* (ESS), reflecting properties of the human visual system. According to the authors, this measure is well suited for assessing distortion of low quality images, which are typically obtained by attacks on encrypted visual data. The more similar two images are in terms of their luminance information, the closer LSS is to 1. Negative values of LSS reflect significant dissimilarities in luminance. ESS is computed by block-based gradient comparison and ranges, with increasing similarity, between 0 and 1. The authors use Sobel edge detection to determine the gradient of a given block.

The influence of a perceptual model in LSS/ESS is weak. The proposed measure is in fact very similar to PSNR in the LSS component. However, the added component of edge similarity is an interesting feature. ESS is the more interesting part in the context of the work presented here, as it reflects the extent for structural distortion. Therefore, for a number of evaluations in this thesis, we will use the ESS as a quality measure along with the PSNR. We use the weights and blocksizes proposed by Mao and Wu (2004) in combination with Sobel edge detection. However, not too much hope should be given to this quality measure to present a substantial improvement over PSNR. For the two images in Figure 1.2, the ESS scores are very close with 0.24 for 1.2(a) and ESS 0.27 for 1.2(b) (but at least there is a difference, as opposed to the PSNR scores).

1.4 ORGANIZATION OF THIS THESIS

This thesis is organized in four parts, with a prologue and an epilogue. Each of the main parts deals with a different method for media encryption: bitstream-oriented format-compliant JPEG2000 encryption, lightweight compression-integrated encryption with parameterized wavelet filters, compression-integrated encryption with key-dependent subband structures, and bitplane-based encryption.

As already mentioned, we chose JPEG2000 not only as the main representational format but furthermore also partly as the object of our research. This wavelet-based compression standard provides rich functionality and scalable representation. Encryp-

tion techniques tailored to JPEG2000, which can preserve some of its functionality, form a major part of this thesis. As visual data encoded with JPEG2000 is the subject of investigation in several chapters, we give an overview of the standard in Chapter 2.

Part I is on format-compliant encryption of JPEG2000. It has been shown that previously proposed format-compliant encryption approaches (both bitstream-oriented and compression-integrated) leak information that can compromise the security level of the scheme. In some cases the extent of leaked information is so large that even schemes that only aim at providing content security are compromised. We review this situation and propose a method for header encryption to alleviate the information leakage.

In Parts II and III the focus stays on format-compliant JPEG2000 encryption, and we turn to compression-integrated methods that use key-dependent transform domains. We propose and evaluate two secret transform domains in the context of lightweight JPEG2000 encryption. In Part II, parameterized wavelet filters are used to create a key-dependent transform domain. In Part III, we investigate the utility of key-dependent subband structures, in our case isotropic and anisotropic wavelet packets, for the same purpose. In both parts we first present the method, investigate compression performance and keyspace size and quality and then discuss possible attacks and applicability.

Part IV to some degree stands apart from the other parts. We turn away from JPEG2000 and to the security of fingerprint data, which in the other parts we only use as examples (one of the reasons for which is that fingerprint images have been reported to be well suited for compression with wavelet packets). We analyze a recently published lightweight encryption scheme that is targeted at fingerprint images (but could be used for natural images as well). We show that the security of the scheme is insufficient and present an efficient attack that results in a total break.

In the final part, the epilogue, we summarize our results, give an outlook on possible future research directions and conclude.

THE JPEG2000 STANDARD 2

As many of the security techniques we discuss in this thesis directly or indirectly relate to the JPEG2000 standard, in the following chapter we will give an overview of JPEG2000. There are many sources providing comprehensive information on the JPEG2000 standard: There are the documents of the ISO and ITU-T (ISO/IEC 15444-1, 2000; ITU-T T.800, 2002) and the standard book by Taubman and Marcellin (2002). The book by Acharya and Ray (2004) also provides a good introduction. Christopoulos et al. (2000) provide an overview of part 1 of the standard. In the following brief introduction we will only refer to those portions of each part of the standard that are relevant to our work.

2.1 BASICS

One of the most important properties of JPEG2000 is that it is a *scalable* format. Scalability means that source data is encoded only once but can be decoded in multiple ways to fit different needs. This is important, for example, for serving a variety of commonly used display devices ranging from PDAs and mobile phones to HDTV cinema displays – each with distinct capabilities, e.g., in terms of display resolution and computing power. In a scalable format, the same encoded bitstream can be used to decode versions of the original source media in the suitable resolution or quality, or, if we are dealing with video data, framerate. Other modes of scalability exist, but resolution and quality scalability are the most important modes in the context of still visual data. The different modes of scalability can be combined, e.g., to form a bitstream that is scalable in both resolution and quality.

Two – sometimes contrary – requirements have to be fulfilled. On the one hand, it is important that the scalable coding is not less efficient than the coding for a target resolution or bitrate. The embedded bitstream, which can be cut off to produce a target rate or resolution, should not be (much) larger than the bitstream produced for the full rate or resolution. On the other hand, in an embedded bitstream we expect each of the versions of lower bitrates and resolutions to be efficiently coded, (nearly) as efficiently as if the embedded rate or resolution had been the overall target resolution or rate. It is clear that realizing full scalability is a tough problem, and it is impres-

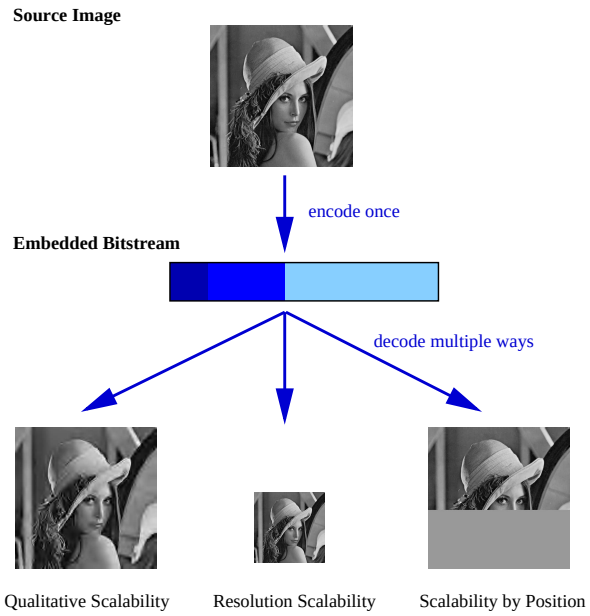


Figure 2.1: Scalable coding

sive to what extent JPEG2000 realizes scalability (albeit the embeddedness is not of arbitrary granularity). JPEG2000 supports both resolution scalability and quality scalability. Resolution scalability is realized through the multiresolution property of the wavelet transform. For each resolution, qualitative scalability is realized by introducing (a discrete set of) quality layers. JPEG2000 also provides other modes of scalability, e.g., scalability by component or position.

Information on the history of the JPEG2000 standardization process can be found in Taubman and Marcellin (2002). The first call for contributions to JPEG2000 were issued in March 1997. An ad hoc working group lead by Touradj Ebrahimi was instantiated to formulate the requirements. The requirements that were established by the group included

- superior low bit-rate performance,
- progressive transmission by pixel accuracy and resolution,
- lossless and lossy compression, and
- random code-stream access and processing.

The decision for a wavelet-based approach was made soon during the standardization process in December 1997. Through a series of verification models JPEG2000 was de-

veloped between the years 1998 and 2000. In December 2000, Part 1 of the JPEG2000 standard was passed.

Apart from the name, JPEG and JPEG2000 do not have much in common. Where JPEG is based on the discrete cosine transform (DCT), JPEG2000 is based on the wavelet transform. JPEG2000 uses mechanisms for coefficient encoding which are a lot more complex to create a scalable bitstream. Furthermore, JPEG2000 performs especially well at (extremely) low bitrates and outperforms JPEG by an order of magnitude or more. Other than with JPEG and JPEG-LS, in JPEG2000 lossy and lossless compression are integrated into the same algorithm. Contrary to other image coding standards, JPEG2000 goes beyond the specification of the mere image coding procedure, and in different parts (see below) provides a whole suite of standardized methods for diverse areas of application, like motion JPEG2000, file formats or interactivity tools (JPIP).

On a more technical note, JPEG2000 also differs from other wavelet-based image codecs like EZW (Shapiro, 1993) or SPIHT (Said and Pearlman, 1996) in its principal approach. While the latter are based on the zerotree hypothesis, JPEG2000 codes the wavelet coefficients in groups called codeblocks. Each codeblock is coded independently of the other blocks and an embedded bitstream is produced for each of them. In a subsequent step, rates are allocated to each of the bitstreams and a progression order is fixed. This allows for scalability on different levels, which arguably is one of the most important features of JPEG2000.

Although JPEG2000 was intended as the successor of JPEG, rather than replacing JPEG it filled areas of application that JPEG could not provide for, especially where applications require a scalable representation of the visual data. It took some time for JPEG2000 to really gain momentum, but recently JPEG2000 has evolved into the format of choice for many specialized and high end applications. For example, the Digital Cinema Initiative (DCI), an entity created by seven major motion picture studios, has adopted JPEG2000 as the (video!) compression standard in their specification for a unified Digital Cinema System (Digital Cinema Initiatives (DCI), LLC, 2007). JPEG2000 is well suited for digital libraries and has for example been evaluated for Google Books (Langley and Bloomberg, 2007). A very recent report by the Digital Preservation Coalition investigates JPEG2000 as a possible standard format for digital preservation (Buckley, 2008). For biometric data in the new generation of EU passports, JPEG2000 is the recommended format for digitally storing the portrait image as a biometric feature on the passport[1]. In the area of medical imaging JPEG2000 provides important functionality like region of interest coding (Foos et al., 2000; Norcen

[1]The specification for the EU-passports has been adopted by 18 member states. See http://ec.europa.eu/justice_home/doc_centre/freetravel/documents/doc/c_2006_2909_de.pdf [accessed on 18 March 2008] for the German version. As the United Kingdom and Ireland so far have not adopted the specification, the English version is available only as a "working document",

et al., 2003). As Buckley (2008) states, "[i]nstead of replacing JPEG, [JPEG2000] has created new opportunities in geospatial and medical imaging, digital cinema, image repositories and networked image access" (p. 2).

2.2 PARTS

The JPEG2000 standard is a comprehensive set of documents that cover all areas of still image coding, but also standardize advanced functionality. There are 13 parts, numbered 1 to 6 and 8 to 14, because part 7 has been abandoned and integrated into another part. Part 1 of the standard deals with the core coding system. The content of this part follows the tradition of image coding standards and exclusively focuses on image coding. Part 2 extends this functionality with more flexible mechanisms for image coding, like arbitrary subband decomposition structures, and custom wavelet filters. The other parts focus on different areas like volumetric imaging (Part 10), wireless applications (Part 11) or XML representation (Part 14). As of this writing all parts are finished with the exception of Parts 10 and 14. Of the 13 parts, apart from Part 1 and Part 2 two parts are relevant to the work presented here: Part 4, which deals with conformance testing and specifies when a codestream can be denoted format-compliant, and Part 8 (JPSEC), which deals with security aspects in JPEG2000.

2.2.1 *Part 1: Core Coding System*

JPEG2000 Part 1 (ISO/IEC 15444-1, 2000; ITU-T T.800, 2002) defines the core coding system. As already mentioned, JPEG2000 breaks with the tradition of wavelet-based codecs to utilize the zerotree hypothesis. The basic zerotree hypothesis was formulated by Shapiro (1993), and later generalized by Said and Pearlman (1996). According to this hypothesis child coefficients in a subband of higher resolution are more likely to be insignificant if their parent in the subband of lower resolution at the same spatial location is also insignificant. The good compression performance of wavelet-based schemes was attributed to a significant extent to the zerotree hypothesis.

Taubman and Marcellin (2002) argue, however, that although the children of insignificant coefficients tend to be insignificant, "this is more a property of the energy compaction property of good transforms, and less a property of specific image data dependencies" (p. 323). They show that compression performance of SPIHT remains competitive even if the detail subbands are alternately transposed (and thus the parent-child-relationships destroyed). Good compression results obtained after permuting zerotrees as reported by Norcen and Uhl (2004a) support this observation. In JPEG2000

see http://ec.europa.eu/justice_home/doc_centre/freetravel/documents/doc/c_2006_2909_en.pdf [accessed on 18 March 2008].

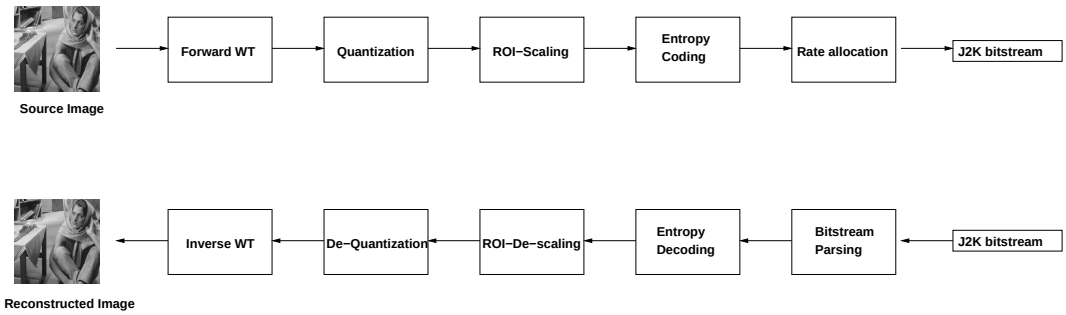


Figure 2.2: The JPEG2000 coding pipeline

zertotree coding is replaced by embedded block coding, based on a concept called Embedded Block Coding with Optimal Truncation (EBCOT) which was proposed by one of the later developers of JPEG2000 (Taubman, 2000).

The coding pipeline, which is separated into tier 1 and tier 2 coding, is shown in a simplified version in Figure 2.2. After the source image has been acquired, it undergoes tiling and component transform. For both lossless and lossy compression a wavelet transform is performed. Part 1 defines two kinds of wavelet filters: in the lossless case a reversible 5/3 integer wavelet is used, in the lossy case the non-reversible 9/7 CDF wavelet is used. In JPEG2000 Part 2 more filters are possible.

After the wavelet transformation codeblocks are formed. Note that from this point onwards, each codeblock is coded independently of the other codeblocks. The next step is quantization (only for the lossy case, of course). In this step the wavelet coefficients are mapped to quantization indices. There are two ways of signalling the quantization step size, *expounded* and *derived*. In chapter 5, we will turn to the two signalling methods again in the context of parameterized wavelet filters.

The next step in the coding pipeline is region of interest (ROI) scaling. In this step the bitplane representation of the (quantized) wavelet coefficients that are in a region of interest can be shifted to give them higher prioritization.

Entropy coding marks the end of tier 1 coding. The transform coefficient data are coded with an adaptive arithmetic coding scheme, the MQ Coder.

Tier 2 coding refers to the packetization and rate allocation. The rate-distortion characteristics are computed for each codeblock and valid truncation points are defined. Depending on the requested number of quality layers, the contribution of the codeblocks to each quality layer are defined (as illustrated by Figure 2.3). The contributions are portioned into units called JPEG2000 packets. Each packet corresponds to a quality layer at a certain resolution (or precinct to be exact), as illustrated by Figure 2.4.

The notion of packets will be important for Chapter 3. Finally, the packets are ordered according to the requested progression order and the final bitstream is written.

Note that the standard only defines the elements of a valid JPEG2000 bitstream and how they have to be interpreted. Accordingly, the coding pipeline could (theoretically) be realized differently.

2.2.2 *Part 2: Extensions*

Part 2 of the JPEG2000 standard (ISO/IEC 15444-2, 2004; ITU-T T.801, 2002) is termed "Extensions". It adds additional and, one could say, more exotic options. Part 2 gives a wide range of options to fine-tune JPEG2000 processing. For example, it introduces variable DC offset, variable scalar quantization and trellis coded quantization and allows to specify multiple component transformations as well as non-linear transformation and single sample overlap discrete wavelet transformations. Furthermore it provides for visual masking to enhance image quality especially for displays. It also defines an extended fileformat and metadata definitions.

The extensions introduced by part 2 that are relevant to the work presented here are arbitrary wavelet transforms and arbitrary decomposition structures. The former introduces the possibility to define custom wavelet filters to be used for transformation. This not only allows for transcoding of the output of other wavelet-based compression algorithms to JPEG2000, but can also be used as a method for providing lightweight security, as we will discuss in detail in Part II. JPEG2000 allows to specify arbitrary wavelet packet decomposition structures used for transformation, both isotropic and anisotropic. As a major part of our work relates to wavelet packets, we will discuss this in more detail in Part III.

2.2.3 *Part 4: Conformance Testing*

JPEG2000 Part 4 (ISO/IEC 15444-4, 2004; ITU-T T.803, 2004) deals with "conformance testing". This part specifies compliance testing procedures for encoding and decoding JPEG2000 (Part 1) codestreams. In the context of our work this part of the standard is important because it gives guidelines to determine if a codestream is strictly format-compliant. These guidelines complement the available reference implementations, which we use to verify if a given (encrypted) codestream is format-compliant.

2.2.4 *Part 8: JPSEC*

JPEG2000 Part 8 (ISO/IEC 15444-8, 2007; ITU-T T.807, 2006) has the title "Secure JPEG 2000" and is referred to as JPSEC. It "intends to provide tools and solutions in

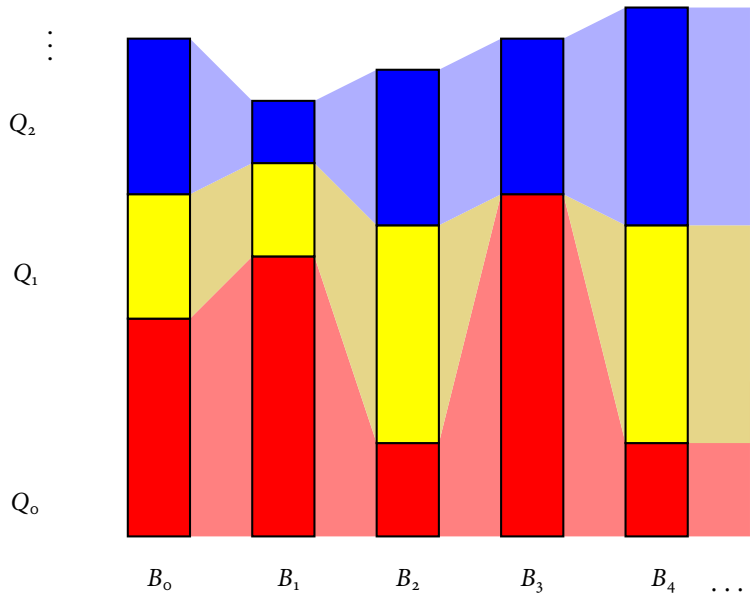


Figure 2.3: Codeblock contributions to quality layers in JPEG2000

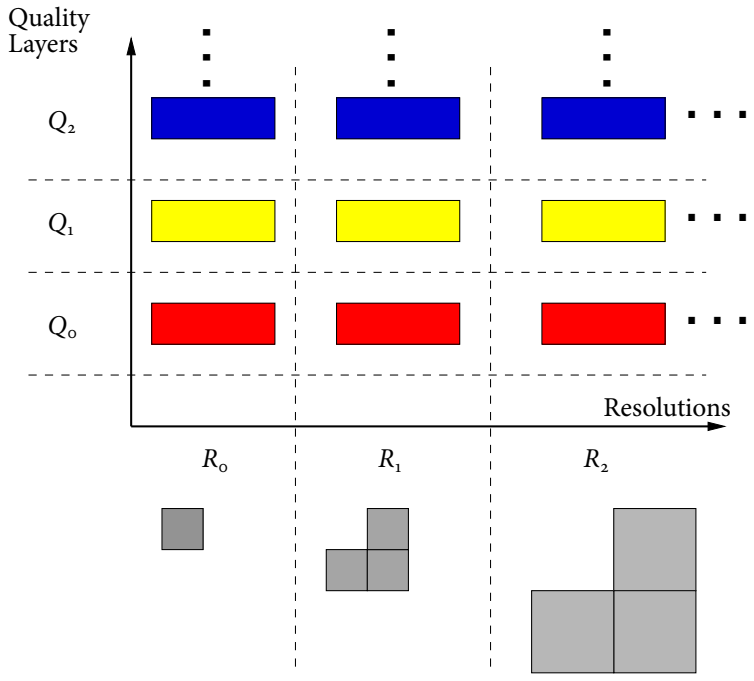


Figure 2.4: The notion of packets in JPEG2000

terms of specifications that allow applications to generate, consume, and exchange Secure JPEG 2000 codestreams" (p. vi). Specifically, the scope of this part of the standard is given as to define:

- a normative codestream syntax containing information for interpreting secure image data;
- a normative process for registering JPSEC tools with a registration authority delivering a unique identifier;
- informative examples of JPSEC tools in typical use cases;
- informative guidelines on how to implement security services and related metadata.

In this respect JPSEC deals with many of the topics of multimedia security that we mentioned in Chapter 1. JPSEC extends the codestream syntax to allow parts which are created by security services, e.g., encryption or watermarking. Furthermore, complementing the normative part, informative examples for algorithms are given.

We will take JPSEC into account in part I, when we turn to JPEG2000 packet header protection (which is missing from JPSEC). Further investigation of the interplay between encryption in JPSEC and different compression parameters can be found in Engel et al. (2007a).

2.3 IMPLEMENTATIONS

Several implementations of JPEG2000 exist, most are restricted to part 1.

JJ2000[2] and *JasPer*[3] are implementations in Java and C, respectively, and cover part 1 of the standard. While the license under which JJ2000 is released contains a number of restrictions, the JasPer license is very permissive.

Kakadu[4] is the implementation by David Taubman. It also covers many topics of part 2 and 3 of the JPEG2000 standard. Different licensing models are available, a binary distribution is provided for evaluation purposes.

The *verification models* provide implementations of the standard that reflect the progress at the time.

OpenJPEG[5] is another implementation in C, which covers part 1 and is released under BSD license.

[2] http://jj2000.epfl.ch/
[3] http://www.ece.uvic.ca/~mdadams/jasper/
[4] http://www.kakadusoftware.com/
[5] http://www.openjpeg.org/

There are also a number of purely commercial implementations.

Our own experiments are mainly conducted with JJ2000 and modifications of JJ2000. For example, we added support for parameterized wavelet lifting and arbitrary wavelet packet decomposition structures to JJ2000. The experiments on JPEG2000 header protection are also based on a version of JJ2000 which we extended with the necessary methods.

Part I

JPEG2000 PACKET HEADER PROTECTION

3 JPEG2000 PACKET HEADER PROTECTION

All proposals for format-compliant encryption schemes for JPEG2000 that have been made to date only encrypt packet body data, but leave packet header data in plaintext. In this chapter (cf. Engel et al., 2007a,b) we show that for providing strict confidentiality, leaving the packet headers in plaintext severely compromises the security of these schemes, as discriminative – and for some settings even visual – information can be extracted from the headers. We propose a set of format-compliant transformations of the packet header data that confines this information leakage.

3.1 INTRODUCTION

As already mentioned in Section 1.2.3, a significant number of approaches have been proposed which aim at achieving fully format-compliant encryption (sometimes in conjunction with a reduction in computational demands). All of these approaches, including the informative example of JPEG2000, Part 8 (ISO/IEC 15444-8, 2007) and to the best of our knowledge all related approaches that have been put forward to date, propose the exclusive encryption of the packet *body* data. The packet *header* data are invariably left in plaintext.

In Section 3.2 of this chapter, we will discuss to what extent leaving the packet header data in plaintext is problematic in terms of security for approaches that aim at providing full confidentiality. (As opposed to approaches that only offer a degradation in visual quality, these approaches aim at actually fulfilling the same function as full encryption and not disclosing any of the visual content, cf. Chapter 1.)

In Section 3.3, we propose a set of transformations that allows to protect the JPEG2000 packet header data in a fully format-compliant way, i.e., the resulting bitstream can be processed like a "normal" JPEG2000 bitstream, by any decoder compliant with JPEG2000, Part 1.

In Section 3.4, we first discuss general security considerations and then evaluate the proposed scheme with respect to the first of three scenarios: we look at how header protection can confine the leakage of information from the packet header for encryption schemes that aim at providing full confidentiality in a format-compliant way.

A second scenario can be found in approaches that aim at format-compliant encryption that provides sufficient degradation in quality, rather than full confidentiality (and trades off security for a decrease in computational demands). These approaches can be improved by a combination with the proposed header transformation. We discuss the possibilities and limitations of the proposed scheme with respect to this second scenario in Chapter 4. This chapter will also introduce the third scenario, namely, the utility of header protection as an encryption scheme on its own that can be used to provide transparent encryption.

3.2 INFORMATION IN THE JPEG2000 HEADER

The main JPEG2000 header contains general information on the bitstream, e.g., the coding style settings. The information is too general to infer from it anything substantial with regard to the specific visual content. The same is true for the tile and tile-part headers.

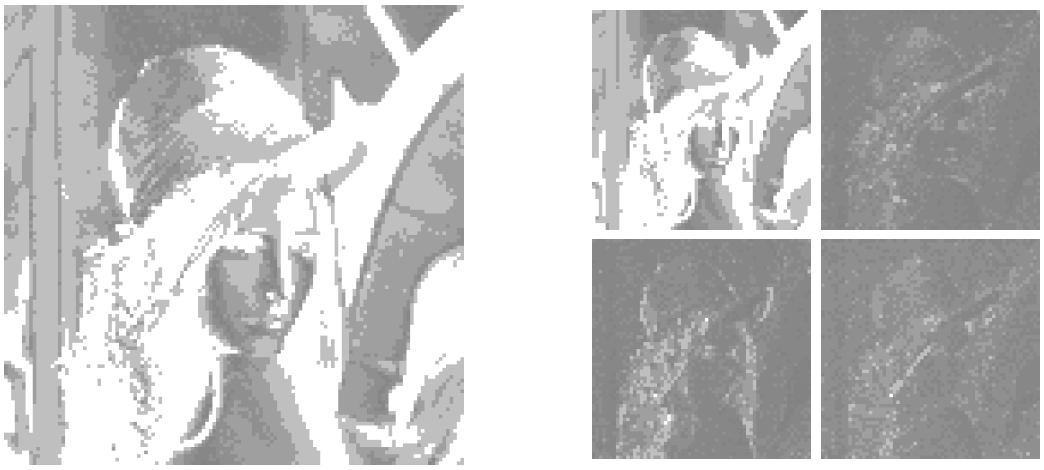
The situation is different for the packet headers. The packet headers contain information on four properties of the packet. We will refer to each different kind of header information as a "class" in the following. The four classes are:

- the length of the contribution of each codeblock to the packet (CCP),
- the number of leading zero-bitplanes (LZB),
- the inclusion information of each codeblock,
- and the number of coding passes that are contained in the packet for each codeblock.

From the sum of the CCP lengths, the length of the packet body can be derived.

The packet header information is specific to the visual content, and it is specific enough to be used as a fingerprint. Some suggestions have been made in this direction in the context of indexing, retrieval and classification. Tabesh et al. (2005) use the number of bytes spent on each subband for texture classification. Descampe et al. (2006) use a set of classifiers based on the packet header and packet body data to retrieve specified textures from JPEG2000 image databases. Liu and Mandal (2001) use the number of leading bitplanes as a fingerprint to retrieve specific images. Any class of header information alone is discriminative enough to be potentially used as a fingerprint (as will be illustrated in Section 3.4). The combination of all classes definitely is.

In the context of a security application, the fact that the information contained in the packet headers can be used as a fingerprint for a specific image is quite problematic,



(a) wlev=0 (b) wlev=1

Figure 3.1: LZB images, codeblock size 4 × 4

as it means that an attacker can link an encrypted image to its plaintext. The more distinctive the fingerprint, the more accurate is this assignment.

Such a fingerprint alone is problematic enough. The situation is even worse with respect to the leading zero bitplanes. While the other classes of header information do not reveal any *visual* information, the number of leading zero bitplanes can be used to get a rough estimate of the original image from which, especially for small codeblock sizes, the visual content is discernible. Figure 3.1 shows images that can be created from the information on the number of leading zero bitplanes for the Lena image coded with a small codeblock size of 4 × 4. The LZB images are created by interpreting the number of leading zero-bitplanes as pixel values (normalized over the range of grayscale values). Figure 3.1(a) shows the LZB image that can be created if no wavelet transform is employed. It can be seen that the LZB image is a crude quantization on a codeblock basis, from which the basic visual content is easily discernible. Coding settings that employ no wavelet transform are not very likely to be employed in practice, but also if the image is wavelet transformed, the edge information in the highpass subbands (which are not further processed regardless of the decomposition depth) that is revealed by the LZB-information is enough to spoil full confidentiality, as shown in Figure 3.1(b). For larger codeblock sizes less information is leaked through the number of leading zero bitplanes, but in order to provide full confidentiality, this data should generally not be accessible.

3.3 FORMAT-COMPLIANT HEADER PROTECTION

In the format-compliant encryption scheme we propose here, on the one hand, the aim is to destroy the information needed to create a strong fingerprint. On the other hand, the information needed to perform such tasks as rate adaption should be kept intact: the distinction between packet body and packet header and the distinction between individual packets has to be made available to the decoder. The resulting bitstream also needs to be fully format-compliant, i.e., the transformed header information has to be consistent with the available (encrypted) packet body data.

Furthermore, it should of course be possible to reconstruct the original packet headers from the encrypted packet header data by the use of a key. In the following subsections we propose a reversible, key-dependent transformation for each class of information in the packet header. Thereby the key is used to create a (pseudo-)random keystream. The choice of random generator can range from a physical source of randomness to using a symmetric cipher like AES in output feedback mode. In our tests we use the standard (linear congruential) random generator provided by Java 2 Standard Edition 5.0.

3.3.1 *CCP Lengths and Number of Coding Passes*

JPEG2000 explicitly signals both, the number of coding passes and the length of each codeblock contribution. (Depending on the block coding, in other codecs these numbers can be derived from each other, in JPEG2000 they are independent, see Taubman and Marcellin (2002)). In order to prevent access to this information, we distribute the CCP lengths and number of coding passes in each packet among all non-empty codeblocks. The number of coding passes gives only little information to a potential attacker; the lengths of the codeblock contributions are a more valuable source of information, as they are more distinctive. The algorithm we propose here can be applied to both (for clarity of reading we give the description for lengths only).

We need an approach that is key-dependent and reversible. Adding a random number modulo the total length of all CCPs to the offset of each CCP in the packet is not feasible: some offsets might change their relative positions during this procedure and the decoder would lack information about which random number is associated with which CCP. The situation leads to rather unique requirements for the transformation: given a vector of non-zero positive integers (the CCP-lengths or the number of coding passes) and a random keystream, we want an output vector that randomly redistributes these number among all positions (using all possible mappings), preserves the overall sum of the original vector, and has the same number of elements, each of which has to be a non-zero positive integer.

To achieve a transformation that adheres to these requirements, we change packet lengths (and number of coding passes, respectively) in overlapping pairs, starting with the first and the second length, moving on to the second and the third length, and so on. We redistribute the lengths between a pair of lengths by adding a random number from 0 to the total length of the two packets. This addition is performed modulo the packet size minus one and after the modulo operation we add one. Thus the size of any packet can never become zero. To avoid that an attacker can obtain the original sum of the pairs, we shuffle the lengths before and after redistributing them. We give the procedure in pseudo-code below. `v[]` is a vector of non-zero positive integers (indexing starts at 1). `random(0,1)` returns a random float number in $[0,1)$. `mod` is the modulo operation, which can return a negative residual (as is the case in many programming languages).

```
shuffle (v)
borders := size(v)-1
for i := 1 to borders
  sum := v[i] + v[i+1]
  r := ⌊random(0,1)⌋*sum
  newBorder := ((v[i]+r) mod (sum-1)) + 1
  v[i] := newBorder
  v[i+1] := sum - newBorder
end for
shuffle(v)
```

The transformation can be reversed easily by unshuffling the input, traversing it from end to start, using the random numbers in reverse order, setting `newBorder` as:

```
newBorder := (v[1]-r-1) mod (sum-1)
if (newBorder <= 0) then
  newBorder := newBorder + (sum-1)
end if
```

and finally unshuffling the result again.

This approach allows to completely redistribute lengths and coding passes among the codeblocks in a packet. The number of possible alterations depends on the number of codeblocks, and how they make their contributions to the individual packets. If a small number of packets contains many contributions, more alterations are possible than if a large number of packets only contain a small number of contributions each. Two non-empty codeblock contributions in a packet are enough to hide their lengths and number of coding passes. Therefore, in practical scenarios there will always be ample opportunity to sufficiently randomize CCP lengths and numbers of coding passes.

3.3.2 *Leading Zero Bitplanes*

The number of leading zero bitplanes (LZB) for each codeblock is coded by using tag trees (Taubman and Marcellin, 2002). As discussed above, this information is even more critical than the other classes of header information, as by using the number of LZB an attacker can obtain information on the visual content of the encrypted image (for small codeblock sizes).

For the transformation of the number of leading zero bitplanes, two options are investigated. One option is to adapt the method proposed above for CCP lengths and number of coding passes (and allow zeros). For the other option, we simply use the keystream to generate random bytes. We then add a random byte to the number of leading zero bitplanes modulo a previously determined maximum number of skipped bitplanes. For decoding, the random byte is subtracted instead of added.

The maximum number of skipped bitplanes needs to be signalled to the decoder, e.g., by inserting it into the key or by prior arrangement. Note that the maximum number of skipped bitplanes needs to be greater or equal to the original maximum number of skipped bitplanes (otherwise the modulo operation cannot be reversed). Theoretically the new number could be arbitrarily high (we found no restrictions in that respect in the standard), but most implementations will have a maximum number of bitplanes for the representation of coefficient data that must not be exceeded.

When the number of leading zero bitplanes is changed, the length of the output from the associated tag tree might change as well. This change in length has to be reflected in the tile header, otherwise the decoder will complain. Alternatively, if only a single tile is used, the length in the tile header can be set to be unspecified. Furthermore, the maximum number of bitplanes needed to represent the coefficients in each subband can be derived from information contained in the main header: "The maximum number of bit-planes available for the representation of coefficients in any subband, b, is given by M_b as defined in Equation E.2" (ISO/IEC 15444-1 (2000), p. 70). Equation E.2 in ISO/IEC 15444-1 (2000) basically derives the number M_b from information contained in the `QCD` and `QCC` marker segments in the main header. Therefore, to achieve full format-compliance, the main header needs to be changed accordingly. Otherwise decoding will still work, but the decoder might issue a warning. Note, however, that neither of the reference implementations JJ2000 and JasPer, which we used in our tests, issued a warning.

The total number of possible changes for all packets depends on the number of available codeblocks. If more codeblocks exist, more information can be randomized.

3.3.3 *Inclusion Information*

Each packet contains the inclusion information for a certain quality layer for all codeblocks in the precinct associated with the packet. For the sake of simplicity, we assume that no precinct partitioning is used. In this case each packet contains inclusion information for all codeblocks of all subbands in the resolution associated with the packet. There are four types of inclusion that codeblock c can have in packet p:

FI – c is included in p for the first time, i.e., c has not been included in any previous packet,

NI – c is not included in p and has never been included in any previous packet,

PI – c has been included in a previous packet and is also included in p, and

PN – c has been included in a previous packet but is not included in p.

The sequence of inclusion information of each codeblock is coded depending on the type of inclusion. *FI* and *NI* are coded by an inclusion tag tree. For *PI* and *PN*, i.e., the codeblock has been included in a preceding packet, a 1 is coded in the header if the codeblock is included again in the current packet, and a 0 is coded if the codeblock is not included in the current packet.

The goal of the proposed approach is to permute inclusion information for each packet in such a way that the original inclusion information cannot be derived without the key and that the resulting "faked" inclusion information complies with the semantics of JPEG2000. We limit the approach to permutation of the original inclusion information and do not split available packet body data from one codeblock to more codeblocks, and we also do not merge contributions from distinct codeblocks in a single codeblock. Also, the permutation is applied per packet, i.e., we do not merge the packet body data from different packets, as this would have a detrimental effect on the scalability properties of the resulting bitstream. These restrictions do not interfere with the aim to prevent the creation of a strong fingerprint from the sequence of inclusions (although they would help to hide the number of inclusions of each type) and have the advantage of facilitating straightforward reversibility.

We distinguish two kinds of packets: an *empty packet* is a packet for which a header is written, but which does not contain any codeblock contributions, i.e., all codeblocks are included as *NI* or *PN*. In a *non-empty packet* there is at least one contribution from one of the codeblocks, i.e., at least one codeblock is included as *FI* or *PI*.

We define the *inclusion vector* v_p for a packet p as the sequence of the inclusion information for each codeblock associated with p (i.e., each codeblock in the subbands of the resolution associated with p):

$$v = [I_{c_i}], \forall c_i \in p, I_{c_i} \in \{FI, NI, PI, PN\}.$$

The order of elements in the inclusion vector follows the order in which the codeblocks are scanned during image coding. In JPEG2000 this is an ordering by subband in the sequence HL, LH, HH followed by a lexicographical ordering over the 2-D coordinates of all codeblocks in the subband. Obviously, the order in which subsequent items of the same inclusion type appear are irrelevant, and count as the same permutation. We give the number of possible distinct permutations further below.

An arbitrary permutation of the inclusion vector of each packet would not produce a format-compliant bitstream. Consider the following example: After the permutation of the inclusion vectors for two packets p_l and p_{l+1} let codeblock c be included in packet p_l as *FI*. An arbitrary permutation could assign inclusion type *NI* to c in p_{l+1}. This would lead to a contradiction because c can never be *NI* after its first inclusion.

Only the first permutation in a resolution r may be an unrestricted permutation (but permutations do not necessarily have to start at the first packet of the first layer). After the first permutation, the permutations for the subsequent packets have to regard the inclusion information that has been signaled in the directly preceding packet. The following transitions are possible:

p_l	*FI*	*NI*	*PI*	*PN*
p_{l-1}	*NI*	*NI*	*FI*, *PI*, *PN*	*FI*, *PI*, *PN*

A codeblock can only be included as *FI* or *NI* in p_l if it has been included as *NI* in packet p_{l-1}. *PI* and *PN* can follow a previous inclusion of *FI*, *PI*, or *PN*. It follows that permutations can only be performed for non-empty packets, because for empty packets the positions of the inclusions of types *NI* and *PN* are fully determined by the previous packet. Also note that the number of inclusions of type *FI* plus the number of inclusion of type *NI* in p_l is always equal to the number of inclusions of type *NI* in p_{l-1}. The same is true for inclusions of type *PI* and *PN*: their number in p_l is equal to the number of inclusions of type *FI*, *PI*, and *PN* in p_{l-1}.

We perform the permutations in compliance with these transition rules to produce a – syntactically as well as semantically – consistent sequence of inclusion information over the packets of each resolution. First, in each resolution, the first packet suitable for permutation is searched: This packet has to have at least one non-empty codeblock contribution (*FI* or *PI*) and one empty inclusion (*NI* or *PN*). Packets for which the codeblocks all have the same inclusion information are obviously not suitable for permutation, and packets with mixed *FI* and *PI* only occur after the first candidate packet. The inclusion information in the first candidate packet is permuted.

For the subsequent non-empty packets, the inclusion information of the immediately preceding packets is regarded in the permutation. The inclusion vector v for a packet p_l is split into two vectors: $v^{(1)}$ contains all *FI* and *NI* inclusions, and $v^{(2)}$ contains all inclusions of type *PI* and *PN*. Both vectors are permuted randomly (using the

key-stream) to form $\hat{v}^{(1)}$ and $\hat{v}^{(2)}$. According to the possible transitions given above, the elements of $\hat{v}^{(1)}$ are assigned (in the randomized order) to the positions that are marked as *NI* in the packet of the previous layer, p_{l-1}. The elements of $\hat{v}^{(2)}$ are assigned to the remaining positions (again in the randomized order). The inclusions in p_l that mark a non-empty codeblock contribution, i.e., *FI* and *PI*, are assigned the length and number of new coding passes of the non-empty codeblock contributions of the correct inclusion vector v in the order in which these contributions appear in v (these CCP lengths and number of coding passes may be subject to transformation later, or maybe have already been transformed). All first inclusions (*FI*) in $\hat{v}^{(1)}$ are assigned the number of leading zero-bitplanes in the order of *FI*-inclusions in v. After all packets have been processed, the new header information is written. If the key that has been used for the permutation is known, this procedure is reversible. Note that the permutation in this approach crosses subband boundaries: the inclusion information is reassigned over all codeblocks in the packet.

For empty packets, no permutation is possible. For these packets, the inclusion information only needs to be updated based on the inclusion information in the previous packet, according to the possible transitions.

To illustrate the process of format-compliant permutation we give an example. Let $v_{p_l} = [FI, NI, NI]$ be the inclusion vector of the first candidate package p_l in a resolution with three codeblocks c_0, c_1, c_2. After permutation the new inclusion information is $\hat{v}_{p_l} = [NI, NI, FI]$. The length of the codeblock contribution and number of leading zero-bitplanes is transferred from c_0 to c_2. In the next packet p_{l+1} let the real inclusion information be given by $v_{p_{l+1}} = [PI, FI, NI]$. This vector is split into $v^{(1)}_{p_{l+1}} = [FI, NI]$ and $v^{(2)}_{p_{l+1}} = [PI]$. Considering the "faked" inclusion information $\hat{v}_{p_l}$ of the previous packet, the positions of codeblocks c_0 and c_1 are the candidate positions for inclusions of type *FI* and *NI*. $v^{(1)}_{p_{l+1}}$ is permuted to form $\hat{v}^{(1)}_{p_{l+1}} = [NI, FI]$ and the new inclusion information is assigned to the respective positions of c_0 and c_1. The length of the contribution and the number of coding passes and leading zero-bitplanes are updated for the non-empty contribution. In this example, $v^{(2)}_{p_{l+1}}$ only has one element which has to be assigned to the position of codeblock c_2 to form a consistent sequence of inclusion information. The length of the contribution and the number of coding passes is updated for this codeblock. The new inclusion information for p_{l+1} is $[NI, FI, PI]$.

We now turn to the investigation of the number of possible permutations. For the lower resolutions, there will typically be only few codeblocks that contribute to each packet, so the number of permutations will be very limited. As the permutation of packet headers is only used in conjunction with the encryption of packet bodies, this is not a problem. The fingerprint that could be derived from the inclusion information in the lower resolutions is not very distinctive, the main point is to destroy fingerprints in the higher resolutions.

We can give the number of possible permutations for a packet p_l which contains the codeblock contributions for layer l (for a specific resolution). Let $|C_{p_l}|$ be the total number of codeblocks that are relevant for p_l, and $|FI_{p_l}|$, $|NI_{p_l}|$, $|PI_{p_l}|$, $|PN_{p_l}|$ the number of codeblocks in p_l with inclusion type *FI*, *NI*, *PI* and *PN*, respectively. If p is the first packet to be permuted, we have no restrictions in the possible permutations. Furthermore, in this case the inclusion information in p will consist of either only *FI* and *NI* or of *PN* and *PI* (because otherwise a previous packet would have been the first to be permuted).

Without loss of generality, we assign the positions for inclusions of type *FI* and *PI*. The number of possible permutations is then given as:

$$\binom{|C_{p_l}|}{|FI_{p_l}|}\binom{|C_{p_l}|}{|PI_{p_l}|}. \tag{3.1}$$

If p is a packet that pertains to a higher layer than the first candidate, the inclusion information of the preceding packet p_{l-1} has to be taken into account. Inclusions of type *FI* and *NI* in p can go into positions that are included as *NI* in p_{l-1}. The rest of the positions can be assigned to inclusions of type *PI* and *PN* in p_l. The number of possible permutations is determined by

$$\binom{|NI_{p_{l-1}}|}{|FI_{p_l}|}\binom{|FI_{p_{l-1}}| + |PI_{p_{l-1}}| + |PN_{p_{l-1}}|}{|PI_{p_l}|}. \tag{3.2}$$

The actual number of permutations for a given input image depends on a variety of factors. The used compression parameters influence the number of codeblocks that are available for permutation in the first place. With small codeblock sizes the number of available codeblocks increases. If the number of quality layers is increased, then there are also more packets and therefore more permutations can be performed. The bitrate with which the image is encoded also influences the number of packets that are included in the codestream. Finally, the number of permutations that can be applied to the packets of a resolution is influenced by the point at which the first candidate packet is found, and by how diverse the inclusion information is in this packet and the following packets. If all the codeblocks of a resolution always have the same inclusion information in each packet, then no permutation of inclusion information is possible. Luckily, this case is extremely unlikely in practice. Table 3.1 shows the number of possible permutations (perm.) for the Lena image (512 × 512 pixels) with different compression settings (codeblock size, rate, number of quality layers).

3.3.4 *Combined Format-Compliant Header Transformation*

The format-compliant transformation of the different pieces of information in the packet headers can and should be combined. The format compliance of the combined

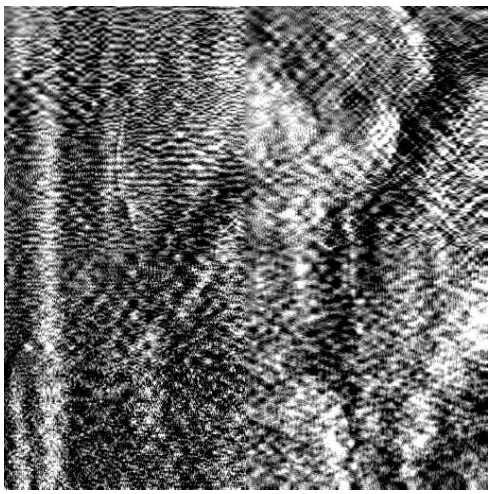
Cblk. size	rate (bpp)	layers	perm.
64 × 64	3	32	10^{217}
32 × 32	3	32	10^{938}
64 × 64	3	12	10^{56}
32 × 32	3	12	10^{257}
64 × 64	0.25	32	10^{42}
32 × 32	0.25	32	10^{146}

Table 3.1: Number of possible format-compliant permutations of the inclusion information for Lena (*wlev*=5)

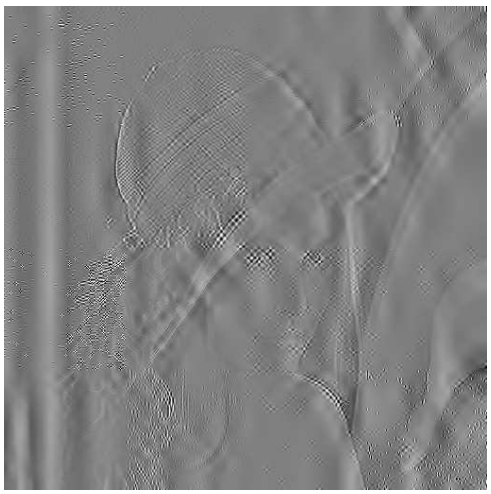
format-compliant header encryption has been verified experimentally by decoding the encrypted files with the reference implementations JasPer and JJ2000. The order in which they are applied is arbitrary, only decoding has to apply the reverse transformations in the reverse order.

3.3.5 *Visual Examples*

In order to give an illustration of the extent of information contained in the header we give some visual examples of reconstructed plaintext images for which only the packet headers have been encrypted and the packet bodies have been left unencrypted (this setup can be used in a lightweight encryption scheme, we will discuss this point in more detail in Chapter 4). As the number of coding passes has only very little impact on the visual quality, we have combined their transformation with the transformation of the CCP lengths. For the transformation of the number of leading zero bitplanes, we used the (slightly adapted) algorithm for CCP and NCP. Examples for the second option, adding random bytes from the key-stream, are given in the following chapter. Figure 3.2 shows the reconstruction for large codeblocks and few quality layers, Figure 3.3 for medium-sized codeblocks and more quality layers and Figure 3.4 for very small codeblocks. It can be observed that the impact of the transformations is increased with the number of codeblocks, especially for inclusion information and leading zero bitplanes (at least visually, PSNR, and in this case also ESS, fail to express the difference in quality).

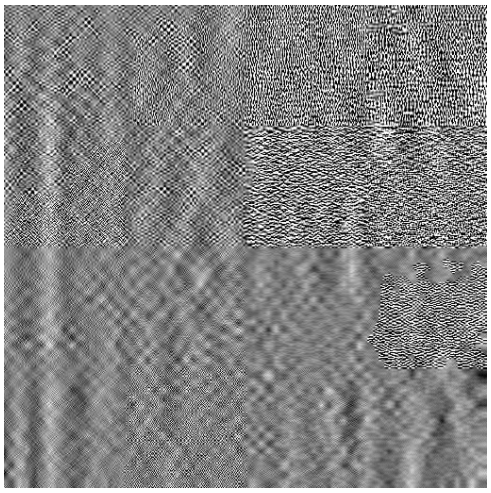


(a) CCP lengths (& number of coding passes), PSNR 11.9db, ESS 0.26



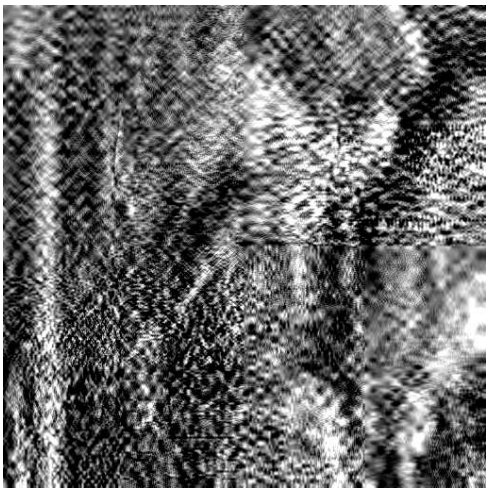
(b) Number of lzb, PSNR 12.8db, ESS 0.23

(c) Inclusion information, PSNR 16.95db, ESS 0.36

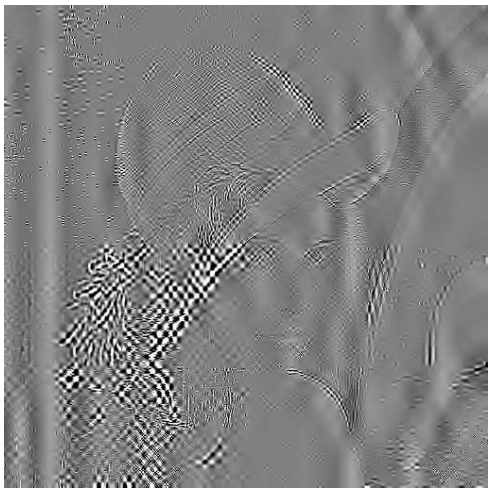


(d) All transformations, PSNR 9.2db, ESS 0.24

Figure 3.2: Visual examples of reconstructions with transformed packet headers for Lena, rate 1 bpp, cblk. size 64 × 64, 16 layers

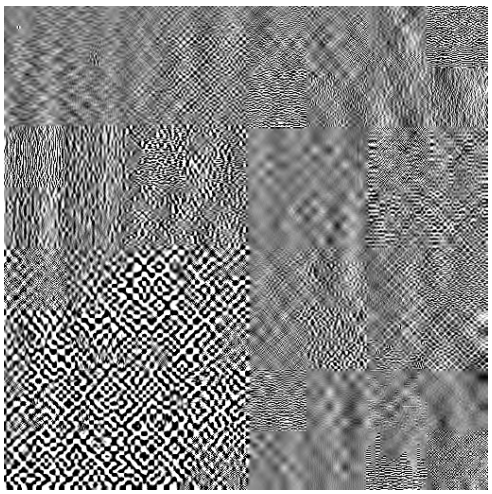


(a) CCP lengths (& number of coding passes), PSNR 12.8db, ESS 0.26



(b) Number of lzb, PSNR 12.1db, ESS 0.32

(c) Inclusion information, PSNR 16.0db, ESS 0.31



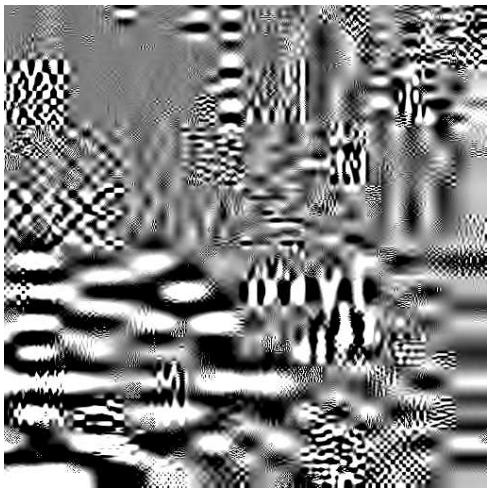
(d) All transformations, PSNR 8.5db, ESS 0.27

Figure 3.3: Visual examples of reconstructions with transformed packet headers for Lena, rate 1 bpp, cblk. size 32 × 32, 32 layers

(a) CCP Lengths (& number of coding passes), PSNR 10.4db, ESS 0.27

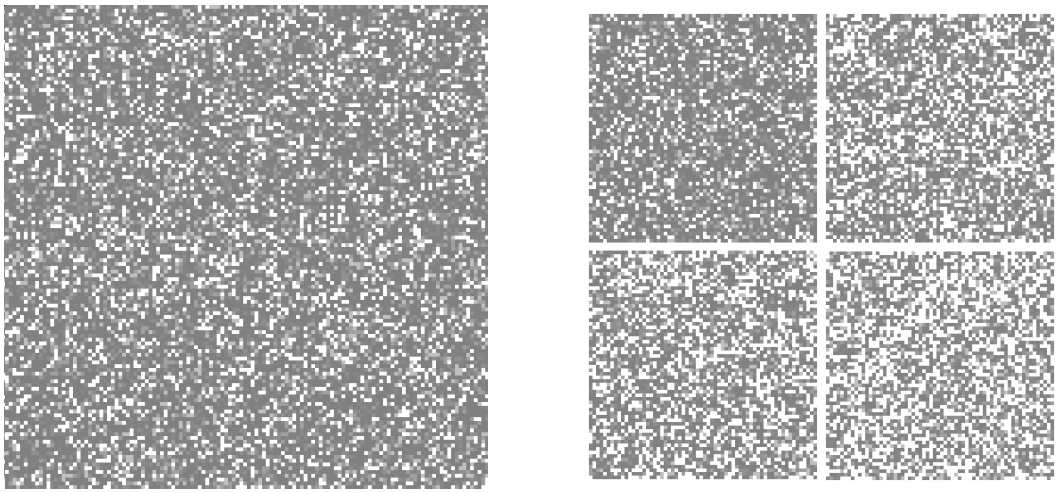
(b) Number of lzb (shuffled and redistributed), PSNR 9.2db, ESS 0.27

(c) Inclusion information, PSNR 15.4db, ESS 0.26



(d) All transformations, PSNR 8.2db, ESS 0.23

Figure 3.4: Visual examples of reconstructions with transformed packet headers for Lena, rate 1 bpp, cblk. size 8×8, 32 layers



(a) Number of LZB transformed, *wlev*=0 (b) Number of LZB transformed, *wlev*=1

Figure 3.5: Visualization of attack after LZB transformations (for wlev=0 and codeblock size 4×4)

3.4 EVALUATION

Generally it should be noted that the proposed key-dependent transformations use permutations, which are principally vulnerable to known-plaintext attacks. Therefore, the key for the transformation of header information should under no circumstances be derived from the key that is used for the encryption of the packet bodies.

In the following, we evaluate the proposed scheme in the first of the three – very different – scenarios. Here, we look at format-compliant packet body encryption schemes that aim at providing full confidentiality. In this scenario we deal with a situation in which all packet body data are encrypted. The motivation to use header protection in this scenario is to avoid the creation of a fingerprint.

The second scenario is format-compliant partial / selective encryption that aims at a reduction in computational complexity. The third scenario is JPEG2000 header protection as a lightweight transparent encryption scheme on it own. Both of these scenarios are discussed in the next chapter.

The visual information contained in the packet header as the number of leading zero bitplanes is a severe security problem for applications that require full confidentiality, as this information is preserved in the encrypted bitstream after packet body encryption. The proposed header protection scheme solves this security problem: the LZB information is completely destroyed by the transformation. Figure 3.5 shows the

same LZB-images for Lena as Figure 3.1, but with the headers transformed. As can be seen, no visual content is discernible anymore.

As discussed in Section 3.2, the information contained in the header information can also be used as a distinctive fingerprint. This fingerprint remains even if all of the packet body data is encrypted. Figure 3.6 illustrates this point. We have tested 175 images, all taken with the same camera model and cropped to 512 × 512 pixels at 8 bpp. All of the images are encoded with JPEG2000 at a bitrate of 0.25 bpp with 32 quality layers and a codeblock size of 32 × 32. The header information in all the packets of each image is recorded. We then compare the header information of a single image from the set to the header information of each other image in the set. The ratio of the number of items in the header information that have the same value (at the same position) to the total number of items is recorded. In the plots, the ordinate shows this value for each class of header information and each image. Note that for CCP lengths and number of coding passes we ignore positions in the header for which the information is 0 for both the reference and comparison image. Figure 3.6 shows the similarity in header information of one image (# 23) with the other images. It can be seen that the similarity measures to other images are confined within a certain range of variance. It is not surprising to see that the similarity of the CCP-lengths is very small for differing images. Interestingly, the number of corresponding items in the inclusion information is very large. This is due to the fact that for the inclusion information we also counted all the inclusions of type *NI* which at this bitrate occur a lot. The variance of the similarity of the inclusion information is confined relatively strictly and therefore also the inclusion information may serve as a discriminating feature.

Obviously the situation will be different if the reference image is re-encoded with different compression parameter settings. But as this illustration shows, any class of header information can be used to link a known JPEG2000 plaintext to a packet-body encrypted ciphertext. For many applications that require full confidentiality such a leak of information constitutes a major compromise of security.

The proposed transformations can be used to prevent the creation of a fingerprint which uses the details of the packet headers, such as proposed by Liu and Mandal (2001), Descampe et al. (2006) and Tabesh et al. (2005). Figure 3.7 shows the same comparison as Figure 3.6, but this time with the header information of the reference image transformed. It can be observed that the proposed transformation methods obstruct the identification of the image in the set of 175 images: the transformed header bears no similarity to the original header. Only for the codeblock lengths a minute trace remains. This is due to packets of the lowest two resolutions which only contain a single codeblock each. For the used compression settings no transformation was possible for these packets.

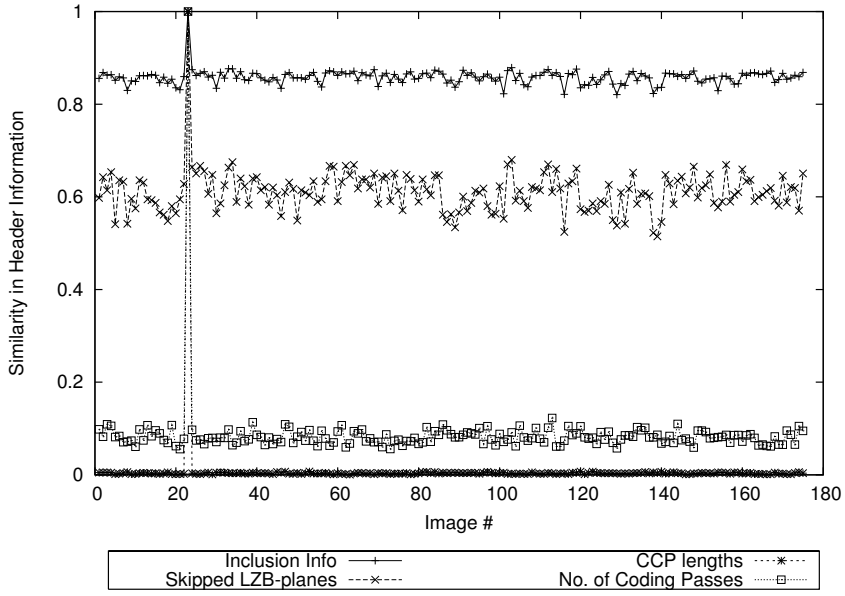


Figure 3.6: Comparison of similarity in header information for 175 images

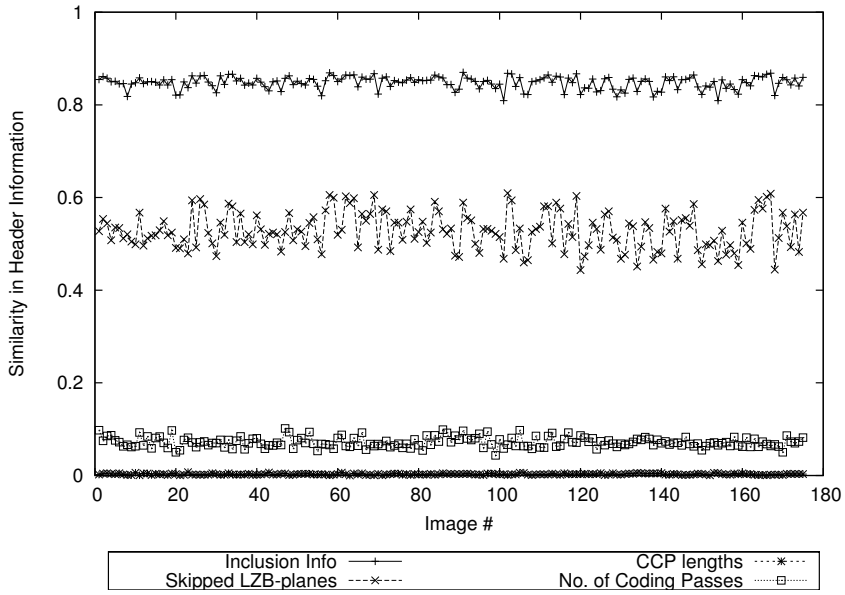


Figure 3.7: Comparison of similarity in header information for transformed header information

Note that as the proposed scheme preserves packet boundaries and does not merge or split the data in the packets, some information does of course remain. A fingerprint that uses the size of the individual packets can still be created. Furthermore, the number of inclusions of each inclusion type stays the same as in the plaintext image. This information could be used to obtain a fingerprint, albeit a much weaker one than if the order of inclusions was known. If packet boundaries were crossed and furthermore inclusion information was split up among codeblocks, a possible fingerprint would be further weakened. However, the downside would be a loss in semantics for the encrypted version (which would make rescaling more unreliable, for example). The highly discriminative fingerprints based on the detailed packet header information are successfully prevented.

3.5 CONCLUSION

In this chapter, we have proposed a fully format-compliant protection scheme for JPEG2000 packet headers. The proposed scheme can be used to confine the information leakage that is present in all encryption scheme that selectively encrypt packet body data. The visual information contained in the JPEG2000 packet header is destroyed. Furthermore, the strong fingerprint based on the details of the packet headers is prevented by the scheme. As the scheme preserves packet boundaries, fingerprints based on more general information like overall packet lengths are not destroyed. However, if format-compliance is desired in a sense that allows to perform tasks like rate adaption in the encrypted domain, the information needed for these tasks will always need to be preserved to some extent. Therefore, while the proposed scheme significantly improves the security of format-compliant encryption schemes that rely on the encryption of packet body data, the combined schemes can never be as secure as full encryption (but it can improve the content security). The straightforward approach to encrypt the *full* bitstream with a cryptographically strong cipher is and remains the *only* option that is secure in the strong cryptographic sense.

4 APPLICATIONS OF JPEG2000 HEADER PROTECTION

In the last chapter we introduced a method to protect the JPEG2000 packet headers and showed how it can be used to restore security of format-compliant packet-body-based encryption schemes. In this chapter (cf. Engel et al., 2007b,c) we will investigate the utility of the proposed header protection scheme with respect to two other application scenarios.

First, we look at if and how header protection can improve partial / selective packet body encryption, where full confidentiality is not the goal, but rather a reduction in computational complexity. As portions of the data remain in plaintext, partial / selective encryption schemes are not suited for full confidentiality (even if the data cannot be used to create a visual reconstruction, they can always be used as a fingerprint). The aim here is to introduce sufficient reduction of the visual quality to prevent a pleasant viewing experience (i.e., sufficient encryption). Traditional approaches encrypt packet bodies in a format-compliant way.

Second, we will investigate to what extent header protection is useful on its own in terms for providing transparent encryption. The resulting encryption scheme is lightweight and necessarily transparent, but comes at very low cost.

4.1 IMPROVEMENT OF PARTIAL / SELECTIVE PACKET BODY ENCRYPTION

As has been detailed before, the aim of partial / selective encryption is not to provide full confidentiality. For schemes that do not strive for providing full confidentiality information leakage in the header is not problematic, as long as the information remains reasonably secure. Therefore, in this case we employ header protection not to restore full security, but to counteract known attacks and possibly lower computational demands.

Before we go into detail, we need to look at the computational demands of the header transformations compared to packet body encryption. As the array-based permutations are all of low complexity, the header transformations are computationally cheap. The necessary operations in each substep are the generation of a random number and exchanging the places of two items in the array. The cipher used for packet body encryption will usually be computationally more demanding. Furthermore, for

Cblk. Size	Layers	Packet Header	Packet Body
64 × 64	16	0.84%	99.02%
64 × 64	32	1.15%	98.71%
32 × 32	32	2.97%	96.98 %
16 × 16	32	7.94%	91.94%

Table 4.1: Distribution of data between packet header and packet body in percent of the total bitstream

practical compression settings the ratio of packet header data to packet body data is very small, as illustrated by Table 4.1 for the Lena image, which shows the percentage of packet body data and packet header data, respectively, to the size of the total bitstream (compression at full rate, nearly lossless).

Typical partial / selective encryption schemes for JPEG2000, like the one proposed by Norcen and Uhl (2003), encrypt a certain percentage of the packet body data, starting from the beginning of the bitstream. This can be done either in layer or resolution progression mode (for a comparison see Norcen and Uhl (2003)). Even if small portions of the bitstream are encrypted, high distortion in visual quality can be achieved.

A straightforward attack is discussed by Norcen and Uhl (2003): The concealment attack discards all encrypted parts of the bitstream. The authors simulate the attack by using JPEG2000 error concealment. During encoding, an error resilience segmentation symbol is inserted at the end of each bitplane. If the decoder cannot decode the segmentation symbol, an error has been detected. We have adapted the error concealment to handle encrypted bitstreams: If the segmentation symbol cannot be decoded correctly, then the affected codeblock contribution is discarded as a whole ("zero concealment"). The attack lowers the level of reduction in visual quality for the encryption scheme. This is illustrated by Figure 4.1, for which 1% of the packet body data has been encrypted: (a) shows the direct reconstruction without error concealment. It can be seen that there is a high degree of visual distortion. However, as (b) shows, the zero concealment achieves a reconstruction that is of a higher visual quality.

The concealment attack can be completely prevented if, apart from a part of the packet body data, all packet headers are encrypted. The header transformation of a packet causes the error concealment procedure to regard the packet body data as erroneous and discard it. Figure 4.2 shows the comparison between packet body only encryption and packet body encryption combined with full packet header encryption (for Lena at a compression rate of 0.25 bpp). As the peak-signal-to-noise-ratio (PSNR) is not suitable as a measure for images of such low quality (see Section 1.3), we have used the edge similarity score (ESS) proposed by Mao and Wu (2004).

(a) direct reconstruction (b) zero concealment

Figure 4.1: 1% of body data encrypted

It can be seen that for low encryption rates the partial packet body approach preserves a lot of edge information. The visual examples for different percentages of encrypted packet bodies that are shown in Figures 4.3(a) to 4.3(d) support the ESS scores. If, however, the packet body approach is combined with full packet header protection, then the edge information disappears from the reconstruction even for very small packet body encryption rates, and a high minimum level of distortion can be guaranteed. Some visual examples are shown in Figure 4.4. It can be seen that compared to the packet body only approach, the additional packet header encryption prevents both, direct reconstruction and zero concealment attacks.

Of course, this comes at the cost of encrypting all headers. But, as we discussed above, for practical coding settings, the size of the packet header data will be only a small fraction of the packet body data. So, rather than encrypting a larger portion of the packet body data, it is advisable for most compression settings to encrypt all packet headers, as for this approach the costs are low and the distortion is high. As an example we take the 512×512 pixel Lena image at 0.25 bpp, 32 quality layers and a code-block size of 64×64 pixels. In this case, the packet header data is 3.5% and the body data is 94.7% of the total bitstream (the rest is main header data). As Figure 4.2 illustrates, the distortion introduced by 1% packet body encryption and full packet header encryption is comparable to the distortion introduced by approximately 45% packet body encryption. Encrypting 45% of the packet body data in these settings amounts to encrypting 42% of the total bitstream. The combined approach of encrypting 1% of

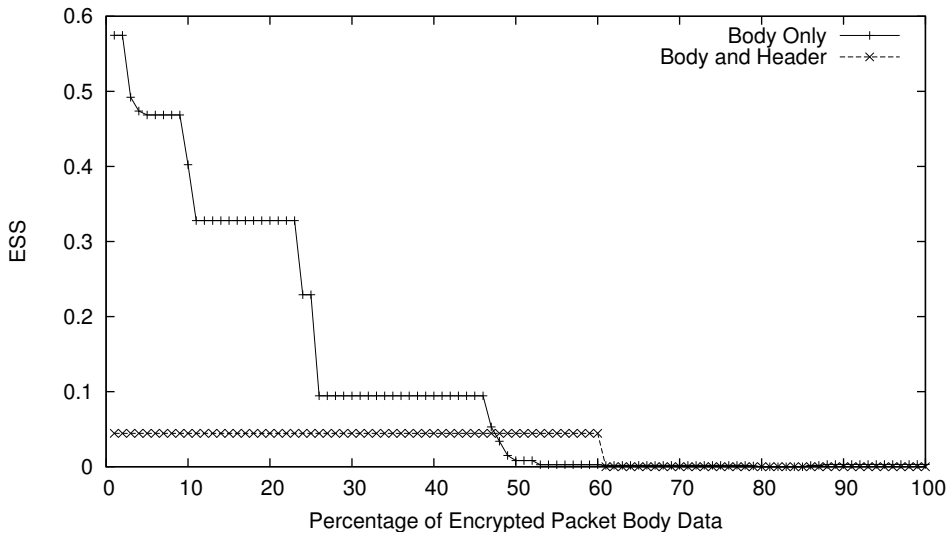


Figure 4.2: ESS comparison of concealment attack with encrypted header and without encrypted header

the body data and transforming all header data amounts to dealing with data that corresponds to 4.4% of the total bitstream (the corresponding reconstructions are shown in Figure 4.4).

In combination with the encryption of a small portion of the packet body data, the packet header protection scheme can be used in applications that do not strive for full confidentiality. The overall number of possible transformations is too high for a successful brute-force attack. Only the packets in the lowest resolutions have few enough codeblocks to make trying all possible headers an option. So while an attacker could gain an idea of the visual content by attacking the packets of the lower resolutions for which only the packet headers but not the packet bodies have been protected, the full visual quality version of the original content remains protected. This is also true in the presence of a preview image that could have been obtained from side information. As Said (2005) points out, many partial encryption schemes are vulnerable to low complexity attacks based on the availability of such side information. In the context of the proposed scheme, a low quality preview image cannot be used to reconstruct the JPEG2000 packet header information. We have also found no way to reconstruct the packet header from the unencrypted packet body data.

The question arises whether, as the header transformation guarantees a high level of reduction in visual quality, it would be a good approach to only use packet header protection for the whole bitstream and leave the packet bodies in plaintext. Because the

lower resolutions in the bitstream only have a small number of possible transformations, for practical coding settings, the lowest resolution will often only be constituted of a single codeblock. In this case, the packets of this resolution will invariably only have a single entry of inclusion information, CCP length, LZB information and number of truncation points. Therefore, a basic visual quality will always remain if only header encryption is employed. For high visual distortion it is therefore inevitable to use packet body based encryption at the beginning of the bitstream. However header encryption can clearly be used to provide transparent encryption, a case which we discuss in the following section.

4.2 TRANSPARENT JPEG2000 ENCRYPTION WITH PACKET HEADER PROTECTION

In this section we investigate the utility of header encryption if it is not used in combination with another (packet-body-based) method, but on its own for providing transparent encryption. In this case, for encryption the transformations of the packet headers are applied to the JPEG2000-coded source image, but this time the packet body data is left in plaintext. Without the header data the packet body data cannot be interpreted correctly. Especially the higher resolutions offer a wide range of different permutations preventing the decoding of the full quality image (in the sense of making decoding computationally too expensive; the proposed encryption method is definitely lightweight). For the lower resolutions the header data is left in plaintext to provide a preview image. Control over the extent of transparency can be exerted by choosing at which point in the JPEG2000-codestream to start with the header transformations.

Employing the set of header transformations for transparent encryption promises some advantages. Apart from low computational complexity and a fully format-compliant ciphertext, the proposed method has several beneficial properties, e.g., it allows to perform tasks such as rate-adaptation and visual hashing in the encrypted domain. We evaluate the security of the proposed approach and discuss possible attacks. The proposed approach succeeds in protecting the full-quality version of the visual content reasonably well, especially considering its low computational demands.

We have already given some visual examples of the impact of header protection on the (subjective) visual quality of the image. In Figures 4.5 and 4.6 we give some more example to illustrate the effectiveness of the scheme for transparent encryption. The test image[1] is shown in Figure 4.5(a). It is an 8 bpp grayscale image of 1024×1024 pixels. For the illustration we used a bitrate of 1 bpp, a codeblock size of 32×32 and 32 quality layers. For the transformation of the number of leading zero bitplanes, we used the

[1] Courtesy of Jumeirah, http://www.jumeirah.com/.

option of adding random bytes from the key-stream (other than in the last chapter, where we used the shuffling algorithm).

Figures 4.5(b) to 4.5(d) show direct reconstructions of the image with a single piece of header information encrypted (with header transformation for the encrypted image starting at the first resolution). The impact of transforming the different parts of the header information can be observed. Figures 4.6(a) to 4.6(c) show the direct reconstruction for all header transformations combined, with the start of the transformation set to a different resolution level for each figure. It can be seen that if the transformations start at a higher resolution level, lesser distortion is introduced. It should be noted that the direct reconstructions do not represent the preview images, but rather the result of reconstructing *all* of the available (transformed) data (in a way, it is the attacker's view). In order to get a preview image, the portion of the bitstream that has not been distorted needs to be decoded. An actual preview image is shown in Figure 4.6(d). This preview image was taken from a bitstream where the transformations start at resolution 3, so the preview image is equal to resolution 2 of the original image (i.e., the third resolution contained in the bitstream).

4.2.1 *Efficiency*

The transformations used in the proposed scheme are computationally cheap. The array-based permutations used in the transformations are all of linear complexity. The necessary operations in each substep are the generation of a random number and exchanging the places of two items in the array. The size of the packet headers compared to the packet bodies depend on the compression settings. Similar to the table for Lena in the last section, Table 4.2 shows the number of header and body bytes and the ratio of header bytes to the total number of bytes in the bitstream for the test image shown in 4.5(a) of 1024 × 1024 pixels and at a compression rate of 1 bpp.

Cblk. Size	Layers	Header Bytes	Body Bytes	Ratio
64 × 64	16	1823	129072	1.4%
32 × 32	32	4603	126283	3.5%
16 × 16	32	11379	119510	8.5%
8 × 8	32	25748	105176	18.6%

Table 4.2: Ratio of header bytes to total number of bytes in the bitstream for different compression settings))

As can be seen, the size of the headers increases with the number of codeblocks and the number of layers. With more information in the headers to be permuted, the

scheme needs more computational complexity, but also gives more security, as the number of possible transformations increases. The proposed scheme is significantly more efficient than the straightforward approach that needs to encrypt the major part of the packet data.

4.2.2 *Security*

The complexity of a brute-force attack that tries to obtain the full quality version of the image depends on the number of possible transformations. For practical settings the obtained number of possible combined transformation will be very high (2^{987} for the inclusion information alone in the example used above). However, the complexity of a brute-force attack can only give an upper-bound for security, often there will be easier attacks. This is especially true for partial encryption.

The proposed key-dependent transformations use permutations, which are principally vulnerable to known-plaintext attacks. Another security threat for the proposed scheme is the fact that an attacker can iteratively try to get a better quality than allowed. The attacker can make a guess on the header information (using the information that is preserved by the encryption, e.g., the types of inclusions in the packet), and then iteratively perform systematic alterations on the header information and reconstruct an image. In each iteration, if the quality improves compared to the previous construction, the alteration is accepted, otherwise the previous state is restored.

The problem for the attacker is that there is no measure with which to assess the quality of the obtained image (apart from using a human observer). The demands for an attack increase with the number of resolutions. The lower resolutions will have significantly fewer packets than the higher resolutions, therefore an attack will be easier (and possibly be performable by a human observer) on the lower resolutions, while the protection of the higher resolutions is stronger. However, it should be taken into account that obtaining a version of better visual quality than the preview image is possible without too much effort, and (compression) settings should be chosen accordingly.

Full confidentiality, i.e., no preview image at all and high security for the visual data, cannot be provided by the proposed scheme, even if the packet body data for the lowest resolution is encrypted. This is because the protection for the lower resolutions is significantly weaker than the protection of the higher resolutions, and it will almost always be possible for an attacker to get a rough estimate of the visual content from these lower resolutions.

4.2.3 *Applicability*

Apart from the advantages that are normally gained with format-compliant encryption, the fact that the packet body data remains in plaintext also enables techniques that can be used for indexing, retrieval and classification. For example, visual hashing Norcen and Uhl (2004b) can be applied without the need to decrypt.

Generally, applications that use the packet body data for classification, indexing and retrieval will work with the proposed scheme. Also, schemes that use general information from the header will work. For example, the scheme proposed by Tabesh et al. (2005) uses the number of bytes spent on each subband as a texture classification tool. Although this data is retrieved from the header, this scheme will still work with encrypted headers, as the sum of all codeblock contributions is preserved by the proposed encryption scheme. The possibility to perform these tasks in the encrypted domain can be an important feature, which the packet-body-based techniques cannot provide.

On the other hand, classification, indexing and retrieval techniques based on detailed features of the packet header can be disabled by the proposed scheme. For example, the approach by Liu and Mandal (2001), which uses the number of leading zero bitplanes as a fingerprint, can be disabled, just as the approach by Descampe et al. (2006) which uses a set of classifiers based on the packet header and packet body data.

The proposed transparent encryption scheme offers two advantages: efficiency and format-compliance. As only the header information needs to be encrypted, the proposed scheme has low computational demands, even compared to other partial encryption schemes. Full format-compliance allows to perform various tasks in the encrypted domain, including indexing, classification and retrieval techniques. The quality of a preview image for transparent encryption can be controlled quite precisely by choosing an appropriate point in the sequence of packets where to start header transformation.

As regards security, the scheme definitely and expressly has to be denoted lightweight. The use of permutations makes it vulnerable to known-plaintext attacks, iterative attacks can obtain higher quality images than the preview image. However, the full quality version remains reasonably secure, considering the low effort used for encryption.

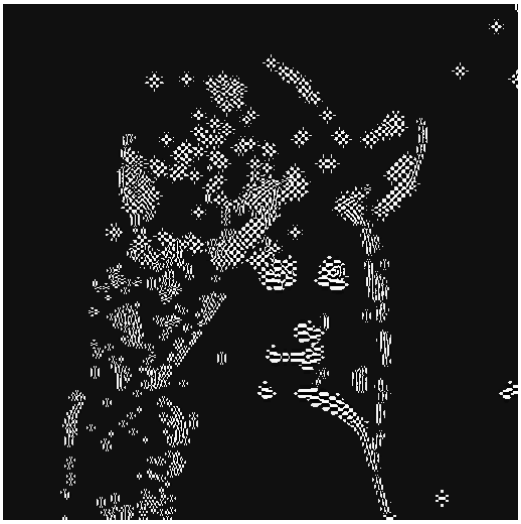
4.3 CONCLUSION

In the context of encryption schemes that aim at sufficient distortion of visual quality rather than at providing full confidentiality, we have shown that the proposed header protection scheme can help to improve the level of guaranteed visual distortion.

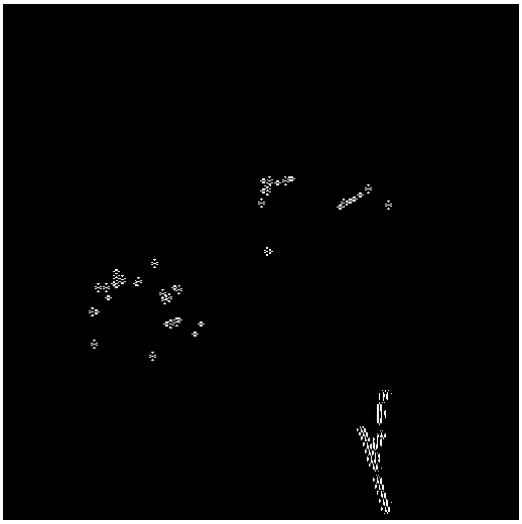
We have also shown that if applied on its own the proposed method can be used as a lightweight transparent encryption scheme that provides reasonable protection of the full quality visual data at very low computational cost.

(a) 1% of body data encrypted, PSNR 13.7, ESS 0.57

(b) 10% of body data encrypted, PSNR 12.9, ESS 0.4

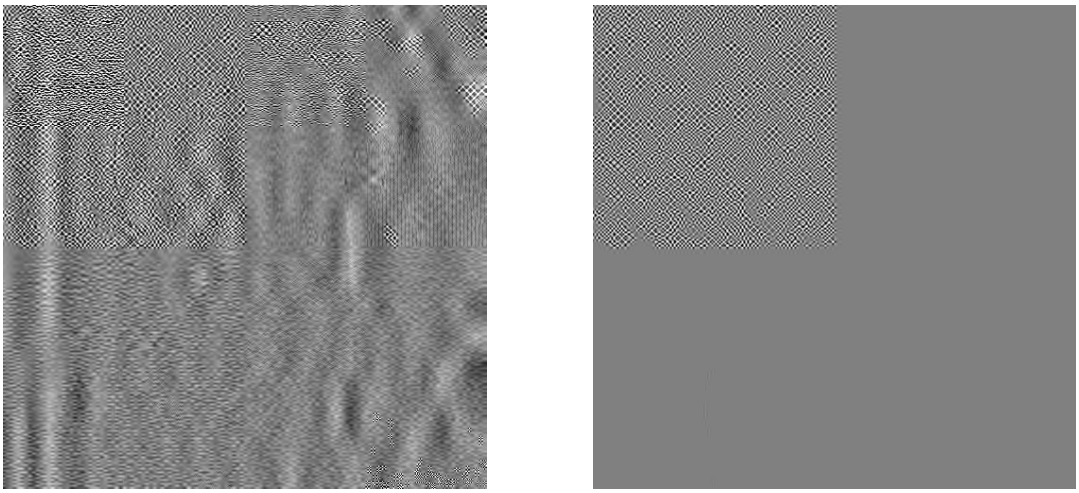


(c) 60% of body data encrypted, PSNR 12.6, ESS 0.003



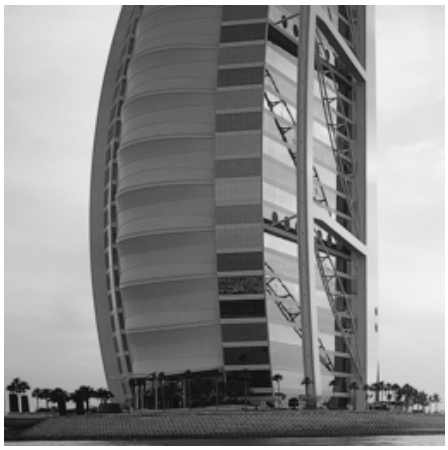
(d) 85% of body data encrypted, PSNR 12.5, ESS 0

Figure 4.3: Visual examples of packet body only encryption (reconstructions use zero concealment, histogram equalized for (c) and (d))

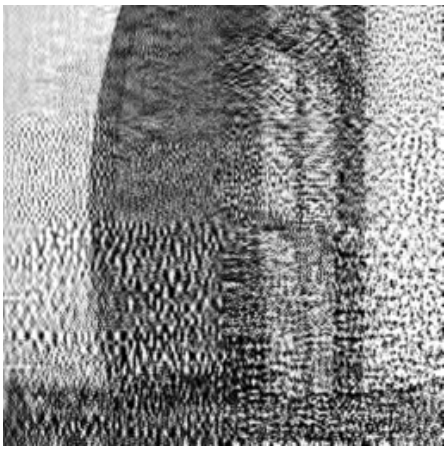


(a) direct reconstruction (b) zero concealment

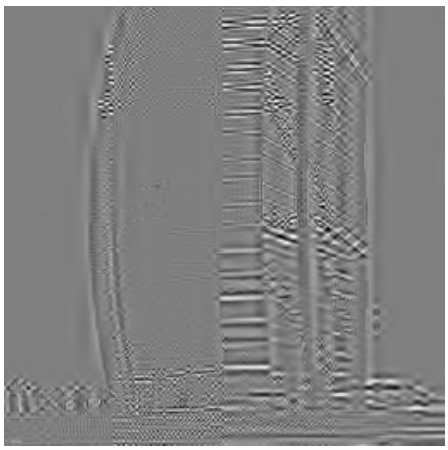
Figure 4.4: 1% of body data encrypted plus all packet headers transformed



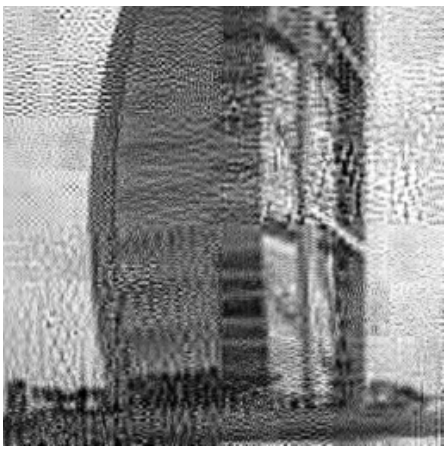
(a) Test image



(b) CCP lengths (& number of coding passes), PSNR 9.7 dB

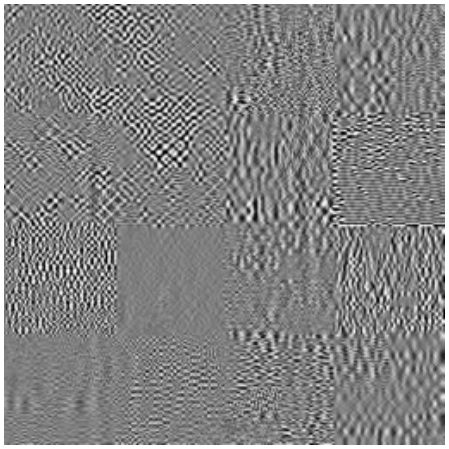


(c) Number of lzb, PSNR 10.6 dB

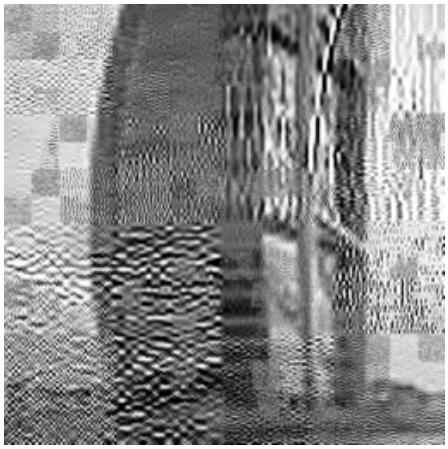


(d) Inclusion information, PSNR 12.1 dB

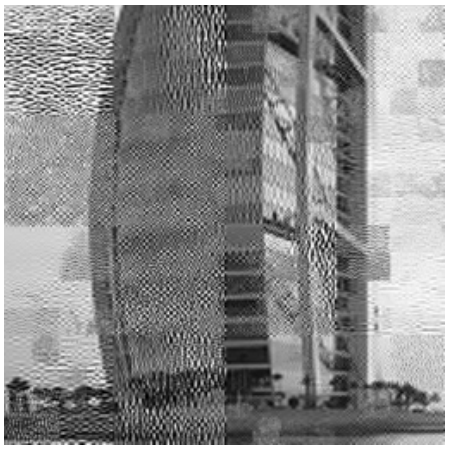
Figure 4.5: Test image & visual examples of reconstructions with transformed packet headers: 1 bpp, cblk. size 32 × 32, 32 layers



(a) Transformations start at resolution 0, PSNR 7.2 dB



(b) Transformations start at resolution 2, PSNR 9.7 dB



(c) Transformations start at resolution 3, PSNR 9.8 dB

(d) Preview image for 4.6(c) (yields resolution 2), PSNR 10.2 dB

Figure 4.6: Visual examples of reconstructions for all transformations and different levels of transparency

Part II

PARAMETERIZED WAVELET FILTERS

5 PARAMETERIZED WAVELET FILTERS

We shift the focus from bitstream-oriented methods to compression-integrated methods in the two following parts. In this first chapter (cf. Engel and Uhl, 2005a), we present a lightweight compression-integrated transparent encryption scheme for JPEG2000 that is based on and integrated into the wavelet lifting scheme. Parameterized biorthogonal filters are constructed based on a key. The proposed method comes at extremely low computational cost and provides natural support for transparent encryption.

A wavelet parameterization based on the lifting scheme (Sweldens, 1996; Daubechies and Sweldens, 1998) is used to produce a family of biorthogonal wavelet filters and construct a key from the parameters. The employed filters significantly improve the compression performance as compared to previously suggested parameterization types. Additionally, we increase the size of the available keyspace by using secret non-stationary and inhomogeneous wavelet decompositions. This method, which produces a bitstream compliant to JPEG2000 Part 2, can be regarded as a special kind of header encryption since only the definitions of the filters used during wavelet decomposition need to be encrypted, the data itself (i.e., packet bodies, packet headers) remain in plaintext. As a consequence, only minimal computational encryption effort is required. In terms of realizing transparent encryption, the proposed approach produces bitstreams from which images of degraded visual quality can be decoded with any decoder that is compliant to JPEG2000 Part 1. In order to get the full quality version, the correct key has to be obtained and a codec compliant to JPEG2000 Part 2 has to be used for decoding.

We discuss the advantages and disadvantages of this encryption scheme with respect to keyspace, computational demands, compression performance, and security.

5.1 RELATED WORK

Recent work by Köckerbauer et al. (2004) assesses the feasibility of using parameterized orthogonal filters (adapted from Schneid and Pittner (1993)) for lightweight encryption within JPEG2000. The compression performance of the obtained orthogonal filters remains markedly below the established biorthogonal filters and varies

considerably over the range of parameter values. To overcome the latter deficiency, a heuristic is proposed to avoid filters with low compression performance. Uhl and Pommer (2004) use a parameterization of biorthogonal filters (as proposed by Hartenstein (1997)) which yields some filters that can compete with the established filters. Still the parameterization is ill-suited for the aforementioned purpose as the variation of the obtained filters with regard to their compression performance is even worse than in the orthogonal case, with the worst filters going down below 5 dB in a PSNR comparison. In all of the above approaches, the algorithm for parameterization of the wavelet filters introduces significant computational overhead. Furthermore, with their focus on providing full confidentiality rather than realizing transparent encryption they differ from the approach presented here.

Different parameterization schemes have been employed for various security techniques in a number of wavelet-based codecs. Apart from lightweight encryption, which we discuss below, wavelet parameterizations are also used in other areas of multimedia security: For increasing the security of watermarking schemes, different parameterizations have been investigated by Meerwald and Uhl (2001), Dietl et al. (2002, 2003a,b), Brachtl et al. (2004) and Huang et al. (2004). The lifting scheme has been used in constructing a watermarking method for JPEG2000 (Seo et al., 2001). Meixner and Uhl (2005) aim at increasing security for visual hashes. Laimer and Uhl (2008) use the same lifting parameterization that we use here to construct a secure key-dependent hash in JPEG2000 (based on our modified version of JJ2000).

5.2 PARAMETERIZED WAVELET LIFTING IN JPEG2000

In our approach we use a lifting parameterization of the well-known CDF 9/7 wavelet filter that is presented by Zhong and Jiao (2001) and is based on work by Daubechies and Sweldens (1998). The authors use the construction theorem presented by Cohen et al. (1992) to formulate conditions for the lowpass and highpass filter taps, h and g, respectively,

$$h_0 + 2\sum_{n=1}^{4} h_n = \sqrt{2},\ g_0 + 2\sum_{n=1}^{3} g_n = \sqrt{2} \tag{5.1}$$

$$h_0 + 2\sum_{n=1}^{4} (-1)^n h_n = 0 \tag{5.2}$$

$$g_0 + 2\sum_{n=1}^{3} (-1)^n g_n = 0 \tag{5.3}$$

$$2\sum_{n=1}^{3} n^2 (-1)^n g_n = 0. \tag{5.4}$$

A possible transformation of the CDF 9/7 wavelet into lifting steps is presented in Daubechies and Sweldens (1998).

$$s_n^{(0)} = x_{2n} \tag{5.5}$$
$$d_n^{(0)} = x_{2n+1} \tag{5.6}$$
$$d_n^{(1)} = d_n^{(0)} + \alpha(s_n^{(0)} + s_{n+1}^{(0)}) \tag{5.7}$$
$$s_n^{(1)} = s_n^{(0)} + \beta(d_n^{(1)} + d_{n-1}^{(1)}) \tag{5.8}$$
$$d_n^{(2)} = d_n^{(1)} + \gamma(s_n^{(1)} + s_{n+1}^{(1)}) \tag{5.9}$$
$$s_n^{(2)} = s_n^{(1)} + \delta(d_n^{(2)} + d_{n-1}^{(2)}) \tag{5.10}$$
$$s_n = \zeta s_n^{(2)} \tag{5.11}$$
$$d_n = d_n^{(2)}/\zeta \tag{5.12}$$

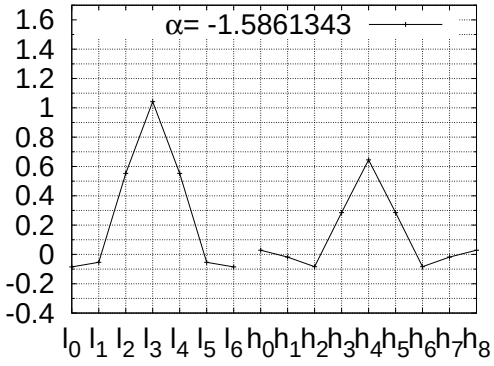
These lifting steps can be used to express the filter taps of h and g as functions of the four parameters $\alpha, \beta, \gamma, \delta$ and a scaling factor ζ. By combining these functions with conditions (5.1)-(5.4), Zhong and Jiao (2001) derive a parameterization that is only dependent on a single parameter α.

$$\beta = \frac{-1}{4(1+2\alpha)^2} \tag{5.13}$$
$$\gamma = \frac{-1-4\alpha-4\alpha^2}{1+4\alpha} \tag{5.14}$$
$$\delta = \frac{1}{16}\left(4 - \frac{2+4\alpha}{(1+2\alpha)^4} + \frac{1-8\alpha}{(1+2\alpha)^2}\right) \tag{5.15}$$
$$\zeta = \frac{2\sqrt{2}(1+2\alpha)}{1+4\alpha} \tag{5.16}$$

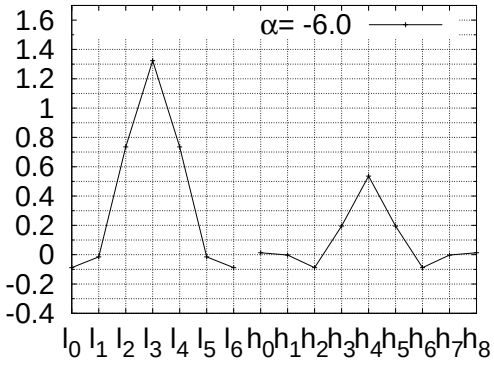
Figure 5.1 shows examples of parameterized biorthogonal filter taps. By setting α to the value originally proposed by Daubechies and Sweldens (1998), $-1.58613\ldots$, the original 9/7 wavelet is obtained. As the parameterization is based on the lifting scheme, it comes at virtually no computational cost. The only computations needed are the evaluations of the four functions for the scaling factor and the parameters other than α, and the calculation of the lowpass and highpass synthesis filter taps for normalization.

Since one of the stated requirements is the retention of compression performance that is comparable to the performance of the original 9/7 wavelet, we first assess the potential range that can be used for α. As illustrated by Figure 5.2, we found that with the exception of the open interval $]-1.4, 0.2[$ the filters produced by the 9/7 parameterization all achieve compression results that are competitive with the original CDF 9/7 wavelet. The filters within the interval $]-1.4, 0.2[$ are not stable and do not achieve highpass and lowpass separation. Therefore they are not suitable for compression, independent of the visual data to be transformed.

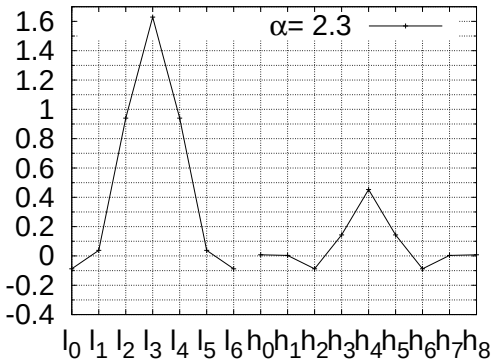
Figure 5.3 shows the compression performance for parameterized filters in terms of the LSS/ESS measure. For the evaluation of the compression performance of the



(a) CDF 9/7



(b) α = -6.0



(c) α = 2.3

Figure 5.1: Examples of parameterized biorthogonal wavelet filter taps

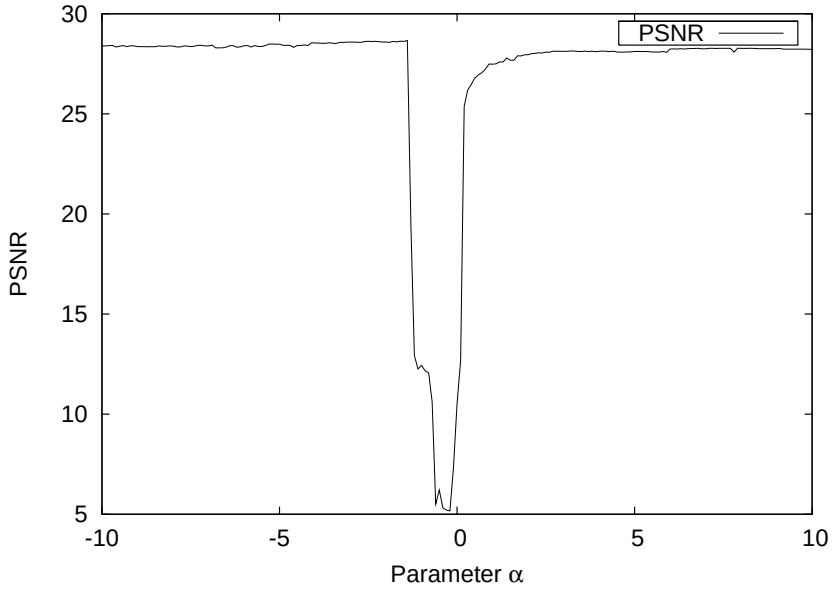


Figure 5.2: Compression performance (PSNR) of parameterized 9/7 wavelet filters for "Lena", rate 0.1 bpp

parameterized wavelet filters both measures behave in a similar way to the PSNR measure. In the next section we employ the measure for the evaluation of attacks.

5.3 KEYSPACE

The potential range for α is further restricted by the fact that in both positive and negative direction the variation of the produced filters drops rapidly with higher α. Figure 5.4 illustrates the problem for encryption. If the source image has been encrypted with a small absolute value for α, the PSNR curve of the reconstructed image with different parameters shows a steep peak for the correct value. Obviously an isolated peak is a favorable situation for encryption as only the correct parameter value will yield the full quality image, while even with small deviation the quality is reduced considerably. With increasing absolute parameter values the peak flattens, making larger absolute values for α increasingly less useful for encryption. These results are in contrast to the results reported by Huang et al. (2004), who use the same parameterized lifting scheme for watermark security and report that even a minute change in the value of α makes correct watermark detection virtually impossible.

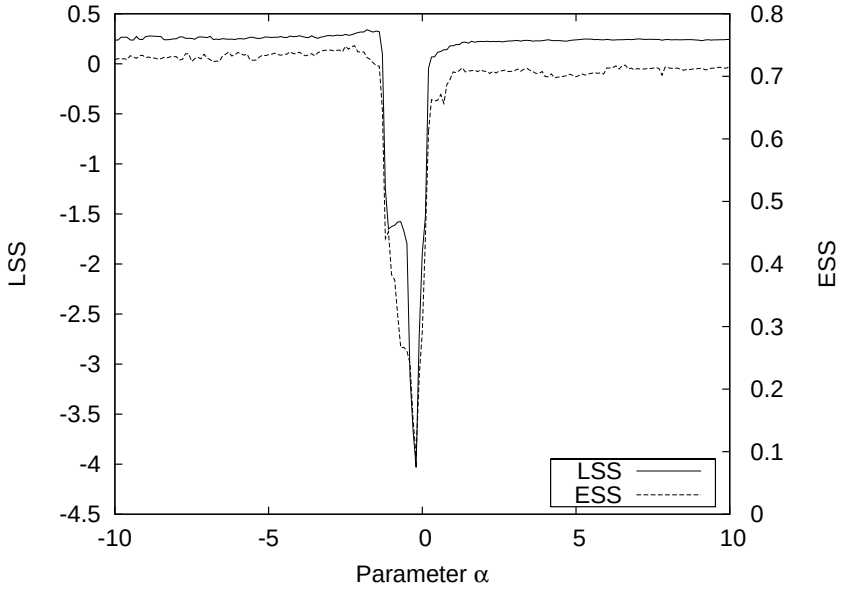


Figure 5.3: Compression performance (LSS/ESS) of parameterized 9/7 wavelet filters for "Lena", rate 0.1 bpp

In consideration of the transparent encryption scheme, the reduction in quality should not occur as high frequency distortions that alter the image beyond recognition. Figures 5.5 and 5.6 show the image quality for the same attacks as shown in the previous figure, but measured in LSS and ESS, respectively. Whereas the graph for LSS is very similar to that of PSNR, it can be seen that the peaks for ESS are less steep than those for LSS and PSNR. This means that distortion is introduced in terms of loss of luminance information rather than loss of structural (i.e., edge) information. Images obtained with a wrong parameter still contain most of the structure of the original image, but lack the correct luminance information. This interpretation is congruent with the visual impression, as illustrated by Figure 5.7, which shows some examples of reconstructed images for "Lena" encrypted with parameter $\alpha = 2.5$: (a) shows the image reconstructed with the correct parameter, (b), (c) and (d) show the image reconstructed with incorrect parameters. For (b) and (c), both of which are reconstructed with a parameter that has the same sign as the parameter used for encryption, it can be seen that, while there is a definite loss of luminance information, much structural information is preserved. The reconstructed image shown in (d) illustrates the fact that for images encrypted with a negative parameter value, image reconstruction with a positive parameter value leads to severe distortion. Still, the little similarity to the original image that remains is structural.

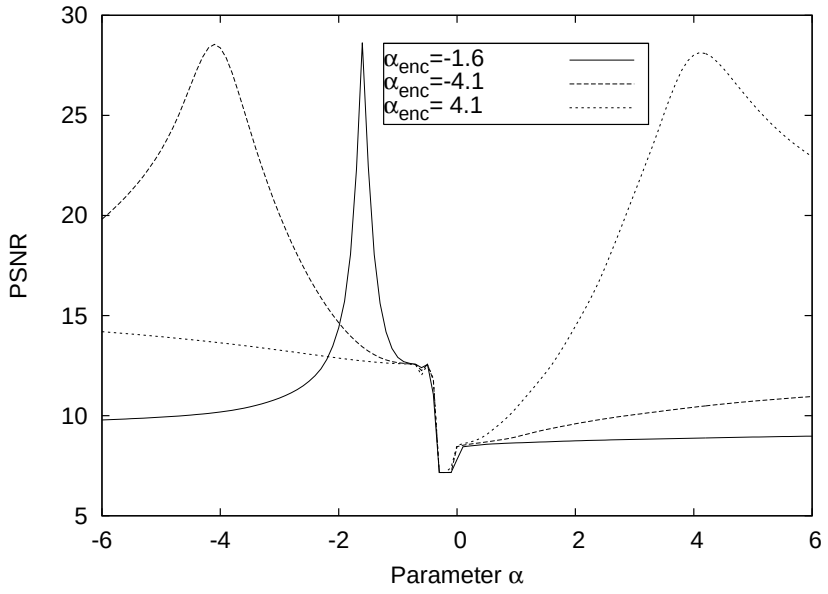


Figure 5.4: Attacks on "Lena" (PSNR), rate 0.1 bpp

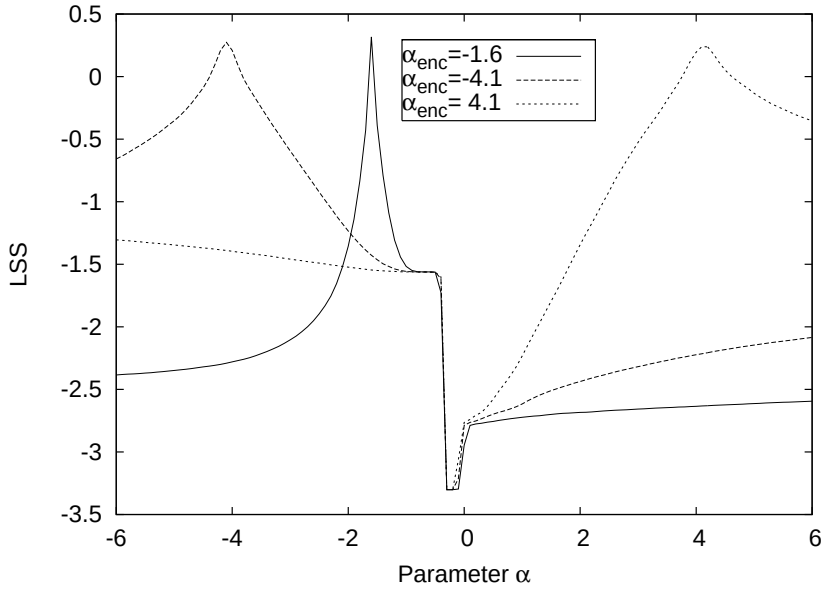


Figure 5.5: Attacks on "Lena" (LSS), rate 0.1 bpp

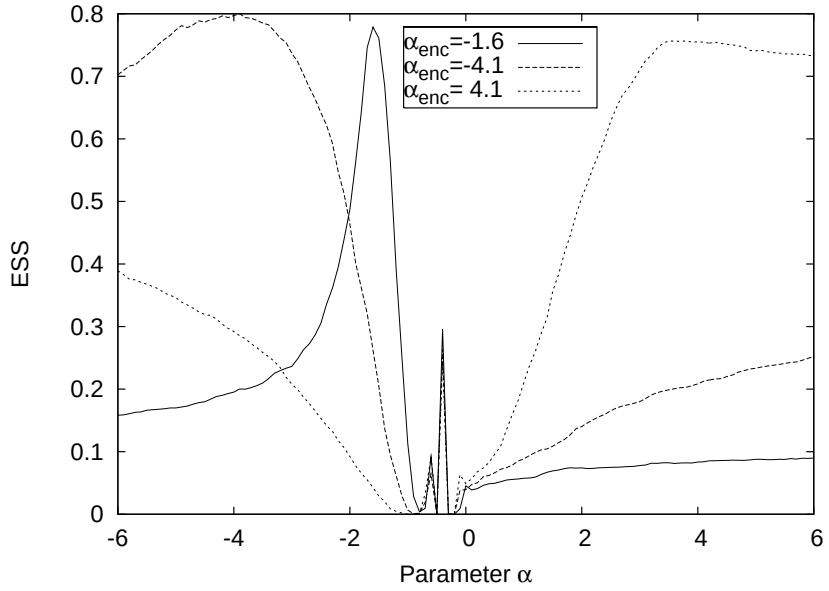


Figure 5.6: Attacks on "Lena" (ESS), rate 0.1 bpp

In order to find a sensible range for α that uses as many values as possible while keeping vulnerability of individual values at a minimum, we introduce a measure for the usability of values of α. In the application setting described in the introduction, it is acceptable to gain access to the visual data with a wrong key, as long as the obtained version is sufficiently degraded in quality. In our test runs, we assume that it is acceptable if the quality of an image decoded with the wrong parameter is below 80% of the PSNR value of the image reconstructed with the correct key. Figure 5.8 shows an evaluation of the parameter space, in which for each α used for encoding, we consider decoding values from the direct neighborhood in the interval $[-6 + \alpha, \alpha + 6]$ (with a sampling step size of 0.1). For all values in this interval for which the quality loss is less than 20%, the PSNR value above the 80% level of the correct value is added to the unweighted measure, which is plotted by the first line. The second line shows the same measure, but with the contributions of individual decoding values weighted by their distance from the correct value. While the unweighted measure reflects the overall performance of a single value, the weighted measure also reflects how fast the PSNR quality degenerates with the distance from the correct value. To a certain degree, the choice for the range of α to be used for encoding will depend on the security requirements of the actual application. In our test runs we use the interval $[-6, 6]$, which, as illustrated by Figure 5.8, comprises most of the usable keys.

(a) Reconstruction with correct $\alpha_{\text{dec}} = -2.5$ (PSNR 38dB, LSS 0.997, ESS 0.95)

(b) Reconstruction with $\alpha_{\text{dec}} = -1.58613$ (PSNR 14.7dB, LSS -1.25, ESS 0.46)

(c) Reconstruction with $\alpha_{\text{dec}} = -6.0$ (PSNR 12.5dB, LSS -1.73, ESS 0.37)

(d) Reconstruction with $\alpha_{\text{dec}} = 2.5$ (PSNR 9.3dB, LSS -2.51, ESS 0.1)

Figure 5.7: Reconstructed images and quality measure results for "Lena" ($\alpha_{\text{enc}} = -2.5$), rate 1 bpp

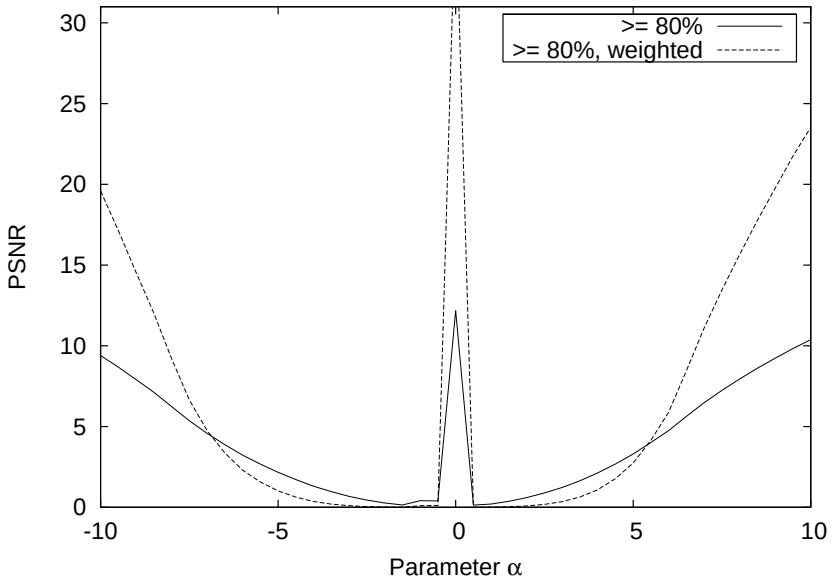


Figure 5.8: Accumulated quality measure of parameterized 9/7 wavelet filters, "Lena", rate 0.1 bpp

As the employed measure is dependent on the properties of the filter rather than on the visual data to be encrypted, we can safely use this interval regardless of the source image. Considering the earlier evaluation of compression performance, we thus get a range of $[-6, -1.4] \cup [0.2, 6]$ that can be used for encryption. As parameters in the immediate vicinity of the value used for encoding still yield more than the desired quality, a large enough stepsize between the discrete values of α has to be chosen.

Altogether, the obtained keyspace in one dimension is quite restricted and hardly suitable for real-life application. In order to enlarge the keyspace, we use different parameters for the horizontal and the vertical wavelet decomposition on different decomposition levels. These techniques have been called "non-stationary" (varying on each decomposition level) and "inhomogeneous" (varying in vertical and horizontal orientation) in the context of adaptive compression (Uhl, 1996). Pommer and Uhl (2001) use this idea without parameterization for selective encryption. Neither of these methods results in a deterioration of compression performance. Similar to the 1D case we get steeper peaks for smaller absolute values. In order to show that non-stationary and inhomogeneous variations of parameters are principally useful, we present the results of test runs with relatively small values for α. For the inhomogeneous case, Figure 5.9 shows the results for encoding values of $\alpha_{\text{hor}} = -1.6$ and $\alpha_{\text{ver}} = -2.1$ on all decomposition levels and a step size of 0.1 in the interval $[-6, 6]$ in each direction for

decoding. It can be seen that with one correct parameter, results of degraded quality can be obtained, but the full quality version can only be accessed with both parameters correct. Figure 5.10 shows the situation for non-stationary variation of α. The sequence of encoding parameters is shown in the caption of each plot. For decoding, we left all parameters on the correct position and only varied the parameter α for levels 1, 3, and 5, respectively, with a stepsize of 0.1 in the interval $[-6, 6]$. Surprisingly, there is little difference between a variation of parameters on a low, medium or high subband. The fact that for all levels a clear peak is produced encourages the use of non-stationary variation. In both, inhomogeneous and non-stationary variation, images reconstructed with wrong parameters retain a certain quality. This is favorable for transparent encryption. However, the sign chosen for encoding has an important impact: if the image was encoded using a negative value for α, then decoding it with a positive value yields worse quality than any negative value. It depends on the requirements of the application, if decoding with an arbitrary parameter, especially the parameter for the original CDF 9/7 filter, is required to always produce an image above a minimum quality level. If this is the case, the parameter range may have to be restricted to negative values of α to ensure decodability with sufficient quality when using a decoder compliant with JPEG2000 Part I.

The combination of non-stationary and inhomogeneous variation of α leads to $2l$ parameters from which the keyspace can be constructed, where l is the number of decomposition levels. Depending on the minimum acceptable quality degradation, the number of partitions p for each of these parameters can be chosen. The size s of the resulting keyspace is p^{2l}. Reversely, the size of the used keys is $\log_2(s)$ bit. For example, using a two-digit hexadecimal number to describe each parameter, we could construct an 80 bit key for a 5 level wavelet decomposition. The resulting discretization of the range of each individual parameter into 256 partitions leads to a step size of approximately 0.04 and largely avoids full quality results for similar values. However, near the edges of the parameter range, we found that this partitioning tends to be too fine. Possible solutions are a non-uniform partitioning, a restriction in parameter space, a higher number of wavelet decomposition levels, or the use of smaller partitions (a single hexadecimal number still yields 40 bit keys, inducing a step size of approximately 0.65).

5.4 SECURITY

Figure 5.11 shows the result of a brute force attack that uses three partitions for each parameter. The encryption was done using a key with two decimal digits per parameter, the compression result of the correct key was added for reference. For the attack shown in this figure, a key with individual parameters of low absolute values was chosen. For

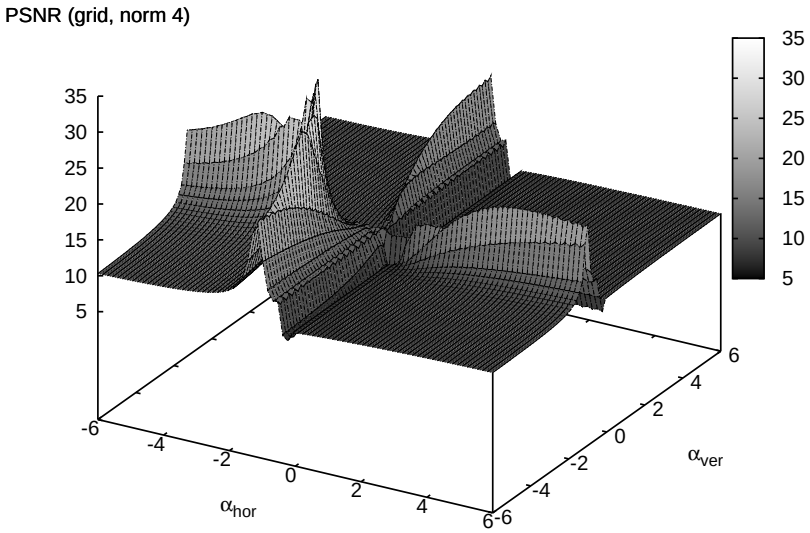


Figure 5.9: Inhomogeneous variation of lifting parameters for "Lena", rate 1 bpp

keys that contain more parameters at the border of the range, similar problems occur as in the 1D case and near-hits achieve high PSNR results. Figure 5.12 shows such a case for a bitrate of 1 bpp. The used key contains values near the border of the range, most notably $\alpha_{\text{hor},2} = -4.1$, $\alpha_{\text{ver},3} = 4.5$, $\alpha_{\text{hor},5} = 4.3$ and $\alpha_{\text{hor},5} = -5.1$. As compared to the previous key, the PSNR of the attacks is significantly higher over the whole range, and the quality of the reconstructed images for key values with many near-hits for the individual parameters (the last third in Figure 5.12) is just below the correct key.

An obvious problem with the presented encryption scheme is that the parameters that make up the key are independent and the searches for the right individual parameter values are separable. Keys that are congruent with the secret key in some positions yield degraded versions of the original visual data. This makes the scheme very vulnerable to known plain text attacks. If the target visual content is not known in plain text, the attack of the full quality version of the source image is more difficult. A conceivable attack is the use of a heuristic search that assesses the "perceptual quality" of reconstructed versions to find the parameters one by one in the relatively small 1D parameter space. A possible choice for such a heuristics is a measure for the

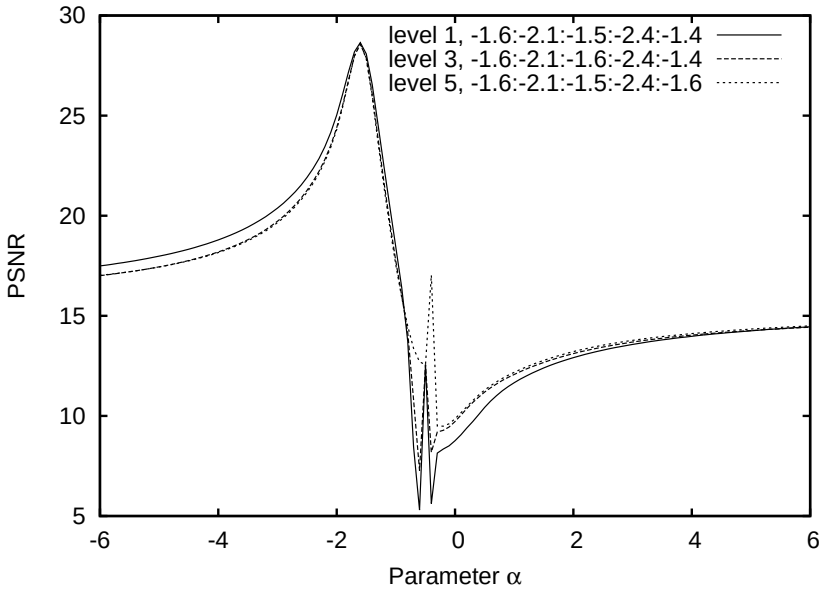


Figure 5.10: Non-stationary variation of lifting parameters for "Lena", rate 0.1 bpp

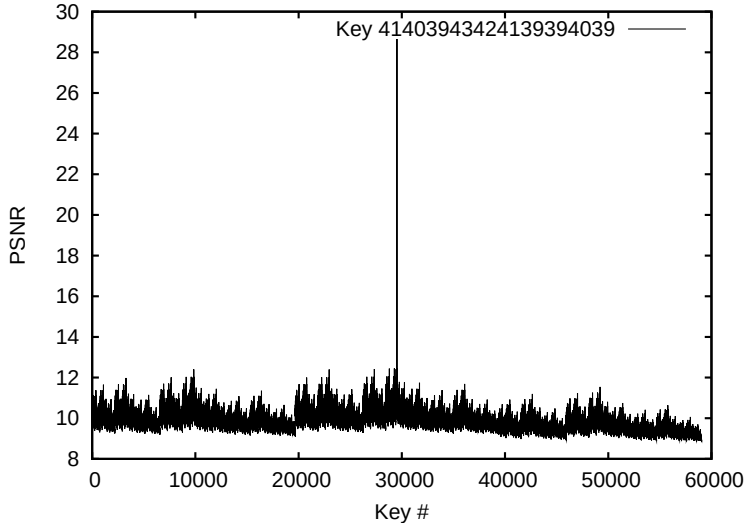


Figure 5.11: Brute force attack on "Lena", rate 0.1 bpp

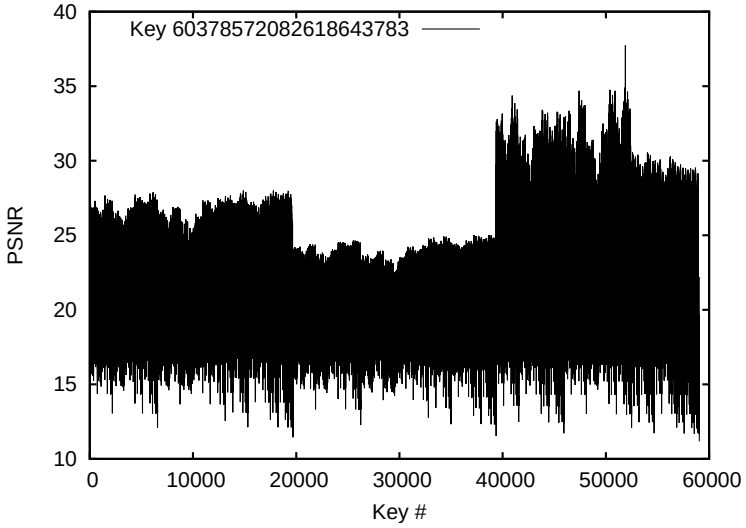


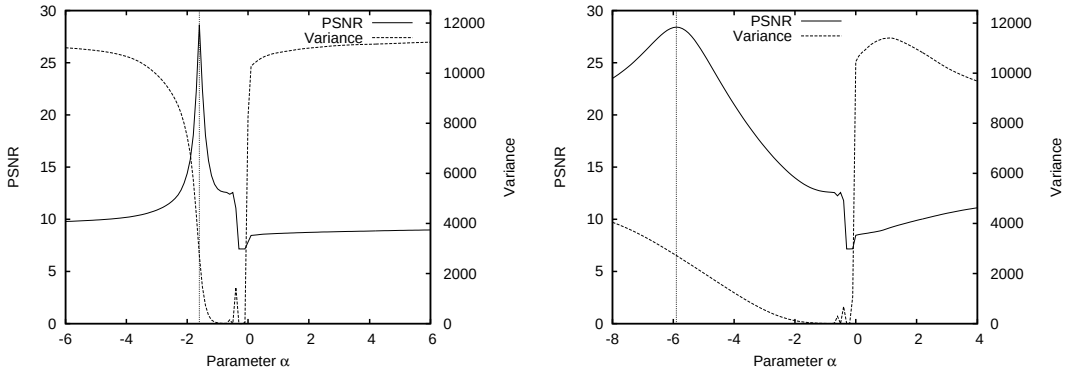
Figure 5.12: Brute force attack on "Lena", rate 1 bpp

smoothness of the reconstructed image, which has been reported to yield expedient attacks on images encrypted using parameterized orthogonal filters (Pommer and Uhl, 2003). The implicit assumption of such an attack is that there is a strong correlation between PSNR and smoothness, and ideally a coincidence of a single maximum in PSNR with a single maximum in smoothness. This being the case would make a number of attacks based on this heuristics feasible, ranging from naive stepwise search for steepest descent to elaborate gradient techniques. Even a simple attack based on such a heuristics would reduce the complexity of the search considerably and could impose a significant degree of vulnerability on the presented approach. To assess the potential of such an attack, we use the sample variance s^2 as an inverse measure of smoothness to test the validity of the correlation assumption,

$$s^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - m)^2,$$

where m is the mean pixel value in the reconstructed image and N is the number of pixels.

Figure 5.13 shows the correlation between PSNR and variance for a single parameter. It can be seen that the strong inverse correlation between a minimum in variance and a maximum in PSNR, which is necessary for the attack described above, does not exist. The attack can be adapted to use the weaker correlation that can be observed: the peak in PSNR never occurring in regions of relatively high variance. This can be used



(a) $\alpha = -1.6$ (b) $\alpha = -5.9$

(c) $\alpha = -4.1$ (d) $\alpha = 4.1$

Figure 5.13: Correlation of PSNR and variance

to make a few "guesses" for each parameter over the whole range and eliminate the ones with high variance to obtain a combination of parameters that will yield an image of comparatively good quality. For a scheme aiming at transparent encryption this vulnerability is not critical, as long as the obtained quality stays below the maximum acceptable quality. As can be seen in Figure 5.13, the correlation between smoothness and PSNR is not strong enough to substantially reduce search complexity for the full quality visual data. Other than for orthogonal filter parameterization, the lifting parameterization is not vulnerable to a heuristic using variance as a measure. This is due to the fact that in the former case, which aims at providing confidentiality rather than transparent encryption, images decoded with the wrong key contain a high amount of high frequency noise. For the presented approach, no such noise is added, rather luminance information (along with a portion of the edge information) is lost.

5.5 CONCLUSION

The main advantage of the presented lightweight encryption scheme is that, while maintaining competitive compression performance, it comes at extremely low computational overhead and innately supports transparent encryption. The relatively restricted keyspace that results from the lifting parameterization can be enlarged by using different parameter values for for the vertical and horizontal wavelet decomposition on each decomposition level. The main problem of our scheme is the separability of these parameters which allows relatively low-cost attacks. However, in many settings that require "soft" encryption, e.g., in the area of mobile multimedia applications, the level of security will suffice.

In the following chapter, we will aim at enlarging the keyspace. A possible approach is a non-uniform partitioning of the possible range of α with more parameters of low absolute value, which produce filters that are less vulnerable to attacks. Another approach which we will investigate is to use wavelet packet decompositions in combination with the best basis algorithm and enlarge the keyspace by using a different parameterized filter for each individual subband.

6 SECURITY ENHANCEMENT FOR PARAMETERIZED WAVELET FILTERS

In this chapter (cf. Engel and Uhl, 2005b), we present improvements in lightweight transparent JPEG2000 encryption with lifting parameterized biorthogonal wavelet filters. As discussed in the previous chapter, if inhomogeneous and non-stationary variation are used, the parameter space is of dimension $2l$ where l is the number of wavelet decomposition levels. A full search of all discrete parameter values (the exact number of which depends on the discretization function used) in this space is not feasible. Only a search with larger discretization bins can be performed, but as has been shown, such a search technique cannot be used to obtain the full quality version of the image. However, for encoding values at the border of the parameter range, close hits in the attack achieve results that are near the original quality, which rules out application scenarios that require a guaranteed degradation in image quality. We will discuss discretization strategies for the parameter range that better reflect the properties of the parameterization and therefore present better security.

Security is further enhanced by a combination with the wavelet packet transform. The wavelet packet transform will be discussed in its own right in all detail in Part III. In the present chapter, we only use the wavelet packet transform to enhance keyspace size. Different methods for the selection of a suitable wavelet packet basis are presented, which also make a certain amount of control in the trade-off between security and computational complexity possible. The combined approach of parameterized filters and wavelet packets is evaluated with regard to compression performance, complexity and security.

6.1 TUNING THE PARAMETER RANGE

In the previous chapter we have established the possible range for α that yields good compression results and maximizes keyspace size as $[-6, -1.4] \cup [0.2, 6.0]$. Discretization is performed in intervals of equal size. The admissible size of the intervals depends on the quality degradation required by the application. In our tests we use 255 bins for discretization. We have found that compression performance for the parameterized filters is influenced by the quantization step signalling strategy chosen for JPEG2000

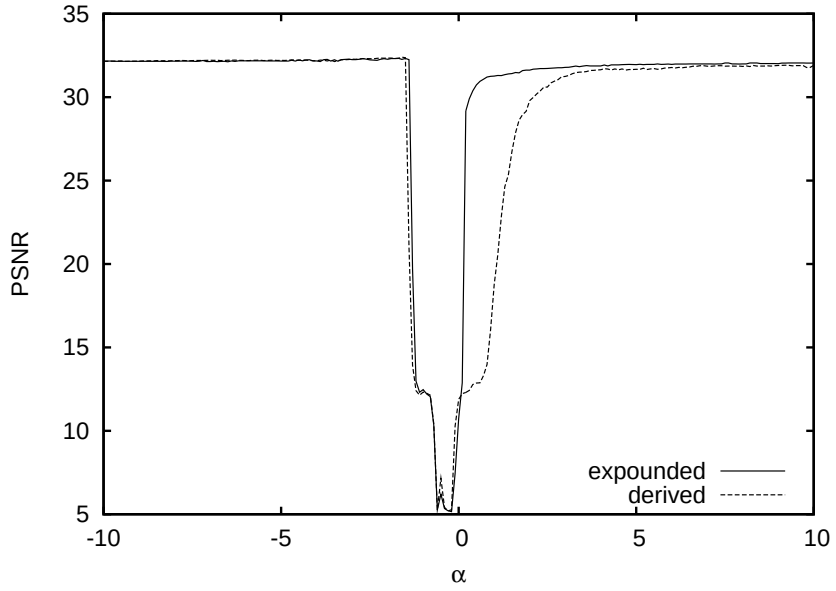


Figure 6.1: Compression performance of parameterized filters for different quantization signalling strategies ("Lena", rate 0.25 bpp)

encoding. JPEG2000 part 1 defines two strategies: if *expounded* signalling is used then quantization steps are coded for each subband individually, whereas if *derived* is used, quantization step signalling is only done for the approximation subband, and the step size is inferred from this value for the detail subbands. As illustrated by Figure 6.1, low positive parameter values generate filters for which *derived* signalling fails. If this strategy has to be used (which is the case for complex wavelet packet decompositions at low bitrate), then the parameter range has to be adapted accordingly.

Interestingly, we also found combinations of parameterized filters, that when used together (by non-stationary and inhomogeneous combination) produced non-competitive results for the expounded signalling strategy. Each of these filters achieves normal compression quality when used "alone" in a classical DWT. Surprisingly, the combinations also work fine with the *derived* strategy. Apparently the combination of these filters produces coefficient data that makes the analysis of the *expounded* strategy fail. In all of these combinations at least one positive filter value is contained, but not necessarily of low value. This leaves two options:

(a) One option is to check the quality of the reconstructed image after encryption. This introduces computational overhead, but preserves keyspace size. If an unsuitable combination occurs, encryption has to be repeated. This event is un-

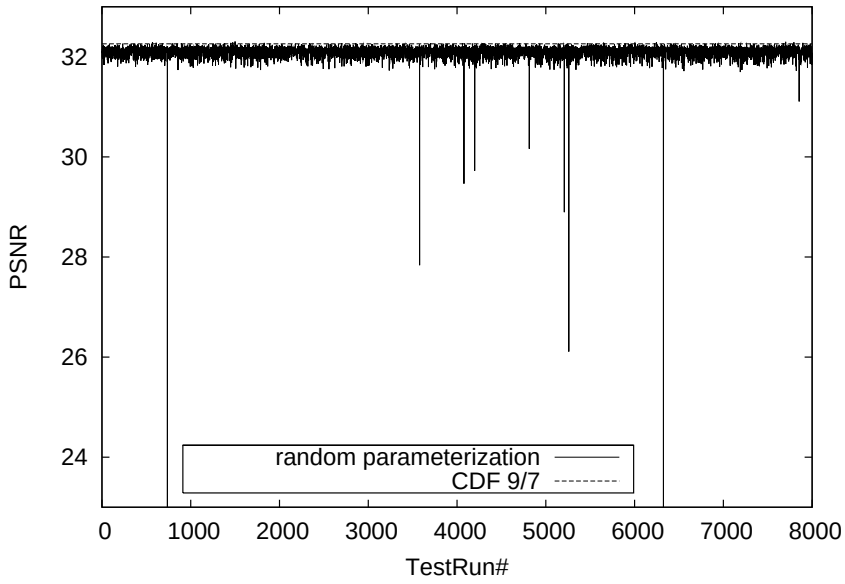


Figure 6.2: Compression performance for random parameterization with *expounded* signalling, ("Lena", rate 0.25 bpp)

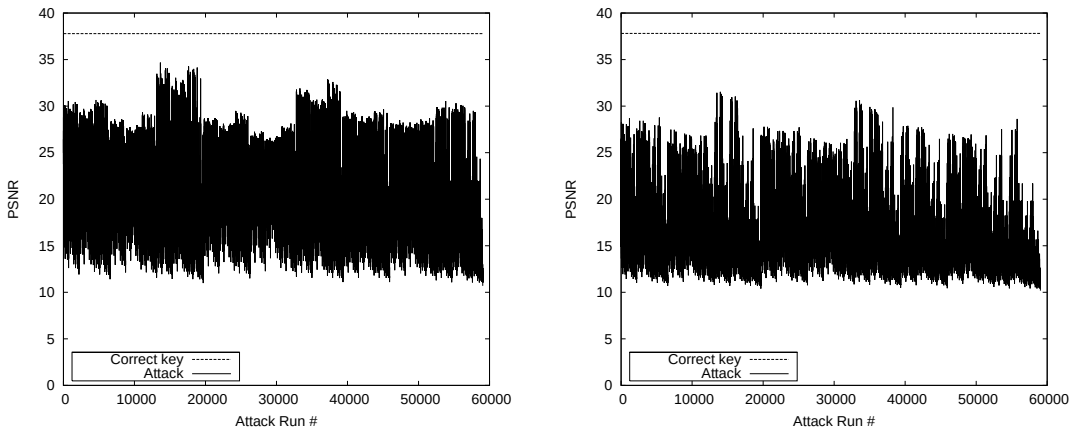
likely, but still the potential overhead in computational complexity might make this option infeasible in some scenarios.

(b) Discarding the positive range leads to a severe reduction in keyspace, but this option completely avoids the unsuitable parameter combinations.

In our tests, we use equal sizes for the negative and positive range, $[-6, -1.4] \cup [1.4, 6.0]$. For this range, Figure 6.2 shows the results of 8000 test runs with randomized non-stationary and inhomogeneous variation for the pyramidal wavelet decomposition with *expounded* signalling. It can be seen that the combinations for which this strategy fails are relatively rare (about 0.1%).

Large absolute values of α_{enc} used for encoding have been shown to be vulnerable to attacks that try to approximate α_{enc}. The environment of values near α_{enc} that yield results of high quality when used for decoding increases in size with larger absolute value of α_{enc}. In order to make the encryption scheme more robust and have parameter values with similar security, we use the square function to partition the range of α_{enc}. Bins nearer the outer borders of the parameter range are made larger than bins near the center (where the filters show more variation).

Figure 6.3 shows that by this measure, simple attacks like probing the keyspace with a number of values for each of the parameters become less effective. The results shown



(a) Uniform discretization (b) Square discretization

Figure 6.3: Examples of discretization strategies, "Lena", rate 1 bpp

are for Lena at rate 1 bpp, with the attack using three guesses for each parameter. Figure 6.3 also shows that the scheme cannot be used for providing strict confidentiality, as any combination of parameters will yield images in which the original content is discernible. For some combinations quality degradation is small, which is not acceptable for most application scenarios.

6.2 WAVELET PACKETS

The wavelet packet (WP) transform (Wickerhauser, 1994) is a generalization of the pyramidal (or Mallat) wavelet transform. In the case of the wavelet packet transform, recursive decomposition is not restricted to the approximation subband, but can also be applied to any of the detail subbands. This results in a larger space of possible decomposition structures, of which the pyramidal decomposition structure is only one element. As all of Part III is concerned with wavelet packets, we will discuss the wavelet packet transform in more detail there. For now we restrict the discussion to the perspective of enhancing the keyspace for the parameterized wavelet filters.

To enhance encryption security, we extend the concepts of non-stationary and inhomogeneous variation in parameters to wavelet packets. Using a pseudo-random number generator, for each subband we create one parameter value for horizontal decomposition and one value for vertical decomposition. For a full level l wavelet packet decomposition this leads to $\sum_{i=1}^{l} 2 \cdot 4^{(i-1)}$ parameters involved, e.g., 682 parameters for level 5. In the pyramidal case there are only $2l$ parameters, i.e., 10 parameters for level 5. Evidently, the advantage of using wavelet packets is a massive increase in keyspace

size. The drawback of using wavelet packets is an increase in computational complexity: Each additional parameter comes at the cost of a wavelet decomposition of a detail subband.

The balance between security and complexity can be regulated by choosing more or less complex wavelet packet decompositions. There are several ways to select a wavelet packet decomposition. Each has different implications for compression performance, complexity and security, which are briefly discussed in the following.

6.2.1 *Pyramidal Wavelet Decomposition*

The pyramidal wavelet decomposition is low in computational demands and yields good compression results for natural images. As discussed above, when parameterized filters are used for lightweight transparent encryption, the level of security is significantly lower than for wavelet packet decompositions.

6.2.2 *Full Wavelet Packet Decomposition*

In the full wavelet packet decomposition recursive decomposition is applied to each subband. If the full wavelet packet decomposition is used, the size of the keyspace is maximized. At the same time, the decomposition and reconstruction of this approach are computationally demanding: The order of complexity for a level l full wavelet packet decomposition of an image of size N^2 is $\mathcal{O}(\sum_{i=1}^{l} 2^{2(i-1)} \frac{N^2}{2^{2(i-1)}})$ compared to $\mathcal{O}(\sum_{i=1}^{l} \frac{N^2}{2^{2(i-1)}})$ for the pyramidal decomposition. A drawback with using the full WP-decomposition is that compression performance drops for most images.

6.2.3 *Best Basis*

The best basis algorithm (Wickerhauser, 1994) applies an additive costfunction to find an optimal wavelet packet decomposition structure for a target image. Because for this purpose first a full wavelet packet decomposition has to be created which is then successively pruned, this method has the highest complexity of all. The advantage of this method lies in the wavelet packet structured being tailored to the source image, which leads to increased compression performance. The size of keyspace depends on the source images used. This fact makes using the best basis an interesting option for databases of images that typically show oscillatory patterns, such as fingerprints or textures. For such images, the wavelet decomposition structures found with the best basis algorithm typically differ markedly from the pyramidal decomposition and compression performance is increased.

6.2.4 *Randomized WP Decomposition Structure*

For natural images the marginal gains in compression performance do not justify the high computational demands of the best basis algorithm. To control the level of complexity in our scheme, we use randomized wavelet packet structures. The algorithm for creating randomized wavelet packet structures is discussed in Part III. It allows to set requirements and limits for the complexity of the resulting wavelet packet structure. Furthermore, it allows to tune the properties of the randomized structures to work well with natural images (at the expense of a smaller number of wavelet subbands compared to a full decomposition). The set of parameters includes maximum global decomposition depth of all subbands, minimum and maximum decomposition depth of the approximation subband, as well as parameters influencing the probability of decomposition for a subband, based on its decomposition depth. It can been shown that the average compression quality for randomized wavelet packet structures with appropriate parameters is only a little lower than the performance of the pyramidal decomposition. This topic is also thoroughly discussed in Part III.

For all wavelet packet decompositions it should be noted that for *expounded* quantization step size signalling, the information overhead increases significantly. Therefore it is necessary to use *derived* signalling of quantization step sizes when complex wavelet packet decomposition with many subbands are used at low bitrates. As discussed above, this makes a reduction in the positive part of the parameter range necessary.

6.3 SECURITY EVALUATION

Attacks that try to search the parameter space become more difficult due to the increased number of parameters and the improved method of discretization. In the following simulated attacks we aim at assessing the contribution that is made to security by these two improvements, so we assume the packet decomposition structure is known to the attacker. Keeping the wavelet packet structure secret can be used to further increase security.

Figures 6.4 to 6.7 illustrate the gain in security that can be achieved by using randomized wavelet filters in combination with wavelet packets. For Figure 6.4, the image "Barbara" was encoded with the pyramidal wavelet transform and uniform discretization was used for the randomized wavelet filters at each decomposition level. Figure 6.5 illustrates the increase in security that can be achieved with the introduction of randomized wavelet packets. Figure 6.6 and Figure 6.7 show the performance for randomized wavelet packet structures in combination with non-uniform discretization for "Barbara" and "Lena", respectively. The random structures were generated with a maximum global decomposition depth of five, a fixed decomposition level for the ap-

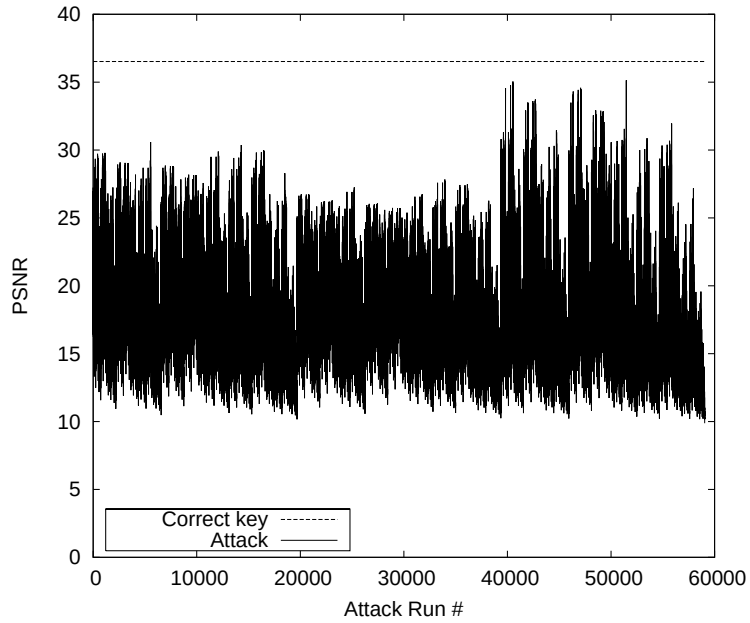


Figure 6.4: Attack on "Barbara" (1 bpp, uniform bins, pyramidal decomposition)

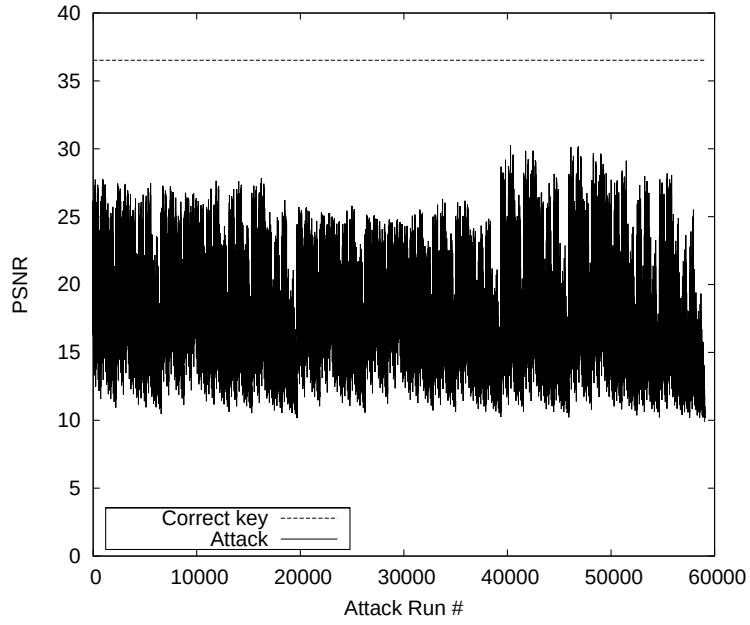


Figure 6.5: Attack on "Barbara" (1 bpp, uniform bins, randomized WP decomposition)

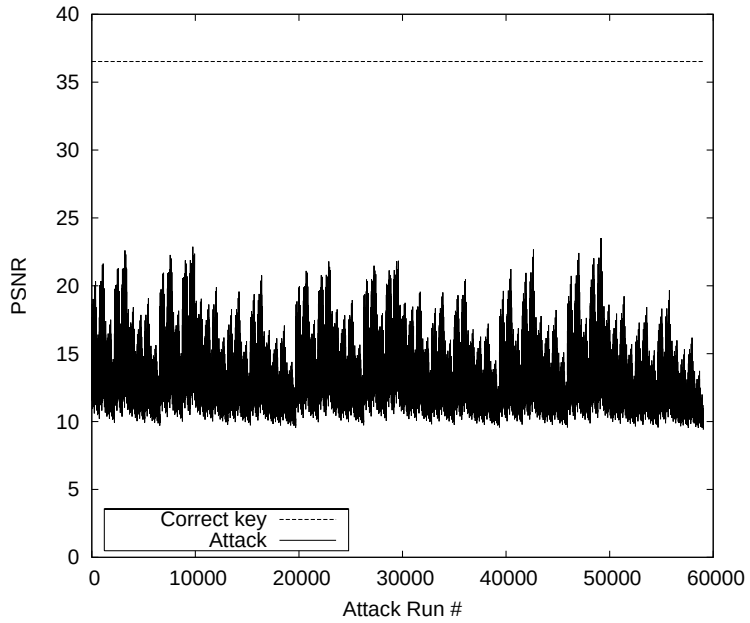


Figure 6.6: Attack on "Barbara" (1 bpp, square bins, randomized WP decomposition)

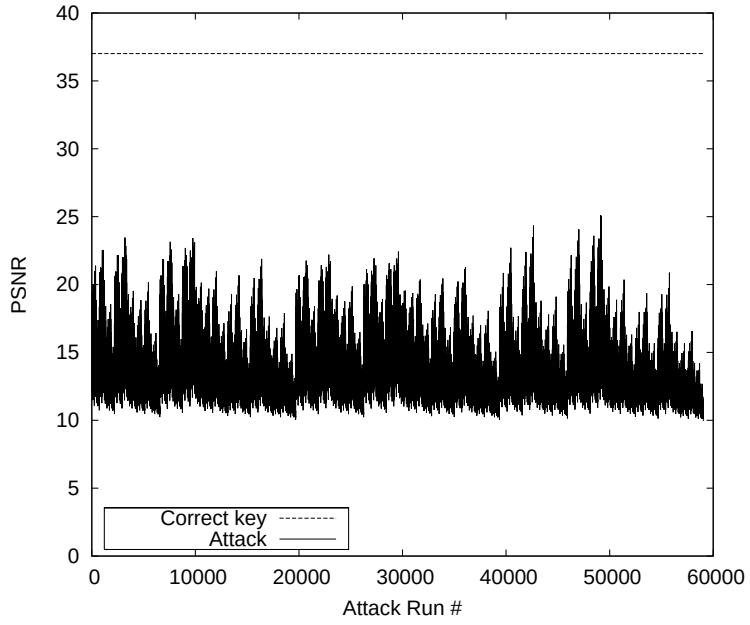


Figure 6.7: Attack on "Lena" (1 bpp, square bins, randomized WP decomposition)

proximation subband of five, and decomposition probabilities that produce wavelet packet structures of medium complexity. For the pyramidal decomposition in combination with uniform discretization some results of the search attack are very near the full quality that should only be obtainable with the correct key (Fig. 6.4). Wavelet packets lead to some improvement (Fig. 6.5). In the case of wavelet packets with non-uniform discretization, the quality degradation is most efficient, and quality for attacked images ranges in a band from about 10 to 23 and 25 dB for "Barbara" (Fig. 6.6) and "Lena" (Fig. 6.7), respectively. For transparent encryption this ensures discernibility, while maintaining a clear degradation of visual quality compared to the original image. Figure 6.8 gives visual examples for both images (attack run #20739).

6.4 CONCLUSION

The proposed approach has the advantage of providing a handle for adjusting the balance between complexity and security. If the pyramidal decomposition is used, a minimum level of security can be achieved at a very low additional computational complexity. We will present a very successful attack on this variant in the next chapter. By adjusting the complexity of the used (randomized) wavelet packet structures, security can be increased at the cost of a rise in complexity. Note that also the computational demands of attacks that try to calculate parameter values by symbolic computation, as will be described in Chapter 7, are significantly increased , because the term to be analyzed (minimized or maximized) becomes increasingly more complex.

A drawback of this approach is that there is a significant amount of variation in the quality of images obtained with wrong keys.

(a) "Lena", original image (b) Pyramidal decomposition, uniform discretization (22 dB) (c) WP decomposition, non-uniform discretization (18.8 dB)

(d) "Barbara", original image (e) Pyramidal decomposition, uniform discretization (23.3 dB) (f) WP decomposition, non-uniform discretization (18 dB)

Figure 6.8: Visual examples of attacks, 1 bpp

7 A SYMBOLIC TRANSFORM ATTACK ON FILTER PARAMETERIZATION

In this chapter (cf. Engel et al., 2006), we present a family of attacks on the lightweight encryption schemes for visual data that have been discussed in the previous chapters. All of the attacks construct a symbolic representation of the inverse wavelet transform. We show that this representation can be used in ciphertext-only attacks, known-plaintext attacks and in attacks in which some information on the plaintext is available. We investigate the success and feasibility of each of these attacks, and conclude that the presented type of attacks poses a principal problem for lightweight encryption schemes that rely on the parameterization of a (linear) transform.

7.1 INTRODUCTION

Some recent propositions for lightweight encryption, like the ones we discussed in the previous chapters, make use of parameterized wavelet transforms to provide security. The family of attacks we discuss here relies on a symbolic representation of the inverse wavelet transform that is constructed for each pixel of the reconstructed image. Depending on the information available to the attacker, this representation can be used in a variety of attacks. We discuss these attacks in the context of parameterized wavelet lifting, as this seems to be the most promising parameterization technique from an encryption point of view, but the principle is applicable to any security scheme that uses wavelet parameterization, or a parameterization of any (linear) transform, to provide lightweight security. In this respect, our main goal here is to discuss the basic feasibility of the symbolic inverse wavelet transform for attacks – sophisticated implementations of the attacks against individual parameterization schemes are not our focus here.

In Chapter 5, we used the biorthogonal lifting parameterization presented by Zhong et al. (2001) with JPEG2000. The attacks we present in this chapter are a threat for the previously presented scheme. Some protection from the symbolic attack can be gained for the scheme by the increase in keyspace size, that we introduced in Chapter 6 by a combination of parameterized wavelet filters with the randomized wavelet packet decompositions, at the cost of higher computational complexity in the transform step.

Most of the attacks discussed here can still theoretically be applied to any of these extensions, but will increase in computational demands and decrease in precision.

7.2 PRINCIPLE OF THE SYMBOLIC ATTACK

As we have discussed in some detail, security schemes that rely on wavelet parameterizations use the degrees of freedom that the wavelet transform provides to produce filters that are suitable for both, providing security and achieving good image compression. Thereby the fact that the wavelet transform is a linear transformation poses a threat for security. Linear transforms are in principle not well suited for keeping information secret. Note that the symbolic attack does not presume a linear transform. It would also work with non-linear transforms that continuously depend on a finite number of parameters. However, with a linear transform it is more likely that the symbolic expressions can be contracted and therefore evaluated faster than when the complete transform has to be performed.

Note that an attacker does not have to obtain the exact parameter value, an approximation is sufficient to yield an image with little distortion. How close the attack value has to be to the encoding value depends on the used parameterization and the used discretization.

The attacks discussed here are based on the symbolical computation of the inverse wavelet transform. Let I be a grayscale image of size $n \times n$ pixels, with luminance values represented as a vector with elements I_i, $i = 0, \ldots, n^2-1$, where I_i is the luminance value of the pixel at position $(i \mod n, \lfloor \frac{i}{n} \rfloor)$. Assume I is decomposed with a parameterized wavelet transformation that depends on m parameters α_j, $j = 0, \ldots, m-1$. The inverse transformation with the correct set of parameters will reconstruct the original image I (assuming, for sake of simplicity, a reversible transform and lossless coding).

An attacker, who does not know the values of α_j, can build a symbolic expression for each pixel value in the reconstructed image containing the necessary operations for the inverse transformation. The resulting term will depend on the values of some of the transform coefficients C_i, $i = 0, \ldots, n^2-1$, all of which are known to the attacker. The only unknowns are formed by the parameters of the wavelet transformation, α_j. By performing a full symbolic inverse wavelet transformation, the attacker can construct a complete symbolic description of the operations necessary to reconstruct I. We illustrate this procedure with an example.

For the parameterization of the CDF 9/7 wavelet, the terms of the symbolic attack become relatively complex. For illustration we therefore construct a parameterized version of the simple Haar wavelet. The lifting steps for the forward transform with the Haar wavelet can be written as follows (Daubechies and Sweldens, 1998):

$$
\begin{aligned}
s_l^{(0)} &= x_{2l} && (7.1)\\
d_l^{(0)} &= x_{2l+1} && (7.2)\\
d_l &= d_l^{(0)} - s_l^{(0)} && (7.3)\\
s_l &= s_l^{(0)} + \frac{1}{2} d_l && (7.4)
\end{aligned}
$$

with the inverse transform written as:

$$
\begin{aligned}
s_l^{(0)} &= s_l - \frac{1}{2} d_l && (7.5)\\
d_l^{(0)} &= d_l + s_l^{(0)} && (7.6)\\
x_{2l+1} &= d_l^{(0)} && (7.7)\\
x_{2l} &= s_l^{(0)}. && (7.8)
\end{aligned}
$$

To construct a simple parameterized version of the Haar wavelet, we change the forward prediction step to

$$
d_l = d_l^{(0)} - \alpha s_l^{(0)}. \tag{7.9}
$$

Accordingly, the forward update step is given by:

$$
s_l = \frac{1}{2}\left((1+\alpha) s_l^{(0)} + d_l\right). \tag{7.10}
$$

The prediction and update steps of the inverse transform are given by:

$$
\begin{aligned}
s_l^{(0)} &= \frac{1}{1+\alpha}(2 s_l - d_l) && (7.11)\\
d_l^{(0)} &= d_l + \alpha s_l^{(0)}. && (7.12)
\end{aligned}
$$

For $\alpha = 1$, the original Haar wavelet is obtained. We denote a horizontal transformation that is followed by vertical transformation by two letters. For example, ds refers to the subband that contains the lowpass transform coefficients for horizontal decomposition and the highpass subbands of the subsequent vertical decomposition.

Imagine that this parameterization is used in a lightweight encryption scheme for an image I of size n^2. For this purpose, α_e, the parameter value used for encryption, is chosen randomly from the range of admissible values. This range would first have to be determined, based on compression performance. (Of course, the Haar filter is not well suited for image compression, and such a parameterization even less.) For our

example we assume $\alpha \in [0.5, 3]$. After transformation with α_e, reconstruction with a wrong α_d, the parameter value used for decryption, will yield a distorted image. An example for the Haar parameterization is given in Fig. 7.1(a), for $\alpha_e = 1.1$ and $\alpha_d = 1.5$.

For the proposed attack, the attacker symbolically computes the inverse wavelet transform. Let S be an $n \times n$ matrix to hold the symbolic expressions. Initially each entry $S_{i,j}$ of this matrix is filled with the representation of the corresponding transform coefficient $C_{i,j}$. Then the operations for each step of the wavelet reconstruction that pertain to a certain position (i, j) are recorded symbolically in $S_{i,j}$. After the full inverse wavelet transformation the complete reconstruction of the whole image is described symbolically by S. Each entry $S_{i,j}$ represents the necessary operations to reconstruct the pixel value at position (i, j).

As an example, consider an image of size 4×4 pixels, for which an attacker wants to construct $S_{2,2}$, the symbolic representation of the pixel at position $(2, 2)$. We assume that a one-level wavelet analysis was done in-place. We denote the vertical and horizontal wavelet transformations by operators F_v and F_h, respectively. The following matrix shows which subband coefficients the entries in the symbolic matrix correspond to for the one-level wavelet transformation:

$$C = F_v F_h S = \begin{pmatrix} ss_{0,0} & sd_{0,0} & ss_{0,1} & sd_{0,1} \\ ds_{0,0} & dd_{0,0} & ds_{0,1} & dd_{0,1} \\ ss_{1,0} & sd_{1,0} & ss_{1,1} & sd_{1,1} \\ ds_{1,0} & dd_{1,0} & ds_{1,1} & dd_{1,1} \end{pmatrix}. \tag{7.13}$$

Initially $F_v F_h S_{2,2}$ contains the representation of the last coefficient in the LL-subband $C_{2,2}$, or $ss_{1,1}$ in the notation used above. After the first vertical analysis transform, $F_h S_{2,2}$ contains the operations necessary to obtain $ss^{(0)}_{1,1}$:

$$ss^{(0)}_{1,1} = \frac{1}{1+\alpha}(2 \cdot ss_{1,1} - ds_{1,1}), \tag{7.14}$$

which translates to

$$F_h S_{2,2} = \frac{1}{1+\alpha}(2 \cdot C_{2,2} - C_{3,2}). \tag{7.15}$$

The reversal of the splitting step makes $F_h S_{2,2}$ contain $s_{2,1}$. The symbolic synthesis step of the horizontal decomposition yields $s^{(0)}_{2,1}$:

$$s^{(0)}_{2,1} = \frac{1}{1+\alpha}(2 \cdot s_{2,1} - d_{2,1}), \tag{7.16}$$

where $d_{2,1}$ has been constructed in position $F_h S_{2,3}$ in the same way as $F_h S_{2,2}$, and is given by

$$F_h S_{2,3} = \frac{1}{1+\alpha}(2 \cdot sd_{1,1} - dd_{1,1}) \tag{7.17}$$

$$= \frac{1}{1+\alpha}(2 \cdot C_{2,3} - C_{3,3}). \tag{7.18}$$

(a) Haar parameterization, $\alpha_e = 1.1$, $\alpha_d = 1.5$ (b) CDF 9/7 param., $\alpha_e = -2.5$, $\alpha_d = -6$

Figure 7.1: Reconstructions with wrong parameters

Putting it all together, we obtain the final $S_{2,2}$:

$$S_{2,2} = \frac{1}{(1+\alpha)^2}\left(2\cdot\left(2\cdot C_{2,2} - C_{3,2}\right) - \left(2\cdot C_{2,3} - C_{3,3}\right)\right). \tag{7.19}$$

Assume that like the image, the matrix is represented as a one-dimensional vector that contains the concatenated lines of the matrix. Let the elements of this vector be denoted by S_i with $i = 0, \ldots, n^2-1$. For example, for $n = 32$, the Haar parameterization and a level one horizontal and vertical transformation, the entry S_{24}, is then given by:

$$S_{24} = \frac{2\left(\frac{\alpha(2C_{12}-C_{29})}{1+\alpha} + C_{28}\right) - \frac{\alpha(2C_{524}-C_{540})}{1+\alpha} - C_{540}}{1+\alpha} \tag{7.20}$$

If a 32×32 version of the Lena image has been transformed with $\alpha_e = 1.4$, then by inserting this parameter along with the transform coefficient values into the equation above, S_{24} takes the value of $I_{24} = 129$, the correct pixel value in this position.

As the transform coefficients are known to the attacker, the parameters of the wavelet parameterization are the only unknown part of the symbolic expressions. If no information on the reconstructed image is available, trying to derive the correct settings of the parameters corresponds to a ciphertext-only attack. A situation in which the reconstructed image is fully or partly available corresponds to a full or partial known-plaintext attack.

7.3 ATTACK SCENARIOS

7.3.1 *Ciphertext-Only*

One possible attack on security schemes that are based on parameterized wavelet filters is the use of a function correlated to image quality. By applying such a measure to the symbolic equations, a single term with the filter parameters as the only unknowns can be obtained. Their values can then be determined by minimizing (or maximizing, depending on the used measure) this term, e.g., by analyzing the first derivative or by employing numerical methods. If there is more than one solution, the possible solutions can be tested very quickly for their usability.

No function exists that can accurately determine if a given image is a natural image. However, as natural images tend to exhibit a certain amount of smoothness, possible indicator functions for image quality could be (inverse) measures of smoothness. A first approach is to use the sum of absolute difference between neighboring pixels and minimize this term. For this purpose the symbolic inverse wavelet transform S is computed for the desired wavelet transformation, as described above. Then the pixel difference p is given by

$$p = \sum_{i=0}^{n^2-2} |S_i - S_{i+1}|. \tag{7.21}$$

The utility of the absolute difference of neighboring pixel values, as a first candidate for such a function, depends very much on the distortion introduced by the used parameterization. If high frequency noise is introduced for wrong parameter settings, then the pixel difference should yield a good indicator for image quality. For the Haar parameterization, the introduced distortions are indeed situated in the higher frequencies. As an example we transform a version of Lena of size 32×32 pixels with $\alpha_e = 1.1$ at a nearly lossless bitrate. Minimizing the symbolic representation of the pixel difference p, we obtain a value of $\alpha_d = 1.17$. A reconstruction with this parameter value yields a PSNR of 40dB.

In the case of biorthogonal lifting parameterization the distortions introduced for wrong parameter settings are of a different kind, as illustrated in Figure 7.1(b). Rather than introducing noise, wrong parameter values produce more of a blurring and smoothing effect (which is one of the reasons why this particular parameterization is well suited for transparent encryption). Thus, the pixel difference attack fails for this parameterization.

The sample variance as an inverse measure of smoothness is another candidate for an image quality indicator. For a symbolic inverse wavelet transform S, the sample variance s^2 is given by:

$$s^2 = \frac{1}{n^2 - 1} \sum_{i=0}^{n^2-1} (S_i - m)^2, \tag{7.22}$$

where m is the mean pixel value

$$m = \frac{1}{n^2} \sum_{i=0}^{n^2-1} S_i. \tag{7.23}$$

For the Haar parameterization, the success of this attack is mediocre. For example, we take a transformation with $\alpha_e = 1.1$ for a 32×32 pixel version of Lena at nearly lossless coding. The symbolic representation of the variance is computed and by minimizing this term we obtain a value of $\alpha_d = 1.27$. While a reconstruction with this parameter value still yields a PSNR of 32.2 dB for nearly lossless coding, the result is relatively far from the expected target. The reconstruction result is illustrated in Figure 7.2(a). For the parameterization techniques used in Pommer and Uhl (2003), the sample variance as an inverse measure of smoothness yields more successful attacks.

For the biorthogonal lifting parameterization, variance does not provide a strong indicator for image quality either. Again, this is due to the nature of the parameterization: rather than introducing high frequency artifacts for wrong keys, the parameterization exhibits a reduction of energy in the highpass subbands. As we have discussed in Chapter 5, the correlation is too weak to substantially decrease search complexity for the full quality image. However, even if the correct parameter cannot be determined precisely, a minimization of the variance can lead the attacker in the right direction. As an example, we transformed an 8×8 pixel test image with a one level wavelet transformation with $\alpha = -1.6$ and then computed the symbolic equations. A minimization of the variance of the equations for $\alpha \in [-6, -1.2]$ yields the solution $\alpha = -1.48$.

As a possible counter-measure for the ciphertext-only attack, the use of different parameters on different resolutions and decomposition directions, i.e., non-stationary and inhomogeneous variation, can be employed (see Chapter 5). For security schemes relying on the biorthogonal lifting parameterization, this increases the number of keys to $2l$ where l is the wavelet decomposition depth. Finding the sequence of parameters from a single equation is not possible, and the attack can only serve to limit the range of possible parameters.

For a single parameter, a problem for this attack is that there is no function that is correlated to image quality for all parameterization techniques. However, the fact that so far no suitable predictor for image quality could be found that works for all parameterizations, does not rule out the existence of a measure that can achieve a general correlation to image quality, which would make this attack a more serious

(a) Haar ciphertext-only attack, $\alpha_e = 1.1$, $\alpha_d = 1.27$

(b) Average luminance value, CDF 9/7 $\alpha_e = -1.6$, $\alpha_d = -1.55$

Figure 7.2: Reconstruction examples for symbolic attacks

threat. In any case, this kind of attack can be used to provide a starting point in a brute-force search and possibly narrow down the range of parameters to be tested.

7.3.2 *Full/Partial Known-Plaintext*

The previous version of a symbolic attack is a ciphertext-only attack, and depends on the existence of a predictor function for image quality. Other versions of the attack do not depend on the existence of such a function. They assume that (parts of) the plaintext, or at least some information on the plaintext is available, which is a realistic assumption with lightweight encryption in general, and with transparent encryption in particular.

Full Known-Plaintext

If the full reconstructed plaintext image is available to the attacker, then the attacker can easily determine the used wavelet parameters by solving the equations for the pixels of the reconstructed image. A situation in which the full plaintext is available is rather unlikely, but a scenario could be conceived in which the attacker has obtained the ciphertext of a set of images (all encrypted with the same α_e) and has received

the full plaintext of a single image from this set in full resolution and quality as an incentive for buying the whole set.

In the case of the biorthogonal lifting parameterization, the set of equations provides the attacker with ample information to deduce the correct value of α. If inhomogeneous and non-stationary variation as discussed above was used for transformation, instead of one parameter α, the attacker has to derive a sequence of parameters $\alpha_0, \ldots, \alpha_{2l-1}$, where l is the wavelet decomposition depth. Even in this case, for images of sufficient size, the attacker will have enough information to derive the correct sequence. The same is true for parameterizations that depend on more than one parameter.

Note that if the full plaintext image is available, other attacks become feasible as well. In many of the parameterizations, the PSNR for different admissible parameter values is a monotonous function that reaches its peak for the correct parameter value. All the attacker needs to do is to approach the correct reconstruction parameters by iteratively reconstructing the ciphertext coefficients and comparing the PSNR of the resulting image to the PSNR of the available plaintext image.

Partial Plaintext Information

In many cases it is not necessary for the attacker to have access to the full plaintext. Symbolic attacks can be conceived that utilize only minimal information about the plaintext image. This assumption is realistic, as many of the proposed schemes support transparent or sufficient encryption. For sufficient encryption, the scheme tolerates image reconstructions with the wrong parameters which yield a discernible version of the original visual data. The only assertion of the scheme is that these reconstructed versions do not exceed a certain quality threshold. For transparent encryption (Macq and Quisquater, 1995), the scheme does not only tolerate image reconstructions of reduced quality, but uses them as a preview image. Such a preview can be of advantage in "try-and-buy"-scenarios, where they serve as an incentive to acquire the correct key to obtain the full quality version of the visual data. In both cases, a potential attacker can make use of the information provided by the reconstructions obtained with wrong parameters.

PLAINTEXT PIXEL SAMPLES For this attack, the attacker is assumed to have obtained individual pixel samples from the reconstructed image. This can for example be achieved by access to a preview image: the attacker selects homogeneous regions or edges that are likely to have the same luminance values in the full reconstructed image. Equating these values with the symbolic representation of the appropriate pixel position, the attacker can construct a linear system of equations, with the parameters

of the wavelet transform as the only unknowns. In the pixel sample attack, the actual or approximate value I_i is known for a number of symbolic terms S_i.

In the case of the Haar parameterization, a single correct pixel value is sufficient to determine the value of α used for encoding. Also for the biorthogonal lifting parameterization, a single correct pixel is sufficient to produce the correct value of α. As an example, we used a 64×64 pixel version of the Lena image. This image was transformed in JPEG2000 at a nearly lossless bitrate of 5 bpp using one level of wavelet decomposition, with the biorthogonal lifting parameterization, $\alpha = -1.6$. Solving the symbolic equation for, e.g. I_{26} with the correct luminance value yields 7 solutions, 6 of which lie in the complex space. The remaining solution is the correct parameter $\alpha = -1.6$.

With inhomogeneous and non-stationary variation and for higher-dimensional parameterizations, more pixel values are needed for an accurate attack, at least as many as the number of filters involved in the parameterization. Depending on the amount and accuracy of information regarding plaintext pixels, the attacker can obtain a more or less accurate solution for the used wavelet parameters.

AVERAGE LUMINANCE VALUE For this attack, the attacker obtains the average luminance value of the reconstructed plaintext image. A good approximation can usually be obtain if a preview image is available. From the symbolic plaintext equations, a symbolic representation of the average luminance value is constructed, i.e.,

$$L = \frac{1}{n^2} \sum_{i=0}^{n^2-1} S_i. \tag{7.24}$$

Then the attacker equates this expression with the obtained value of the average luminance of the reconstructed plaintext image to obtain the parameter value of the transformation. The accuracy of the derived parameter values depends the accuracy of the obtained average luminance value.

As an example we tested this attack with the 64×64 pixel version of the Lena image, using the biorthogonal lifting parameterization with nearly lossless settings and $\alpha = -1.6$. The reconstructed image has a PSNR of 50.2 dB. The mean pixel value of the reconstructed image before quantization to an integer value is 99.0217. If an attacker obtains the average luminance value of 100 from a preview image, $\alpha = -1.55099$ can be derived. This leads to a reconstructed image quality of 42.2dB. (Note that the fact that this attack works shows that the used parameterization does not strictly conform to the construction theorem, because if it did, the average luminance value would be preserved even for wrong reconstruction filters.) An example for a reconstructed images for this case is given in Figure 7.2(b).

As the average luminance value only produces a single equation, it cannot be used for parameterizations that use more than one parameter. Inhomogeneous and non-

stationary variation therefore make the attack unusable. However, if the average luminance value is available for higher-dimensional parameterization, it can be used in conjunction with the other attacks to reduce their complexity.

7.3.3 *Computational Complexity*

In this work we focus on proving that attacks that construct a symbolic inverse transform are successful against encryption schemes that employ wavelet parameterizations to provide lightweight security. To prove this point, the use of a symbolic computation without any optimization at all, i.e., each step of the lifting is computed individually over and over again, is sufficient. We use a straightforward symbolic representation of the inverse wavelet transform. An adapted version of JJ2000 computes the equations of the inverse parameterized wavelet transform and provides output that can be read into Mathematica®, where the equations for the reconstructed pixels are stored in a matrix. Each entry of the matrix corresponds to a pixel in the image and holds the symbolic expression for the reconstruction of this pixel from the transformed image, i.e., the ciphertext.

For an efficient attack, the lifting steps themselves should be represented symbolically and processing should be optimized. That being said, it should be noted that the construction of the symbolic matrix will remain a computationally demanding task, as for each lifting step the symbolic representations have to be handled. However, for specific parameterization, image size and decomposition depth, this matrix has to be computed only once.

The attacks themselves vary in computational demands. For testing we used Mathematica® 5.0 on an AMD Athlon® CPU at 1.66 GHz and 2 GB of RAM. All the timing results pertain to the biorthogonal lifting parameterization with one level of wavelet decomposition. The pixel sample attack is the least demanding and could be performed in 2.7 seconds. The average luminance value attack for a 64×64 pixel image took approximately 1875 seconds. The ciphertext only attack with variance as (weak) image quality indicator is the most demanding attack. We used the `NMinimize` function in Mathematica to minimize the variance of the symbolic matrix for $\alpha \in [-6, -1.2]$, the negative range of the admissible parameter values that we determined in Chapter 5, for an 8×8 pixel test image. Even for such a small image, the calculation time for the correct parameter value is 390 seconds. However, these long calculation times even for small images should not deter from the basic applicability of symbolic computation attacks, as they mainly result from the simple implementation that lacks optimization.

7.4 CONCLUSION

The goal of this work is the investigation of a novel kind of attacks on encryption schemes that use parameterized wavelet transforms to provide lightweight security. We have shown for small images that these attacks are feasible and principally present a threat to such security schemes. A ciphertext-only attack that uses an image quality indicator could potentially be very successful, however, no proper function exists so far that can produce a good indicator for image quality in all investigated parameterizations by only working on the symbolic representations of the reconstructed pixels. A full plaintext attack is very successful on any parameterization scheme as it provides ample information to deduce the parameters used for transformation. It turns out that a much lower fraction of the plaintext or limited information on the plaintext also yields expedient attacks. The success of these depends on the one hand on the complexity and number of parameters of the wavelet parameterization and on the other hand on the accuracy of the available plaintext information. Inhomogeneous and nonstationary variation of the wavelet parameters increase the number of parameters and therefore make the attack more difficult.

Although we have presented some timing results to assess computational demands of these attacks, with the simple implementation and missing optimization in our tests, the presented timing values are not very expressive. There is a lot of room for optimization and it is to be expected that these attacks can be scaled to perform well for larger images. This is the subject of further research.

On a principal note, the attacks presented here show a general problem of lightweight encryption schemes that rely on linear transforms for providing security. Even if these schemes only claim to provide lightweight security, attacks of the style presented here are a potential threat and should be taken into account.

Part III

KEY-DEPENDENT SUBBAND STRUCTURES

8 KEY-DEPENDENT ISOTROPIC WAVELET PACKETS

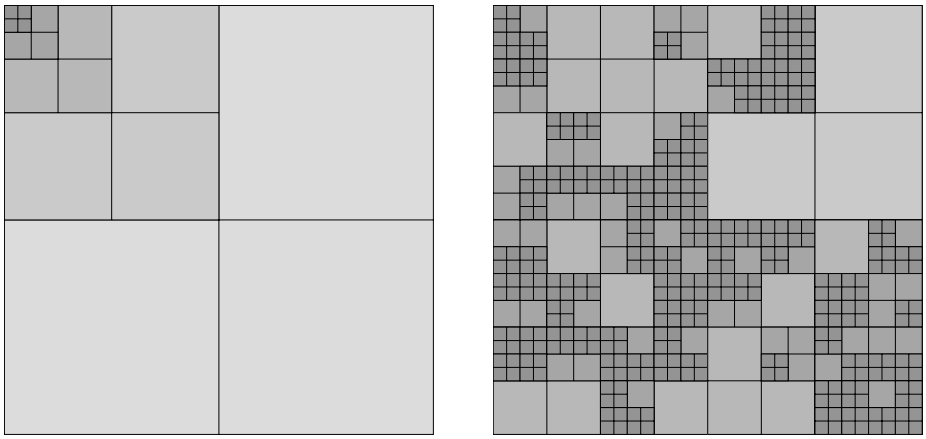
In the following chapters, we propose and analyze lightweight encryption schemes for JPEG2000 based on the wavelet packet transform. These schemes significantly reduce the amount of data to be encrypted compared to full encryption and other partial or selective encryption schemes, at the cost of increased computational complexity in the compression pipeline.

The present chapter is concerned with the isotropic wavelet packet transform. Chapter 9 deals with the anisotropic wavelet packet transform. The compression performance and security of all the proposed approaches are discussed in Chapter 10 and Chapter 11, respectively.

The proposed schemes, like the schemes based on parameterized wavelet filters presented in the last part, apply encryption integrated with compression. Like the previously discussed schemes, the proposed schemes use a secret transform domain for providing security. Other than in the previous schemes, the family of schemes we will discuss in the following does not use an arbitrary wavelet transformation, but rather creates the secret frequency domain by randomized subband decompositions, so-called wavelet packets. An advantage over the previous schemes is that the randomized subband decompositions do not form a linear connection between the used parameters and the transform coefficients and are therefore not vulnerable to the symbolic transform attack presented in Chapter 7.

8.1 WAVELET PACKETS IN JPEG2000

The wavelet packet transform (Wickerhauser, 1994) generalizes the pyramidal (or Mallat) wavelet transform. In the wavelet packet transform, apart from the approximation subband also the detail subband can be decomposed; an example is shown in Figure 8.1. This results in a larger space of possible decomposition structures (of which the pyramidal decomposition structure is a single element). For a specific level of maximum decomposition depth, there are many possible WP-structures – the WPT is an *overcomplete* library of bases. Each structure represents its own wavelet basis and therefore a specific subband decomposition. These decompositions can be adapted to the properties of the image to be transformed into account, for example by using the best



(a) Pyramidal wavelet decomposition (b) Wavelet packet decomposition

Figure 8.1: Example decomposition structures

basis algorithm (Wickerhauser, 1994) or by creating decompositions that are more optimal from a rate-distortion point of view (Ramchandran and Vetterli, 1993; Rajpoot et al., 2001). Schell and Uhl (2002a,b, 2003) present alternative methods for finding suitable wavelet packet decompositions. The wavelet packet transform has been successfully used for compressing natural images with oscillatory patterns (Marpe et al., 1998; Xiong et al., 1998; Meyer et al., 2000), fingerprint images (Hopper, 1994) and Brodatz and fabric textures with oscillatory patterns (Engel and Uhl, 2002, 2003).

A tree representation is well suited to describe a single decomposition. The *decomposition tree* is a quadtree whose nodes represent the binary decisions whether the subband associated with this node is decomposed further or not. A possible way to represent the decomposition tree is as a sequence of bits, each of which stands for a decomposition decision.

In coding schemes that are based on the zerotree hypothesis, the addition of wavelet packets demands significant considerations. These range from the principal question, whether the zerotree hypothesis still holds in the wavelet packet case to the redefinition of the parent-child relationship between coefficients in fine scale detail subbands and coefficients of coarser scale in detail or approximation subbands. For a detailed discussion of these issues along with a possible solution see Kutil (2002).

The addition of wavelet packets to JPEG2000 is more straightforward, because JPEG2000 does not rely on the zerotree hypothesis for efficient coding. JPEG2000, Part 2, allows arbitrary decomposition structures. In order to maximize keyspace size for the proposed encryption scheme, we implemented full support for arbitrary isotropic wavelet decomposition structures in JPEG2000, based on the JJ2000 reference

implementation.In the more recent versions, Kakadu is also able to provide this functionality.

Part 2 of the JPEG2000 standard (ISO/IEC 15444-2, 2004) defines new possible orientations for each subband. Apart from LL, LH, HL and HH, in Part 2 there are also LX, XL, HX, XH. The two letters refer to horizontal and vertical filtering (and decimation). Depending on its position, the letter X denotes no further processing in horizontal or vertical direction. Applying filtering and decimation in only a single direction leads to anisotropic wavelet packets. For the time being we will restrict our discussion to isotropic wavelet packets, where horizontal and vertical filtering are both applied to a subband, if the subband is decomposed. Key-dependent anisotropic wavelet packet structures are discussed in Chapter 9.

The notion of resolutions in JPEG2000 remains unchanged in Part 2, and is only determined by the all low-pass decomposition branch. However, whereas in part 1 of the JPEG2000 standard, this branch can only contain LL-subband, in Part 2, LX or XL are possible: "Since spatial resolutions are not produced with highpass processing and no two spatial resolutions can be the same, there are three possible orientations for each resolution: LL, LX, or HX" (ISO/IEC 15444-2, 2004, p. 92).

8.2 RELATED WORK

We have already mentioned proposals related to key-dependent transforms in Chapter 1. Key-dependent wavelet packets for the use of lightweight encryption were first proposed by Pommer and Uhl (2002, 2003).

Other areas where key-dependent wavelet packet decompositions have been used are watermarking und image hashing. As in the case of key-dependent wavelet filters, the main motivation for introducing key-dependent wavelet packets is increasing the security of existing schemes. Dietl and Uhl (2003, 2004) and Brachtl et al. (2004) use key-dependent wavelet packet decompositions for increasing the security of watermarking schemes. Laimer and Uhl (2007) use our version of JJ2000 with isotropic wavelet packets for a secure image authentication scheme. They generate key-dependent wavelet packet decomposition structures for transformation of the source data and create the hash in the secret transform domain.

8.3 SELECTIVE ENCRYPTION WITH RANDOMIZED WAVELET PACKETS

Pommer and Uhl (2002, 2003) propose the use of wavelet packets for providing confidentiality in a zerotree-based wavelet framework. They propose an algorithm for the randomized generation of wavelet packet structures. The idea of the encryption method is to use a key-dependent, i.e., randomly generated subband structure for en-

coding of the source data. The descriptor of the subband structure is then kept secret. Ideally, without the subband structure no reconstruction is possible. If this is the case depends on the used codec. Pommer and Uhl (2003) evaluate their scheme in a zerotree-based wavelet codec.

We transfer the idea and the central algorithm to JPEG2000 and evaluate its utility in settings that require full confidentiality, but also settings where transparent encryption is preferable.. The entirely different nature of JPEG2000 as compared to the codec used by Pommer and Uhl (2003) leads to a novel situation, especially for potential attacks. In the following, the compression performance of randomized wavelet packets in JPEG2000 is assessed, and the parameters used for the generation of randomized wavelet packets are adapted accordingly. Furthermore, with a slight modification to the original algorithm, transparent encryption can be supported naturally. In contrast to the codec used by Pommer and Uhl (2003), JPEG2000 is not based on zerotrees. Furthermore the independent coding of codeblocks in JPEG2000 leads to a lot of meta information in the bitstream, from which information on the wavelet packet subband structure could be leaked. The impact this has on possible attacks marks a key difference to the previous work.

8.4 RANDOMIZED GENERATION OF ISOTROPIC WAVELET PACKET STRUCTURES

Wavelet packet decomposition structures can be represented as a sequence of binary decomposition decisions. A naive approach for generating randomized wavelet packet structures could just assign a decomposition probability of 0.5 to each subband. However, if during generation of randomized wavelet packet decompositions, the probability for decomposition is the same at each decision, i.e., for each subband, then shallow wavelet decompositions are more probable than deep ones. This bias potentially restricts the keyspace in the context of the proposed encryption scheme.

In terms of security the best way to determine a basis for encoding is to give the same probability to each wavelet packet basis, i.e., to uniformly select from the set of all basis. If each basis is equally likely to be chosen a potential attacker can gain no advantage from knowing the distribution of the bases used for encoding.

A possible problem with the uniform distribution could be that not all bases are well suited to be used for compression. Therefore some of the basis may lead to inferior compression performance. The uniform distribution however is good in terms of comparison, as for security, the uniform distribution forms an upper bound, e.g., for the distance between two randomly chosen decomposition structure.

The second distribution which we investigate aims at discarding the bases which are not suitable for compression. Of course the remaining bases should still form a large keyspace.

8.4.1 *Uniform Distribution*

A maximum degree of security could be achieved by using a uniform probability distribution to choose from the set of possible bases (that have a maximum decomposition depth of g). In the isotropic case a uniform distribution for selecting a basis can easily be achieved: At each node in the decomposition tree a decision is made if this node should be further decomposed. Let l be the level of decomposition for the node. Then the probability $p(l)$ for decomposition for this node is given by

$$p(l) = 1 - \frac{1}{Q_{g-l}} \tag{8.1}$$

where g is the maximum overall decomposition depth and Q_j is the number of possible bases with a decomposition depth up to j. Q_j can easily be determined, e.g., using the recursive formula proposed by Xu and Do (2003):

$$Q_j = Q_{j-1}^4 + 1 \tag{8.2}$$

where $Q_0 = 1$. One possible basis comes from the case where the node is not further decomposed. If the node is further decomposed then the number of possible bases is given by the combination of possible decompositions in the subtree for each subband.

8.4.2 *Compression-oriented Distribution*

In order to limit the selection process to bases that produce acceptable compression results, three parameters are introduced: the maximum global decomposition depth for all subbands *(g)*, the maximum *(m)* and minimum *(n)* decomposition depth for the approximation subband.

In order to make it possible to influence the selection process, two parameters are introduced that allow to favor the selection of deeper or more shallow decompositions: the *base value (bv)* and the *change factor (cf)*. They can be used to influence the probability of decomposition at a single decision point, based on a base probability and a factor that grows or shrinks with the current decomposition depth: The base value *bv* determines the basic probability with which a subband is decomposed. The change factor *cf* alters this probability based on the decomposition depth of the subband: for each level of decomposition, the change factor is added to the base value. If

```
function decomposition_decision:
 if (subband = approximation_subband) then
   if (curr_depth < min_approx_depth) then
     decompose
   else if (curr_depth>max_approx_depth) then
     do not decompose
   else
     inner_decomposition_decision
 else          (not approximation subband)
   if (curr_depth > overall_maximum) then
     do not decompose
   else
     inner_decomposition_decision

function inner_decomposition_decision:
 x = PRNG([0,2[)
 weight = base_value +
          curr_depth * change_factor
 if (x / weight >= 1) then
   decompose
```

Listing 8.1: Random Generation of Wavelet Packets

cf is positive, then the higher the level of the subband, the higher is the chance for it to be decomposed. If *cf* is negative, the chance for decomposition decreases with higher decomposition levels. In this way, the generation process can be tuned to favor deeper or more shallow decompositions. Generally, it is advisable to tune these parameters to produce a balanced distribution.

Another parameter is the seed *s* for the pseudo-random number generator (PRNG). The algorithm by Pommer and Uhl (2003) is given in Listing 8.1.

To achieve *transparent encryption*, we introduce an additional parameter *p* that can be used to optionally specify the number of higher pyramidal resolution levels. If *p* is set to a value greater than zero, the pyramidal wavelet decomposition is used for resolution levels R_0 through R_p and wavelet packets are used for the higher resolution levels, starting from R_{p+1}. With resolution-layer progressions in the final bitstream, standard JPEG2000 codecs can be used to obtain resolutions R_0 to R_p. Note that if higher pyramidal resolution levels are used, *n* should be set to a sufficiently large value, ideally to the same as *m*, in order to avoid wavelet packet decompositions which are very similar to the pyramidal decomposition. Note that a decoder compliant to JPEG2000,

g	Maximum global decomposition depth of all subbands
n	Minimum decomposition depth of the approximation subband
m	Maximum decomposition depth of the approximation subband
p	Level of transparency
bv	Base value of decomposition probability
cf	Change factor of decomposition probability
s	Seed for pseudo-random number generator

Table 8.1: Parameters for generating randomized isotropic WP Bases

part 1 is sufficient to decode the preview image. All possible parameters for generating randomized wavelet packet structures are given in Table 8.1.

Keeping the decomposition structure secret can be achieved in two ways: either the decomposition tree itself is used as the key and not included in the bitstream, or header information containing the used wavelet packet decomposition structure is encrypted with a traditional cipher. If a PRNG is used for generating the wavelet packet structure, the relevant information consists only of the seed and the parameters which are discussed below, otherwise the information consists of the complete wavelet packet structure. In either way, the amount of information that has to be encrypted is very small.

8.5 COMPLEXITY

Wavelet packets bring an increase in complexity as compared to the pyramidal wavelet decomposition, as can be seen in Table 8.2, which has been taken from Pommer and Uhl (2003). The order of complexity for a level l full wavelet packet decomposition of an image of size N^2 is $\mathcal{O}(\sum_{i=1}^{l} 2^{2(i-1)} \frac{N^2}{2^{2(i-1)}})$ compared to $\mathcal{O}(\sum_{i=1}^{l} \frac{N^2}{2^{2(i-1)}})$ for the pyramidal decomposition, with the randomized wavelet packet decompositions ranging in-between. For encryption the generation of the randomized wavelet packet subband structure also has to be taken into account (although compared to the computational demands of the encoding pipeline the complexity introduced by random generation will usually be negligible).

The enhanced complexity has to be taken into account for potential application scenarios. The effort for encryption is dramatically reduced compared to full encryption

and other partial or selective encryption schemes, but the wavelet packet transform introduces computational demands in the compression pipeline.

image size (length of one side)	N
1 line	$\mathcal{O}(N)$
1 image (=total)	$\mathcal{O}(N^2)$
level i decomposition, 1 line	$\mathcal{O}(\frac{N}{2^i})$
level i decomposition, total	$\mathcal{O}(\frac{N}{2^i} \cdot \frac{N}{2^i}) = \mathcal{O}(\frac{N^2}{2^{2i}})$
pyramidal wavelet decomposition, level l	$\mathcal{O}(\sum_{i=1}^{l} \frac{N^2}{2^{2(i-1)}})$
full wavelet packet decomposition, level l	$\mathcal{O}(\sum_{i=1}^{l} 2^{2(i-1)} \frac{N^2}{2^{2(i-1)}})$
intermediate scenario, level l	$\mathcal{O}(\sum_{i=1}^{l} 2^{i-1} \frac{N^2}{2^{2(i-1)}})$

Table 8.2: Order of complexity of wavelet packet transform

8.6 CONCLUSION

We have shown that wavelet packets can help to reduce computational demands for lightweight encryption. The reduction of complexity for encryption comes at the cost of a rise in complexity in the compression pipeline. The actual applicability of the presented approach depends on the scenario in which it is to be used. In the next chapters we discuss how anisotropic wavelet packets can be used to enhance the keyspace size for key-dependent wavelet packet structures. In Chapter 11 we will evaluate the security for both approaches.

9 KEY-DEPENDENT ANISOTROPIC WAVELET PACKETS

In this chapter (cf. Engel and Uhl, 2006b, 2007a; Kutil and Engel, 2008), we investigate how the anisotropic wavelet packet transform can enhance the lightweight encryption technique for JPEG2000 that was proposed in the last chapter. The main motivation to introduce anisotropic wavelet packets in the encryption scheme is a significant increase in keyspace size, due to the fact that the anisotropic transform has more degrees of freedom. We will discuss this point in the next chapter, when we turn to the security evaluation.

In the next chapter, we will show that with the right parameters, complexity and compression performance can be tuned to yield results that are competitive with, and in some cases even outperform, randomized compression-oriented selection of isotropic wavelet packets. In order to determine the trade-off between compression performance and keyspace size in the next chapter, we will again compare the approach to a method that selects bases from the whole set of anisotropic wavelet packet bases following a pseudo-random uniform distribution. A large part of the present chapter will be concerned with how to construct such a uniform distribution, which is not as trivial as it is in the isotropic case.

9.1 ANISOTROPIC WAVELET PACKETS

The anisotropic wavelet packet transform is a generalization of the isotropic case: whereas in the latter, horizontal and vertical wavelet decomposition are always applied in pairs for each subband to be decomposed, this restriction is lifted for anisotropic wavelet packets. An example for an anisotropic decomposition is shown in Figure 9.1.

Anisotropic wavelet packets have been proposed for the compression of image (Kutil, 2003a; Xu and Do, 2003) and video (Kutil, 2003b) data. The main motivation to introduce anisotropic wavelet packets for lightweight encryption is a substantial increase in keyspace size: the space of possible bases is not only spanned by the decision of decomposing or not (as is the case for the isotropic transform), but also by the direction of each decomposition (cf. Engel and Uhl, 2006b). The principal algorithm for encryption stays the same as in the isotropic case: During compression a random specimen of the set of admissible bases is selected for transformation and kept secret.

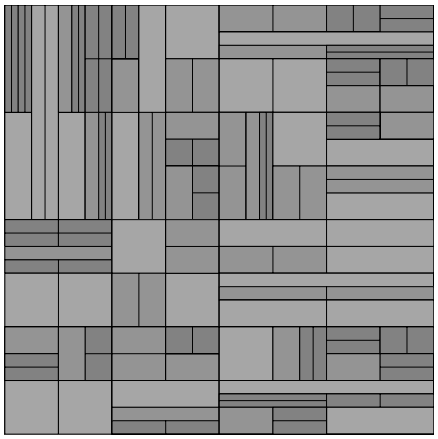


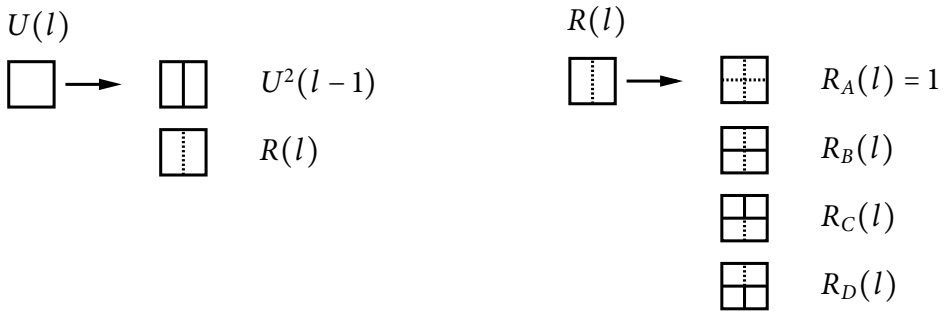
Figure 9.1: Example anisotropic wavelet packet decomposition

The description of the used basis can be used as a separate secret key or encrypted with a traditional cipher and inserted into the bitstream. Only a minimal amount of data needs to be encrypted. JPEG2000, Part 2, allows anisotropic wavelet packets. In our empirical tests we use a modified version of JJ2000 with full support for anisotropic wavelet packets.

Even more than in the case of isotropic wavelet packets, there are some anisotropic wavelet packet decomposition that are ill-suited for energy compaction. To eliminate these bases, which do not produce good compression results, we again introduce parameters that can be used to constrain the possible anisotropic decompositions. This reduces the size of the available keyspace, so the goal here is to strike a good balance between satisfactory compression performance and keyspace size.

9.2 RANDOMIZED GENERATION OF ANISOTROPIC WAVELET PACKETS

In the anisotropic case the construction of a uniform distribution is a lot more sophisticated than in the isotropic case. In order to construct a method for uniform selection, we first need to determine the number of anisotropic wavelet packets for a certain decomposition depth. In the following we only show this number for joint maximum decomposition depths for horizontal and vertical decomposition. More details can be found in Kutil and Engel (2008).



(a) Unrestricted processing (b) Restricted processing

Figure 9.2: Case selection for uniform distribution of randomized AWP bases for joint maximum horizontal and vertical decomposition depth

9.2.1 *Total Number of Anisotropic Bases*

For determining the number of anisotropic wavelet packet bases, the fact has to be taken into account that a horizontal decomposition (**r**) followed by two vertical decompositions (**c**), or a vertical decomposition followed by two horizontal decompositions, lead to an equivalent result ("isotropic decomposition"), which must be counted only once. This is illustrated by Figure 9.3, which shows all possible anisotropic decompositions up to level 2: The decompositions in the first column in line 2 and 4 are equivalent, as they both lead to an isotropic decomposition.

Xu and Do (2003) suggest a way to determine the number of possible anisotropic bases for separate maximum decomposition depth in horizontal and vertical direction. For randomized anisotropic wavelet packets, a joint maximum decomposition depth for both directions is preferable. The method by Xu and Do can be adapted for this case. We determine $A(l)$, the number of bases of joint horizontal and vertical decomposition level up to l, recursively. The root node may not be decomposed, or it may be decomposed either horizontally or vertically, forming two subtrees of $A(j-1)$ possible decompositions in each case, leading to $1 + 2A^2(l-1)$ possible bases.

As mentioned above, there are some decompositions paths that result in the same basis: a horizontal decomposition (**r**) followed by two vertical decompositions (**c**) on the resulting subtrees is equivalent to the case in which the vertical decomposition is done first followed by two horizontal decompositions, as illustrated in Fig. 9.4. There are $2A^4(l-2)$ isotropic decompositions that are counted twice in this formula. Therefore, half their number, $A^4(l-2)$, is subtracted, leading to the formula:

$$A(l) = 1 + 2 \cdot A^2(l-1) - A^4(l-2) \tag{9.1}$$

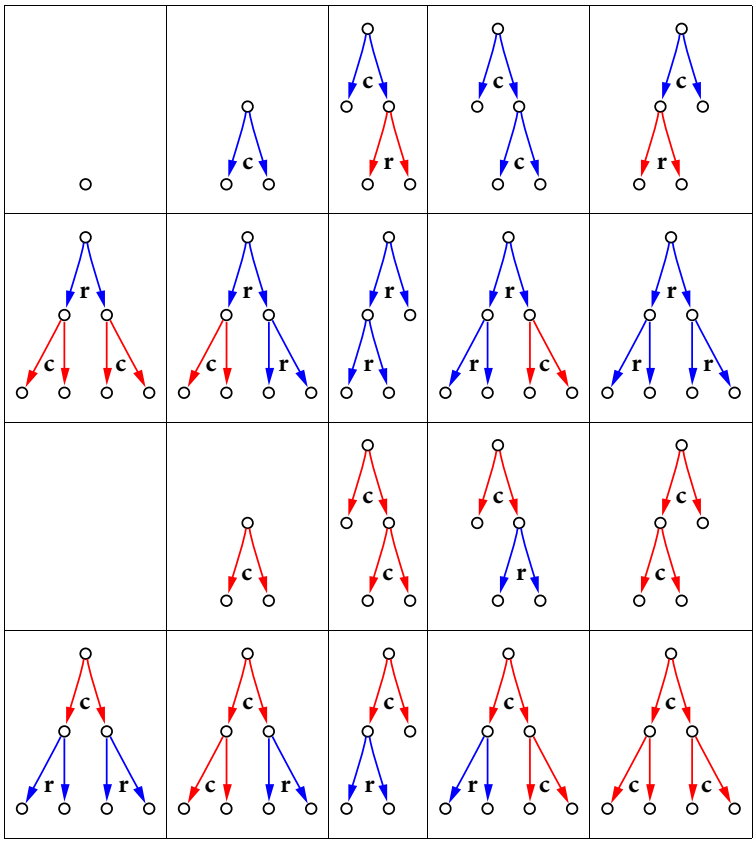


Figure 9.3: All anisotropic wavelet packet decompositions up to level 2

where $A(0) = 1$, $A(1) = 3$, and $A(l) = 0$ for $l < 0$.

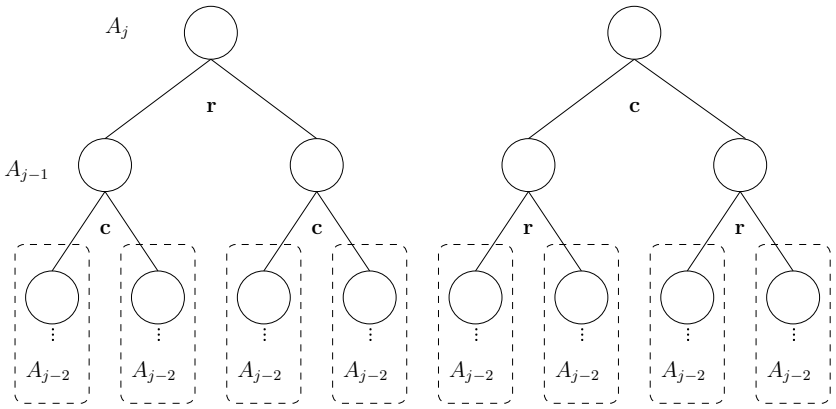


Figure 9.4: Number of equivalent AWP bases

In preparation for defining a uniform distribution later, we use a different approach to determine the number of bases: The decomposition decisions are split into mutually exclusive cases (Figure 9.2), then the number of anisotropic bases is determined recursively by the number of bases contained in the subtree for each case. To account for the fact that there is more than one way to construct an isotropic decomposition, two modes of decomposition are distinguished: *unrestricted* and *restricted*. Without loss of generality, we define the admissible decompositions for a subband without restriction as horizontal decomposition with further unrestricted processing (U), or further processing with restriction for horizontal decomposition (R). Case (R) leads to the restricted case in which horizontal decomposition is forbidden for at least one of the resulting subbands, leading to four possible decompositions (R_A through R_D). Let $U(l)$ be the number of bases up to level l in the unrestricted case and $R(l)$ be the number of bases up to level l in the restricted case, with the number of possible subcases $R_A(l), R_B(l), R_C(l)$, and $R_D(l)$. The possible decompositions in unrestricted and restricted mode are illustrated by Figure 9.2. We recursively determine the number of bases for each possible decomposition in the following way:

$$U(l) = \begin{cases} R(0) & \text{for } l = 0 \\ U^2(l-1) + R(l) & \text{else} \end{cases} \tag{9.2}$$

$$R(l) = R_A(l) + R_B(l) + R_C(l) + R_D(l) \tag{9.3}$$

$$R_A(l) = 1 \tag{9.4}$$

$$R_B(l) = \begin{cases} 0 & \text{for } l = 0 \\ R^2(l-1) & \text{else} \end{cases} \tag{9.5}$$

$$R_C(l) = R_D(l) = \begin{cases} 0 & \text{for } l = 0 \vee l = 1 \\ U^2(l-2) \cdot R(l-1) & \text{else.} \end{cases} \tag{9.6}$$

This formula is more flexible than Equation 9.1 and it forms the foundation for the uniform distribution as well as for determining the number of bases in the compression-oriented approach in Chapter 11.

9.2.2 *Uniform Distribution*

Using a uniform random distribution over the whole set of bases is one way to choose a basis. In this procedure, every decomposition structure up to a maximum decomposition level is equally likely to be chosen for the transformation step. From the perspective of security, a uniform distribution is a good choice, as it does not give a potential attacker any prior knowledge.

We use the case distinction which we introduced above to construct a uniform distribution for the selection of a random basis: the probability for any case to be chosen is the ratio of the number of bases contained in the case to the total number of bases ($U(l)$ for unrestricted processing, $R(l)$ for restricted processing).

9.2.3 *Compression-oriented Distribution*

The parameters for compression-oriented selection of anisotropic wavelet packet bases are given in Table 9.1. Note that these parameters differ from the isotropic case in order to reflect the properties of the anisotropic wavelet packet transform. For our experiments we also investigate explicit minimum and maximum decomposition depth of the detail subbands (parameters d and e). In the isotropic case we did not specify a minimum decomposition depth.

The first four parameters, n, m, e, d, determine the maximum and minimum decomposition depths for the approximation subband and the detail subbands. They influence both, compression performance and keyspace size. Note that the number of decompositions is given here as single decompositions in any direction, whereas for the isotropic wavelet packet transform the number of decompositions usually denotes pairs of horizontal and vertical decompositions. Therefore, a decomposition depth of $2k$ in the anisotropic case is comparable to a decomposition depth of k in the isotropic case.

Constraining the absolute degree of anisotropy for the approximation and detail subbands by setting q and r, respectively, may be necessary to prevent subbands from

n	Minimum decomposition depth of the approximation subband
m	Maximum decomposition depth of the approximation subband
e	Minimum decomposition depth of the detail subbands
d	Maximum decomposition depth of the detail subbands
q	Maximum absolute degree of anisotropy for approximation subband
r	Maximum absolute degree of anisotropy for detail subbands
p	Level of transparency
bv	Base value of decomposition probability
cf	Change factor of decomposition probability
s	Seed for pseudo-random number generator

Table 9.1: Parameters for generating randomized AWP bases

being decomposed excessively in a single direction, as, especially in the case of the approximation subband, this would lead to inferior energy compaction in the frequency domain for the other direction. For the degree of anisotropy Υ of a subband we use the following definition:

$$\Upsilon(h, v) = v - h \tag{9.7}$$

where h and v are the decomposition depths in horizontal and vertical direction, respectively. Note that q and r pertain to the absolute degree of anisotropy, i.e., $|\Upsilon(h, v)|$. Also note that in previous work (Engel and Uhl, 2007a) we used a different measure for the degree anisotropy, reflecting the "squareness", i.e., the degree of isotropy rather than the degree of anisotropy. The "squareness" $S(h, v)$ (with h and v as previously defined) of a subband as an inverse measure of anisotropy is defined as:

$$S(h, v) = \frac{1}{2^{|h-v|}} \tag{9.8}$$

The definition Υ of the degree of anisotropy that we use here is also used in Kutil and Engel (2008), where many further examinations are given, e.g., the expected degree of anisotropy for different distributions.

If at any node during the randomized generation of a anisotropic wavelet packet basis, decomposition of the subband at this node in the randomly chosen direction would result in the degree of anisotropy exceeding the maximum degree of anisotropy, the direction of the decomposition is changed. The degree of anisotropy for the approx-

imation and detail subbands influence both, compression performance and keyspace size.

This is not the case for the following three parameters, which only determine the probability distribution of the randomly generated bases. They behave the same way as in the isotropic case: The seed s initializes the pseudo-random number generator. The base value bv set the basic probability of decomposition and the change factor cf alters this base probability depending on the current decomposition level.

Transparent encryption can be accommodated in the proposed scheme by introducing a parameter p that reflects the number of resolutions that can be decoded without knowledge of the anisotropic decomposition structure. For this purpose, $2r$ decompositions, alternating between horizontal and vertical direction, are applied recursively to the LL-Subband, where r is the total number of resolutions. Of the $2r$ detail subbands generated in this way, only the first $2r-2p$ are subject to further decomposition. The resulting LL-subband, and the corresponding detail subbands for the resolutions R_0 to R_{p-1} are the same as that produced by the pyramidal wavelet transform. Any decoder compliant to JPEG2000, part 1 can be used to decode the first p resolutions.

9.3 COMPLEXITY

The complexity of the anisotropic wavelet packet transform is the same as for the isotropic wavelet packet transform. As the number of anisotropic wavelet packets is higher than the number of isotropic wavelet packets by an order of magnitude, for the application of encryption, the overall number of decompositions necessary to obtain a large keyspace of possible bases will be significantly lower in the anisotropic case compared to the isotropic case. Therefore, in the context of the lightweight encryption scenario proposed here, the complexity of anisotropic wavelet packets is below that of isotropic wavelet packets (for obtaining the same keyspace size).

9.4 CONCLUSION

The proposed scheme increases keyspace size as compared to the approach suggested in the previous chapter, which only uses isotropic wavelet bases. As will be shown in Chapter 10, the compression-oriented selection method for randomized anisotropic wavelet packets also produces better compression results than the isotropic wavelet packet transform.

As the anisotropic wavelet transform and the isotropic wavelet transform are of the same computational complexity, no processing overhead is introduced. In other words, compared to isotropic wavelet packets, the same keyspace size can be achieved

with anisotropic wavelet packets at a significantly lower cost of computational complexity.

As a side product of our evaluation, we have given an approach to generate uniformly distributed anisotropic wavelet packet decompositions. This could be useful for the investigation of their compression performance for different classes of images. A thorough discussion of methods for the anisotropic wavelet packets, not only related to security, can be found in Kutil and Engel (2008).

10 COMPRESSION PERFORMANCE

For the proposed encryption schemes to be applicable in practical scenario, the compression performance with wavelet packets in JPEG2000 has to be comparable to the results obtained with the pyramidal decomposition. In this chapter we start our investigation of compression performance by comparing the compression performance of our implementation to D. Taubman's Kakadu codec. For this purpose we use the well-known WSQ decomposition structure. We then turn to the compression performance of randomly selected wavelet packet bases. We take a small number of test images to motivate parameter settings for the compression-oriented selection of both, isotropic and anisotropic wavelet packets. We compare the achieved compression results to the pyramidal decomposition and to the compression results obtained by the uniform selection method.

In the second part of this chapter, we report results for a larger set of test images. For this set of images we compare the compression-performance at different rates of the pyramidal decomposition, isotropic and anisotropic wavelet packets, each with compression-integrated and uniform distribution for basis selection.

10.1 COMPARISON TO KAKADU

As wavelet packets have successfully been used to compress fingerprint data, in order to test our implementation of wavelet packets in JJ2000, we use the four databases of the Fingerprint Verification Contest 2004 (which we will use again in Part IV of this thesis) and compressed them with the WSQ wavelet packet decomposition structure (Bradley et al., 1993). We compare the compression results with the Kakadu codec (v. 6.0).[1] The results are shown in Figures 10.1 to 10.4 as the average compression results for each database of fingerprint images (each database contains 800 images, examples are shown in Figure 13.1 on page 176). It can be seen that both, the extended JJ2000 codec and the Kakadu codec, achieve similar compression results. (It can also be seen,

[1] The following commandline parameter was used to provide the WSQ structure in Kakadu for the comparison: `Cdecomp="B(B:B:B),B(BBBBB:BBBBB:-),B(B:B:B),B(-:-:-),B(-:-:-)"`.

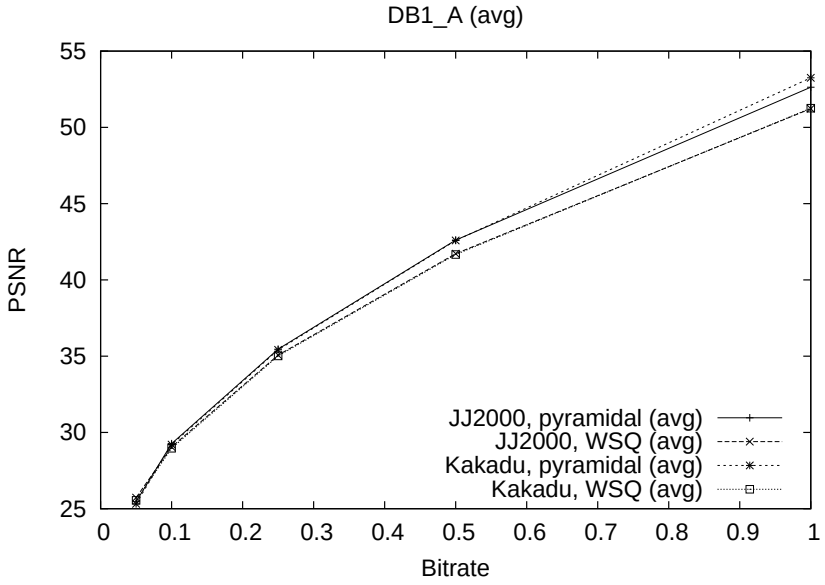


Figure 10.1: Comparison of compression performance of fingerprint images (DB1) between JJ2000 (with wavelet packets) and Kakadu

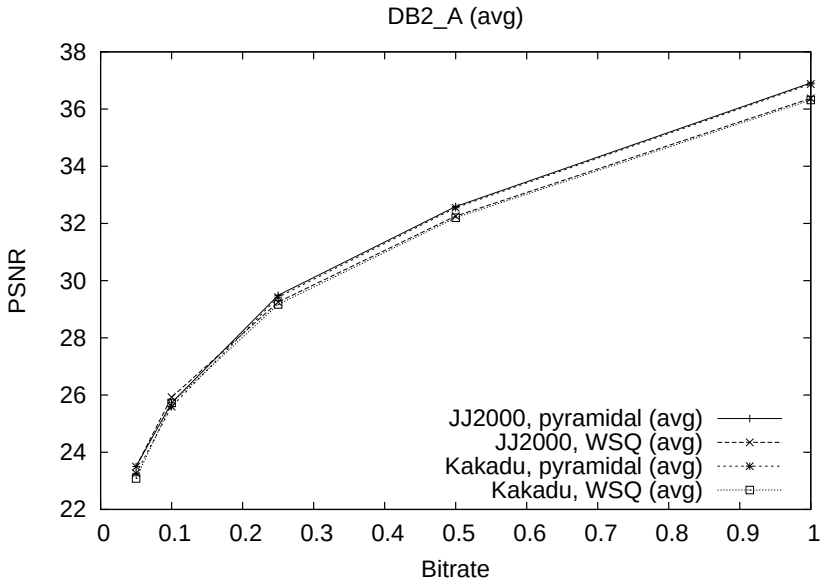


Figure 10.2: Comparison of compression performance of fingerprint images (DB2) between JJ2000 (with wavelet packets) and Kakadu

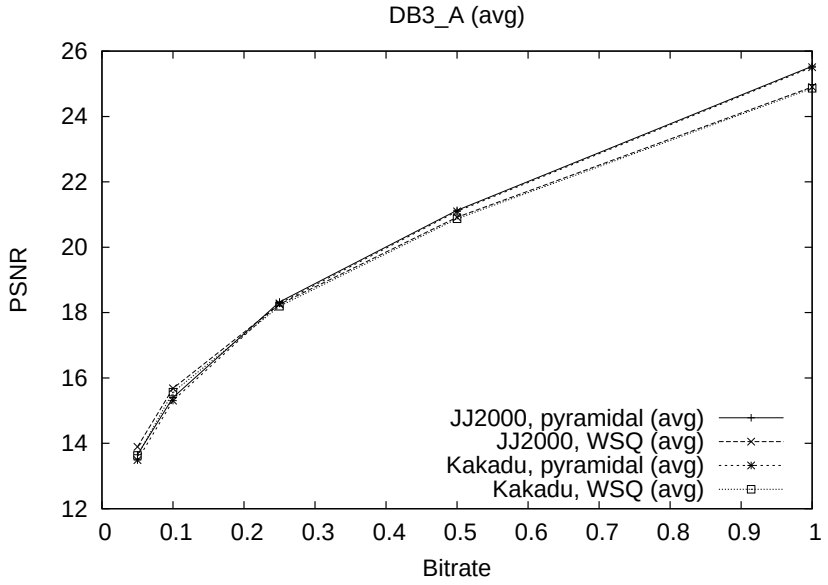


Figure 10.3: Comparison of compression performance of fingerprint images (DB3) between JJ2000 (with wavelet packets) and Kakadu

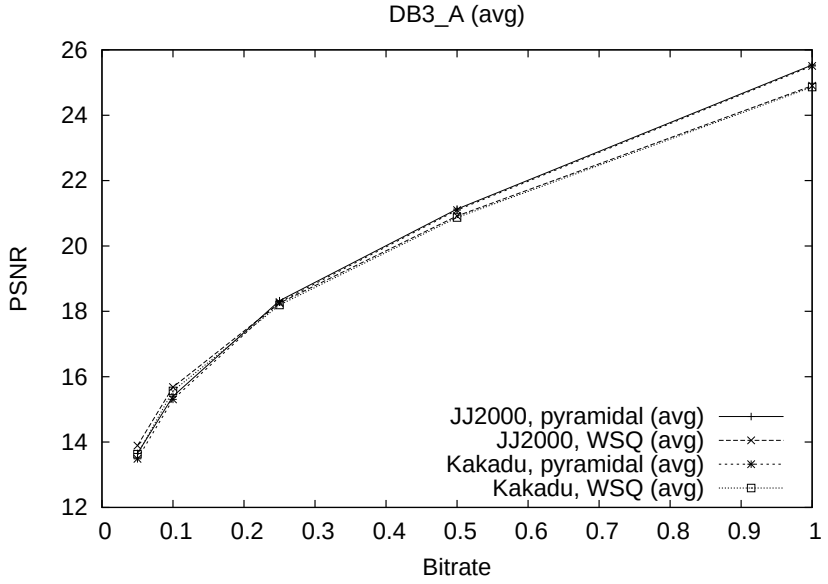
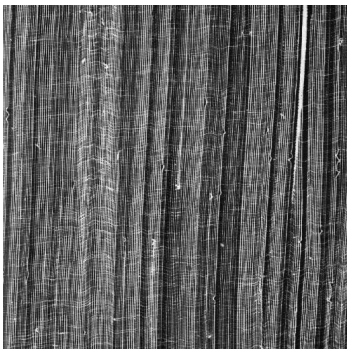


Figure 10.4: Comparison of compression performance of fingerprint images (DB4) between JJ2000 (with wavelet packets) and Kakadu

(a) Lena

(b) Barbara



(c) D105

Figure 10.5: Test images

that in JPEG2000 the WSQ decomposition structure does not result in superior compression performance.)

We will return to the topic of fingerprint images in the last part of the thesis. For now, we only used fingerprint images to compare two different implementations.

10.2 PARAMETERS FOR COMPRESSION-ORIENTED SELECTION

In this section we motivate reasonable parameter settings for the compression-oriented selection. We use three test images, which are shown in Figure 10.5. Apart from the standard images Lena and Barbara, we also tested D105 from the suite of Brodatz textures, to have a specimen of oscillatory patterns. The proposed settings will be verified in a larger test containing 100 images at the end of this chapter. It has to be noted that the optimal parameters will among other factors, of course strongly depend on the image size. We use 8 bpp grayscale images of size 512×512 pixels.

10.2.1 *Isotropic Wavelet Packets*

In the following, we assess test runs which leave one parameter fixed and vary the other parameters through their respective ranges (leading to a total of 68796 test runs per image for g up to 7 and 28665 test runs for g up to 5). If not noted otherwise, the compression rate is 0.25 bpp. The average, minimum and maximum peak-signal-to-noise-ratio (PSNR) for each value of the fixed parameter are plotted. Of the five parameters that influence compression performance, three also affect the number of possible decomposition trees, namely g, m, and n. We will discuss the impact of these parameters in more detail.

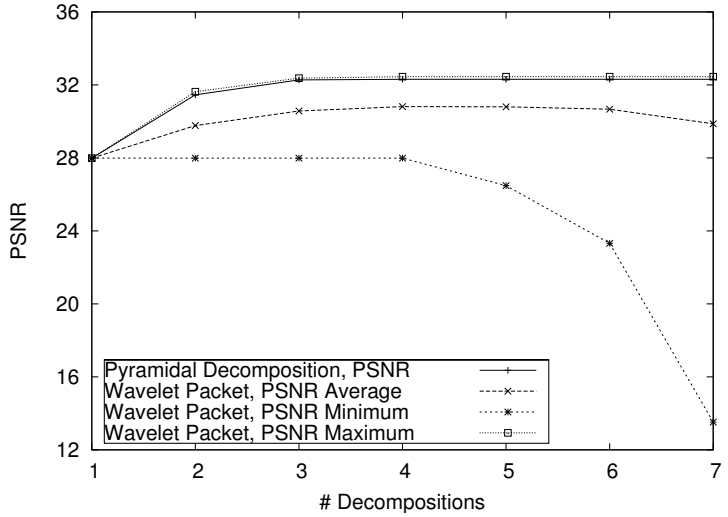
In the comparison of compression quality of the wavelet packet decomposition and the pyramidal wavelet decomposition it is noteworthy that there are wavelet packet decomposition structures that produce better results than the pyramidal structure. As can be seen in Figures 10.6(a) and 10.6(b), this effect is stronger for the image Barbara, because it contains more oscillatory patterns that favor energy compaction with wavelet packets. On average, the wavelet packet decompositions are outperformed by the pyramidal decomposition. At the end of this section, we will present parameter settings that minimize the difference in compression performance.

The other two parameters, *bv* and *cf*, if not set to extreme values, only affect the probability distribution over the possible decompositions, but not the number of potential decompositions. Figure 10.8 illustrates this for Lena: the compression results of different quality are spread over all values of *bv* and *cf*. Pommer and Uhl (2003) suggest to set *bv* to 1 and *cf* to 0. While this is sensible from an image compression point of view, it also means a bias in selection, as deeper decompositions are less probable than shallow decompositions. Because the two parameters do not have much impact on compression quality, we propose to set them to values that, based on the maximum decomposition depth, strike a good balance between decomposition depth, i.e., keyspace quality (as discussed in Section 11.2), and compression performance. Keyspace quality and security implications will be discussed in Chapter 11.

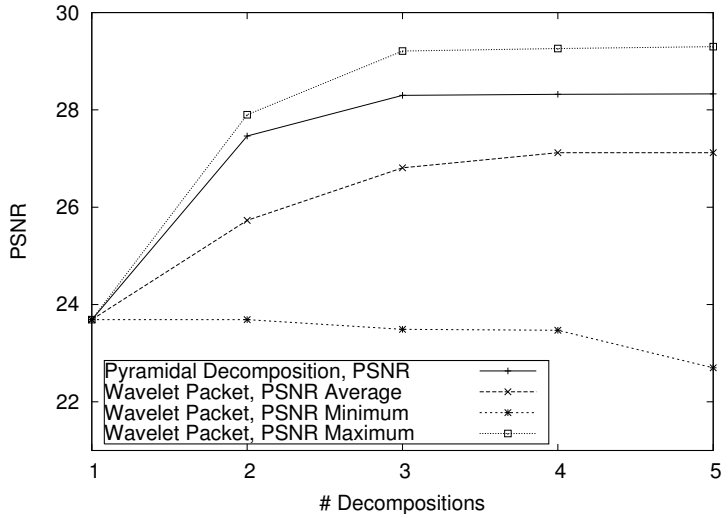
The parameter specifying the maximum global decomposition level, g, has a major effect on both compression performance and the number of potential decomposition trees. As can be seen in Figure 10.6(a), for high overall levels of decomposition, compression performance quickly degenerates. For the size of images used, this is mainly due to the fact that there is an increase in header information for complex wavelet packet decompositions, while no further gain in energy compaction is achieved. We therefore limit our observations to a maximum decomposition depth of 5, which is a reasonable value considering the size of our test images.

Figure 10.7 shows that the minimum decomposition depth of the approximation subband, n, if set too low, has a significant impact on compression performance. Settings of one or two levels produce compression results that are not competitive and should be avoided. Discarding these smaller settings only marginally affects the number of possible wavelet packet decompositions. With regard to maximum decomposition depth of the approximation subband, m, our results show that there is no reason to restrict this parameter. Considering keyspace size as well as compression performance, we propose to set m to the same value as g.

Taking the above observations and the size of the used test-images into account, we can suggest settings that remove non-competitive compression results for the used test images in the isotropic case: $g = 5$, $n = 3$, and $m = 5$.

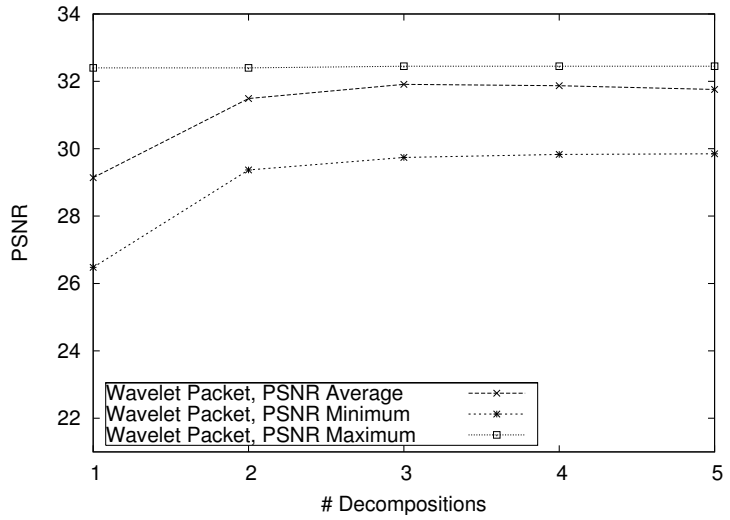


(a) Maximum level of overall decomposition depth (Lena)

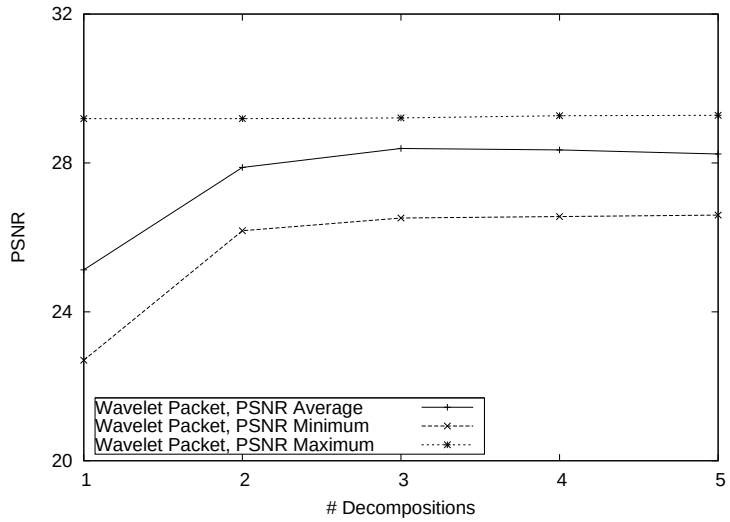


(b) Maximum level of overall decomposition depth (Barbara)

Figure 10.6: Compression performance of randomized wavelet packet decompositions by maximum global decomposition depth

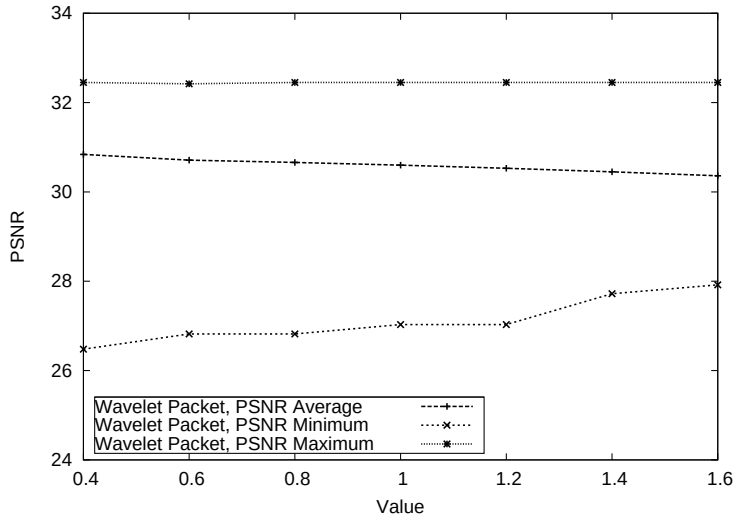


(a) Minimum level of approximation subband decomposition depth (Lena)

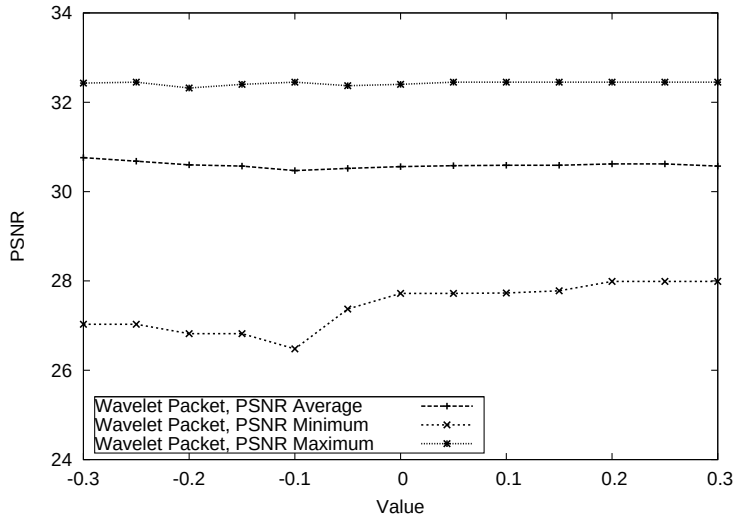


(b) Minimum level of approximation subband decomposition depth (Barbara)

Figure 10.7: Compression performance of randomized wavelet packet decompositions by minimum decomposition depth of approximation subband



(a) Base value



(b) Change factor

Figure 10.8: The impact of change factor and base value (Lena)

10.2.2 *Anisotropic Wavelet Packets*

As for the isotropic case in the previous section, we start off by motivating a choice of parameters for a specific and constrained set of the three standard test image shown in Figure 10.5. Then we verify these results for a large set of test images and also compare them to the compression results obtained by the uniform distribution. Again it has to be stressed that the perfect choice of parameters will depend on a number of factors, like image size and, to some degree, frequency properties of the image content.

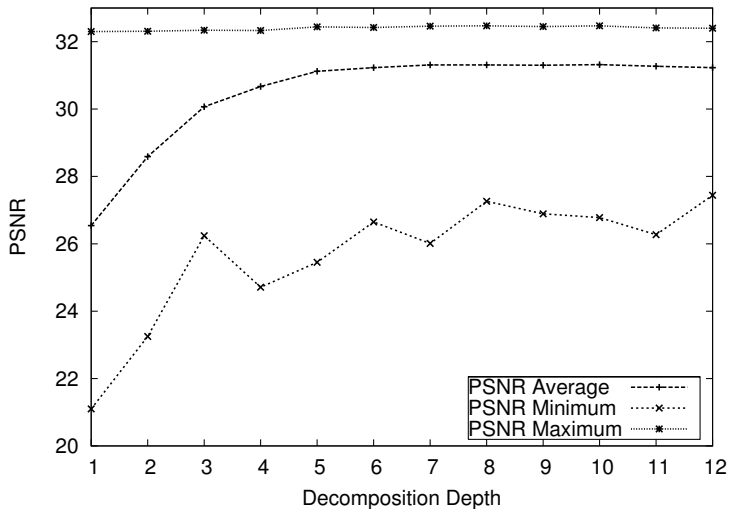
In order to motivate sensible parameter settings that favor good compression results we again show the impact of the respective parameters on the compression quality for the test images. The plots shown for illustration are for the image Lena, 512×512 pixels at a compression ratio of 0.25 bpp, and present the minimum, average, and maximum PSNR of the randomly generated decompositions.

Figure 10.9(a) shows the compression performance by minimum decomposition depth of the approximation subband. As can be seen, setting n is important, as only a sufficient number of decompositions ensures competitive compression results. For the image size used in our tests, setting the maximum decomposition depth m to 12 produced favorable compression results.

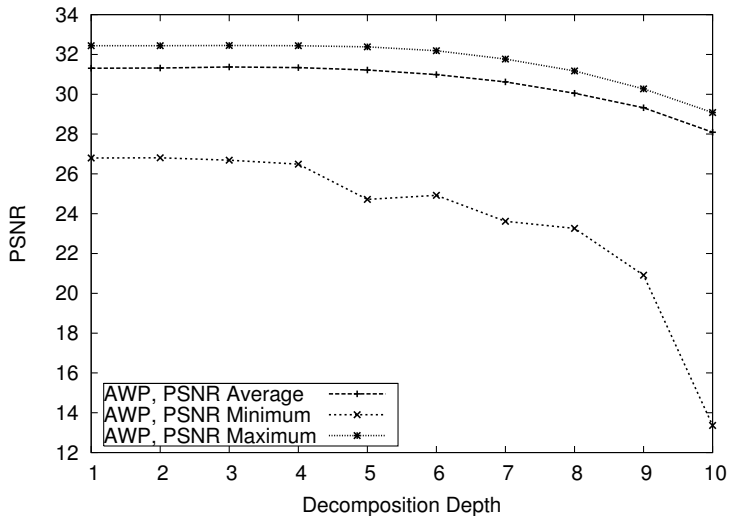
For most natural images, the situation for the detail subbands is different, as illustrated by Figure 10.9(b). A minimum decomposition depth does not improve compression results in this case. If the decomposition depth of the detail subbands is set too high, however, significant overhead is introduced by the large number of subbands that leads to a deterioration of compression performance, so the maximum decomposition depth of the detail subbands, d, should be confined. On the other hand, d must not be set much lower than m, as this affects security for the lower resolutions. In our tests, $d = 8$ produced acceptable results, but, as shown by the results for a larger test set below, this is a conservative setting, and a higher value also produces acceptable compression results.

Figure 10.10 illustrates the impact of different settings of the maximum degree of anisotropy for the approximation subband and the detail subbands. The other parameters are set to $n = 6, m = 12, e = 0, d = 8$. It can be seen in Figure 10.10(a)[2] that a low degree of anisotropy, i.e., a high similarity to the isotropic decomposition, increases minimum and average compression performance. We therefore propose to set q to 1 (which is the minimum). For the detail subbands, it can be seen in that the situation is different: the maximum degree of anisotropy (r) cannot improve minimum compression results. This is illustrated by Figure 10.10(b): The curves for average, minimum and maximum are nearly parallel to the abscissa. This means that the inferior

[2] Note that the degree of anisotropy is here given as the "squareness" of a subband, as defined in Equation (9.8) on page 119.

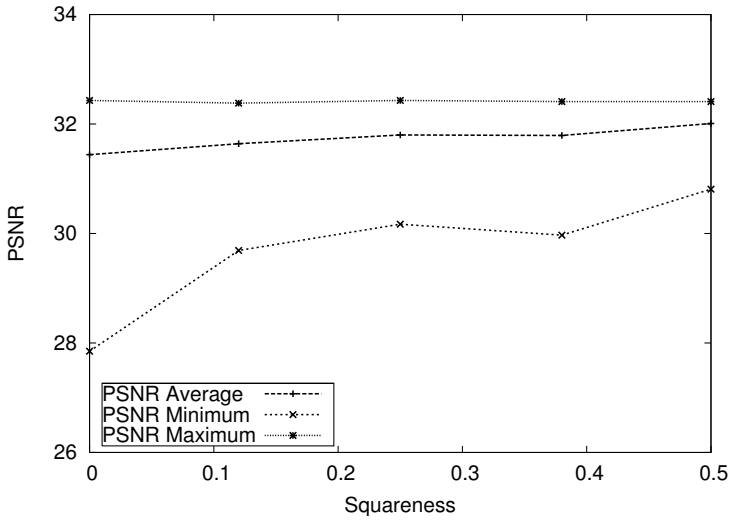


(a) Minimum decomposition depth for approximation subband

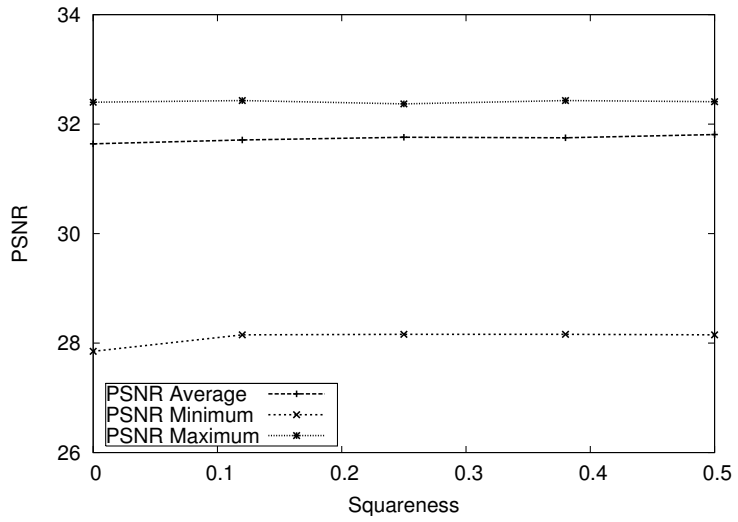


(b) Minimum decomposition depth for detail subbands

Figure 10.9: Influence of minimum decomposition depth on compression performance



(a) Maximum degree of anisotropy for approximation subband



(b) Maximum degree of anisotropy for detail subbands

Figure 10.10: Influence of maximum degree of anisotropy on compression performance

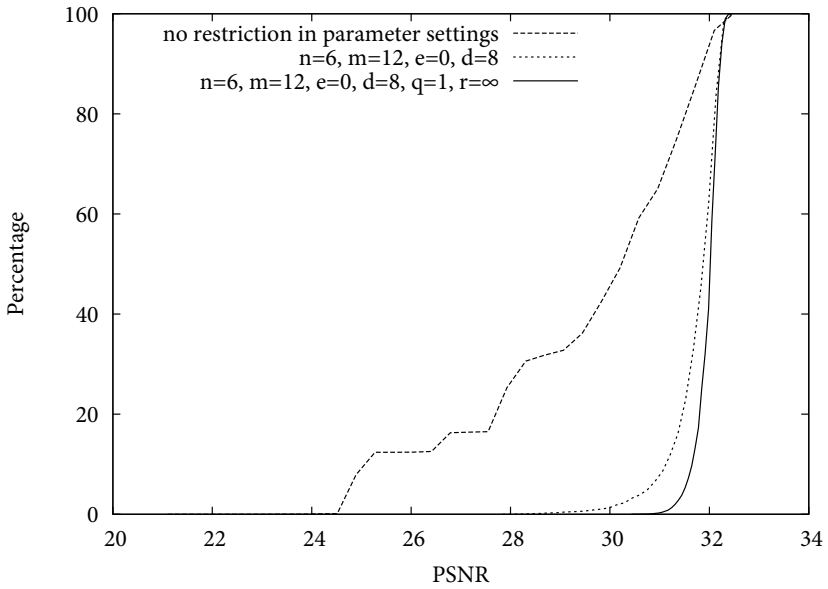


Figure 10.11: Impact of restricting maximum degree of anisotropy for approximation subband

compression results are evenly distributed over all settings for the maximum degree of anisotropy of the detail subbands. This situation obviously yields an advantage in terms of keyspace size.

Figure 10.11 compares the parameter settings by plotting on the ordinate the percentage of samples for which the compression quality lies below the PSNR value of the abscissa. It can be seen that setting the restricting the maximum degree of anisotropy for the approximation subband puts the finishing touch on compression performance.

10.2.3 *Comparison of Isotropic and Anisotropic Compression-oriented Selection*

Tables 10.1, 10.2, and 10.3 show the compression performance for the three test images. It compares the compression-oriented approaches for anisotropic and isotropic wavelet packets with the parameters suggested above at different rates. For each image at each rate and each wavelet packet types 1000 basis were selected. The parameters for randomized selection are the same ones we also use in the larger empirical study with 100 images, and are given in Table 10.4. Note that we diverge a little from the parameters given in the last sections for the anisotropic case and use a maximum decomposition depth of 10 for the anisotropic wavelet packet decomposition for both, approximation subband and detail subbands, as this corresponds to the maximum

decomposition depth of 5 pairs we use in the isotropic case. Also note that with base value and change factor set to the same value for both isotropic and anisotropic wavelet packets, the randomized isotropic decompositions are expected to be deeper than the anisotropic wavelet packets (in the case of anisotropic generation change factor is taken into account twice as often than in the case of isotropic generation). The rationale for making the anisotropic decompositions less deep than the isotropic decompositions is that in the context of encryption, the keyspace generated by anisotropic wavelet packets is much larger for the same decomposition depth, due to the additional dimension that is introduced by the decomposition direction (horizontal or vertical). In the following chapter, when we perform the security evaluation for the proposed encryption scheme, we will analyze this situation in a lot more detail.

The result tables list the average, minimum, maximum and variance of the 1000 tests for two quality measures: PSNR and ESS. For comparison the compression results obtained with the pyramidal decomposition at wavelet decomposition level 5 and 6 are also given.

It can be seen that for the three test images the average compression results of the wavelet packets selected by the compression oriented approach are competitive with the pyramidal decomposition, especially in the anisotropic case. Due to the settings of base value and change factor discussed above, the performance of the anisotropic wavelet packet decomposition is nearer to the pyramidal decomposition because anisotropic wavelet packets generally require smaller decomposition depths for the same keyspace size. The isotropic wavelet packet require deeper decompositions which introduce a higher overhead and therefore result in slightly lower compression performance. What also can be observed is that for the two images with more high frequency content, Barbara and D105, wavelet packet decomposition exist that outperform the pyramidal decomposition. This is especially noticeable at lower bitrates. D105 has high frequency components in vertical and horizontal direction, therefore here the isotropic wavelet packet transform outperforms the anisotropic wavelet packet transform. (Of course the isotropic wavelet packet bases form a subset of the anisotropic and could also be selected by the algorithm for anisotropic selection, but with 1000 selected bases, selecting the relatively few purely isotropic bases from the large set of anisotropic bases is not probable.) The low variance shows that the selected bases produce reliable results, and the minimum shows that even the worst basis of the 1000 randomly generated bases still yields a compression performance that will be acceptable in most application scenarios.

As regards the edge similarity score, it can be observed that no major differences in terms of preserving edge information exist between the pyramidal, anisotropic and isotropic setup. In summary, although some loss of compression performance occurs, the overall results are encouraging for the three test images.

Rate	WP Type	Measure	Average	Minimum	Maximum	Variance
0.125	pyramidal	PSNR	29.44	29.44	29.44	0.0000
0.125	pyramidal	ESS	0.76	0.76	0.76	0.0000
0.125	isotropic	PSNR	27.40	26.38	28.90	0.1687
0.125	isotropic	ESS	0.67	0.60	0.75	0.0005
0.125	anisotropic	PSNR	28.91	28.17	29.43	0.0466
0.125	anisotropic	ESS	0.74	0.68	0.78	0.0002
0.25	pyramidal	PSNR	32.26	32.26	32.26	0.0000
0.25	pyramidal	ESS	0.85	0.85	0.85	0.0000
0.25	isotropic	PSNR	30.64	29.82	31.69	0.1227
0.25	isotropic	ESS	0.81	0.78	0.84	0.0001
0.25	anisotropic	PSNR	31.76	30.88	32.28	0.0492
0.25	anisotropic	ESS	0.84	0.81	0.87	0.0001
0.5	pyramidal	PSNR	35.16	35.16	35.16	0.0000
0.5	pyramidal	ESS	0.92	0.92	0.92	0.0000
0.5	isotropic	PSNR	33.81	33.10	34.89	0.0912
0.5	isotropic	ESS	0.90	0.88	0.92	0.0000
0.5	anisotropic	PSNR	34.64	34.01	35.15	0.0389
0.5	anisotropic	ESS	0.91	0.89	0.93	0.0000
1	pyramidal	PSNR	38.06	38.06	38.06	0.0000
1	pyramidal	ESS	0.94	0.94	0.94	0.0000
1	isotropic	PSNR	36.89	36.34	37.86	0.0596
1	isotropic	ESS	0.94	0.93	0.95	0.0000
1	anisotropic	PSNR	37.64	36.87	38.03	0.0291
1	anisotropic	ESS	0.95	0.93	0.96	0.0000
2	pyramidal	PSNR	42.90	42.90	42.90	0.0000
2	pyramidal	ESS	0.97	0.97	0.97	0.0000
2	isotropic	PSNR	41.14	40.25	42.31	0.1315
2	isotropic	ESS	0.96	0.95	0.97	0.0000
2	anisotropic	PSNR	42.20	41.41	42.87	0.0713
2	anisotropic	ESS	0.97	0.96	0.98	0.0000

Table 10.1: Compression performance of isotropic and anisotropic wavelet packets for test image Lena

Rate	WP Type	Measure	Average	Minimum	Maximum	Variance
0.125	pyramidal	PSNR	25.20	25.20	25.20	0.0000
0.125	pyramidal	ESS	0.68	0.68	0.68	0.0000
0.125	isotropic	PSNR	24.36	23.68	25.27	0.0811
0.125	isotropic	ESS	0.62	0.58	0.67	0.0005
0.125	anisotropic	PSNR	25.25	24.34	25.90	0.0857
0.125	anisotropic	ESS	0.67	0.64	0.71	0.0001
0.25	pyramidal	PSNR	28.35	28.35	28.35	0.0000
0.25	pyramidal	ESS	0.78	0.78	0.78	0.0000
0.25	isotropic	PSNR	27.31	26.60	28.13	0.0549
0.25	isotropic	ESS	0.73	0.70	0.78	0.0002
0.25	anisotropic	PSNR	28.04	26.62	28.72	0.1345
0.25	anisotropic	ESS	0.77	0.73	0.80	0.0002
0.5	pyramidal	PSNR	32.12	32.12	32.12	0.0000
0.5	pyramidal	ESS	0.87	0.87	0.87	0.0000
0.5	isotropic	PSNR	30.95	30.15	31.96	0.0870
0.5	isotropic	ESS	0.84	0.82	0.87	0.0001
0.5	anisotropic	PSNR	31.77	30.29	32.65	0.1279
0.5	anisotropic	ESS	0.86	0.82	0.88	0.0001
1	pyramidal	PSNR	37.10	37.10	37.10	0.0000
1	pyramidal	ESS	0.94	0.94	0.94	0.0000
1	isotropic	PSNR	35.66	34.69	36.88	0.1507
1	isotropic	ESS	0.93	0.91	0.94	0.0000
1	anisotropic	PSNR	36.67	35.13	37.46	0.1277
1	anisotropic	ESS	0.94	0.92	0.95	0.0000
2	pyramidal	PSNR	43.14	43.14	43.14	0.0000
2	pyramidal	ESS	0.98	0.98	0.98	0.0000
2	isotropic	PSNR	41.76	40.65	43.01	0.1130
2	isotropic	ESS	0.97	0.96	0.98	0.0000
2	anisotropic	PSNR	42.61	41.60	43.27	0.0626
2	anisotropic	ESS	0.97	0.96	0.98	0.0000

Table 10.2: Compression performance of isotropic and anisotropic wavelet packets for test image Barbara

Rate	WP Type	Measure	Average	Minimum	Maximum	Variance
0.125	pyramidal	PSNR	15.05	15.05	15.05	0.0000
0.125	pyramidal	ESS	0.63	0.63	0.63	0.0000
0.125	isotropic	PSNR	16.49	14.51	18.38	0.9882
0.125	isotropic	ESS	0.62	0.57	0.66	0.0002
0.125	anisotropic	PSNR	16.08	14.41	18.80	1.0669
0.125	anisotropic	ESS	0.62	0.58	0.67	0.0003
0.25	pyramidal	PSNR	17.72	17.72	17.72	0.0000
0.25	pyramidal	ESS	0.66	0.66	0.66	0.0000
0.25	isotropic	PSNR	19.64	16.49	21.30	1.7483
0.25	isotropic	ESS	0.70	0.64	0.76	0.0007
0.25	anisotropic	PSNR	18.44	15.64	21.53	1.9267
0.25	anisotropic	ESS	0.68	0.62	0.76	0.0008
0.5	pyramidal	PSNR	21.43	21.43	21.43	0.0000
0.5	pyramidal	ESS	0.77	0.77	0.77	0.0000
0.5	isotropic	PSNR	22.83	20.93	23.51	0.4397
0.5	isotropic	ESS	0.81	0.75	0.84	0.0005
0.5	anisotropic	PSNR	21.57	18.28	23.76	1.7710
0.5	anisotropic	ESS	0.78	0.69	0.85	0.0012
1	pyramidal	PSNR	25.20	25.20	25.20	0.0000
1	pyramidal	ESS	0.87	0.87	0.87	0.0000
1	isotropic	PSNR	25.95	24.76	26.54	0.1880
1	isotropic	ESS	0.89	0.87	0.91	0.0000
1	anisotropic	PSNR	25.24	22.82	26.74	0.9071
1	anisotropic	ESS	0.88	0.83	0.91	0.0002
2	pyramidal	PSNR	30.82	30.82	30.82	0.0000
2	pyramidal	ESS	0.93	0.93	0.93	0.0000
2	isotropic	PSNR	30.68	29.82	31.51	0.1011
2	isotropic	ESS	0.94	0.93	0.95	0.0000
2	anisotropic	PSNR	30.67	28.81	31.91	0.4453
2	anisotropic	ESS	0.94	0.92	0.96	0.0000

Table 10.3: Compression performance of isotropic and anisotropic wavelet packets for test image D105

10.3 EMPIRICAL EVALUATION OF COMPRESSION PERFORMANCE

After having established parameter ranges for the compression-oriented distribution for both isotropic and anisotropic wavelet packets, we verify the compression performance of the compression-oriented selection method in a larger empirical study in this section. For this purpose we use a set of 100 grayscale images of 512×512 pixels, taken with four different camera models (and in different locations: Bangkok, Dubai, Oman, Toulouse, Toronto and Texas). Some examples are shown in Figure 10.12.

We use 5 different bitrates: 0.125, 0.25, 0.5, 1 and 2 bpp. For each of the test images we performed the following JPEG2000 compression tests at each of these bitrates.

- *Pyramidal* (1 basis, level 5)
- *Randomized isotropic wavelet packets with uniform distribution* (100 bases)
- *Randomized isotropic wavelet packets with compression-oriented distribution* (100 bases)
- *Randomized anisotropic wavelet packets with uniform distribution* (100 bases)
- *Randomized isotropic wavelet packets with compression-oriented distribution* (100 bases)

The exact parameters are given in Table 10.4. Note that we follow the convention to give the decomposition depth of the isotropic wavelet packet transform in pairs of (horizontal and vertical) decompositions, whereas in the anisotropic case each (horizontal or vertical) decomposition step is counted separately. To ensure comparability the same seeds (and therefore the same decomposition structures) were chosen for each image at each of the five different rates.

The results of our empirical study are summarized for the 5 categories and all bitrates in Figure 10.13 on page 142. The exact results are given in tabular form in Tables 10.5 and 10.6 (starting on page 143). These tables list the average PSNR, LSS and ESS and the number of samples for each category and bitrate.

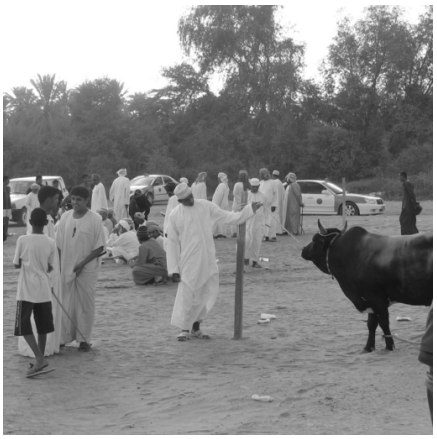
It can be seen that regarding the comparison between pyramidal, isotropic and anisotropic setup, the results obtained for the three test images are largely confirmed. Due to the fact that anisotropic wavelet packets require fewer decompositions for the same keyspace size, the compression performance achieved in the anisotropic setup is superior to the isotropic setup. For the set of natural test images the pyramidal decomposition remains the setup with the best compression performance.

As regards the difference between uniform and compression-oriented selection, it can be seen that the compression performance of the latter is above the compression performance of the former. The difference is more evident for the anisotropic case,

(a)

(b)

(c)



(d)

(e)

(f)

Figure 10.12: Examples from the set of test images

Parameter Name	Isotropic		Anisotropic	
	Unif.	Comp.	Unif.	Comp.
Max. global decomposition depth (g)	5 pairs	5 pairs	10	10
Max. approx. decomposition depth (m)	5 pairs	5 pairs	10	10
Min. approx. decomposition depth (n)	5 pairs	5 pairs	10	10
Max. detail. decomposition depth (d)	5 pairs	5 pairs	10	10
Min. detail. decomposition depth (e)	0	0	0	0
Max. degree of anisotropy approx. sbb. (q)	n/a	n/a	n/a	1
Max. degree of anisotropy detail sbb.s (r)	n/a	n/a	n/a	∞
Base value (bv)	n/a	0.25	n/a	0.25
Change factor (cf)	n/a	0.1	n/a	0.1

Table 10.4: Parameters used for the empirical study

for which a predominant decomposition of the approximation subband in a single direction, which leads to inferior energy compaction for natural images, is possible. Restricting the maximum degree of anisotropy for the approximation subband in the compression-oriented selection leads to compression performance that, for real world applications, is competitive with the pyramidal decomposition.

The bases that give the maximum compression quality could of course also be selected by the uniform distribution. However, it can be seen, that in the total set of bases there are many that yield inferior compression results. In terms of compression performance it is therefore important to limit the number of admissible bases. Only the compression-oriented approach ensures good compression results. In the following chapter we will evaluate if it also yields acceptable security.

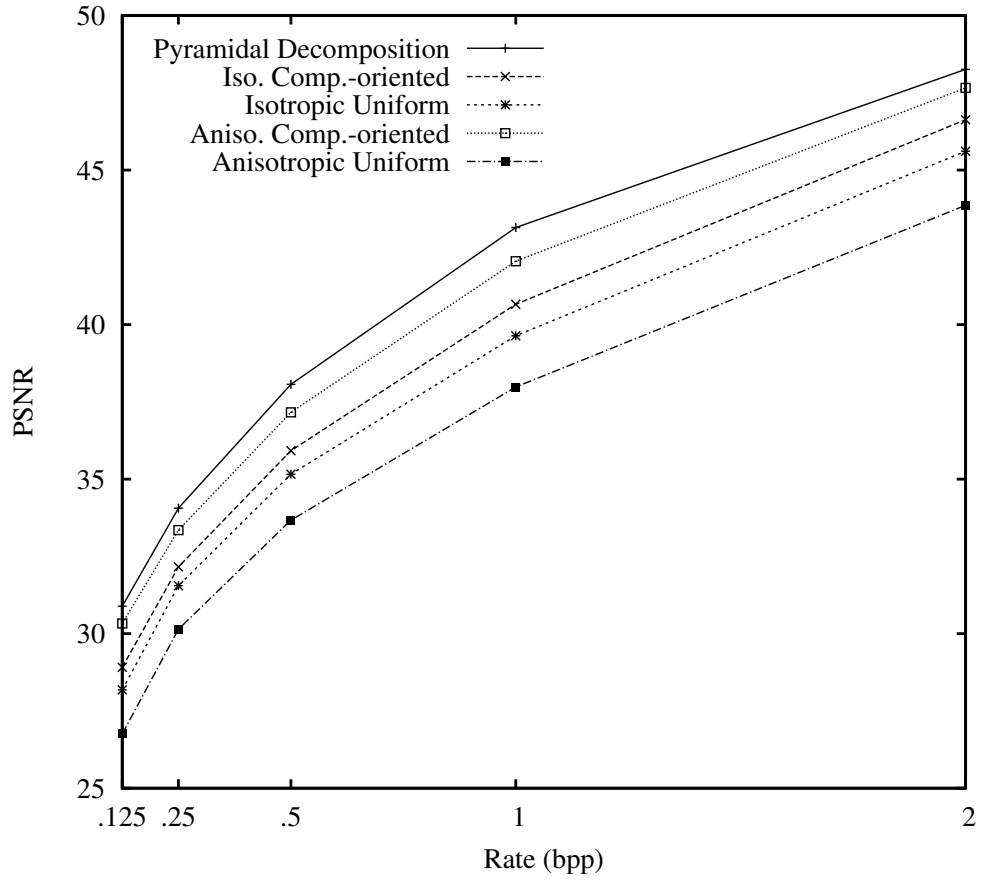


Figure 10.13: Empirical results: average compression performance (100 images)

Rate	Sel. Method	WP Type	Avg. PSNR	Avg. LSS	Avg. ESS	Samples
0.125	pyramidal	pyramidal	30.89	0.50	0.74	200
0.125	compression	anisotropic	30.33	0.47	0.73	10000
0.125	compression	isotropic	28.91	0.37	0.66	10000
0.125	compression	$\sum$	29.62	0.42	0.70	20000
0.125	uniform	anisotropic	26.76	0.18	0.55	10000
0.125	uniform	isotropic	28.18	0.30	0.63	10000
0.125	uniform	$\sum$	27.47	0.24	0.59	20000
0.25	pyramidal	pyramidal	34.06	0.73	0.84	200
0.25	compression	anisotropic	33.35	0.70	0.83	10000
0.25	compression	isotropic	32.16	0.64	0.80	10000
0.25	compression	$\sum$	32.76	0.67	0.81	20000
0.25	uniform	anisotropic	30.14	0.49	0.73	10000
0.25	uniform	isotropic	31.55	0.60	0.78	10000
0.25	uniform	$\sum$	30.85	0.54	0.76	20000
0.5	pyramidal	pyramidal	38.07	0.91	0.91	200
0.5	compression	anisotropic	37.16	0.90	0.90	10000
0.5	compression	isotropic	35.92	0.87	0.89	10000
0.5	compression	$\sum$	36.54	0.88	0.90	20000
0.5	uniform	anisotropic	33.67	0.78	0.85	10000
0.5	uniform	isotropic	35.16	0.85	0.88	10000
0.5	uniform	$\sum$	34.41	0.82	0.87	20000

Table 10.5: Empirical results: compression performance (100 images) 1

Rate	Sel. Method	WP Type	Avg. PSNR	Avg. LSS	Avg. ESS	Samples
1	pyramidal	pyramidal	43.14	0.99	0.96	200
1	compression	anisotropic	42.05	0.98	0.95	10000
1	compression	isotropic	40.66	0.98	0.95	10000
1	compression	$\sum$	41.36	0.98	0.95	20000
1	uniform	anisotropic	37.98	0.95	0.93	10000
1	uniform	isotropic	39.64	0.97	0.94	10000
1	uniform	$\sum$	38.81	0.96	0.93	20000
2	pyramidal	pyramidal	48.26	1.00	0.98	200
2	compression	anisotropic	47.66	1.00	0.98	10000
2	compression	isotropic	46.63	1.00	0.98	10000
2	compression	$\sum$	47.15	1.00	0.98	20000
2	uniform	anisotropic	43.86	1.00	0.97	10000
2	uniform	isotropic	45.61	1.00	0.97	10000
2	uniform	$\sum$	44.73	1.00	0.97	20000
$\sum$	$\sum$	$\sum$	36.38	0.75	0.85	201000

Table 10.6: Empirical results: compression performance (100 images) 11

11 SECURITY EVALUATION

In this chapter we turn to evaluate the security of the encryption schemes that rely on key-dependent wavelet packet subband structures, both isotropic and anisotropic. We investigate the applicability of these approaches in two scenarios: for providing full confidentiality and for providing transparent encryption (as defined in Chapter 1).

The usability of the proposed approaches depends on a couple of factors:

- *Compression performance*: the loss in compression performance introduced by the secret transform domain should be minimal. We have shown in the last chapter that the performance is acceptable. In this chapter we will discuss if the entailing restriction in keyspace size is acceptable in terms of security.
- *Keyspace size*: even when taking the compression-oriented parameters into account, the remaining keyspace, i.e., the number of possible bases, should be large enough to make a full search infeasible. In the last chapters we have already laid the groundwork for determining the number of possible bases. In this chapter, we will investigate the number of bases in more detail.
- *Keyspace quality*: by this we mean how well distinguished the basis in the keyspace are with regard to reconstruction results. We need to show that the bases are sufficiently different that an attacker cannot use a similar basis like the one used for encoding to obtain the original image (or a version that is of almost the same quality). We will perform this investigation in three steps, using a distance metric between decomposition structures:
  - First, we will investigate the actual degree of quality reduction in terms of PSNR that is achieved by the different wavelet packet types and the different selection methods if a random wavelet packet structure is used for reconstruction.
  - Second, we will investigate what distances are achieved by the different distributions.
  - Third, we will investigate if the distance metric between the decomposition structure used for encoding and the decomposition structure used for

reconstructing is a reliable predictor for the quality of the reconstructed image.

- *Codec-specific Attacks*: at the end of this chapter, we will investigate if there is a way to deduce any visual information or information on the wavelet packet decomposition from the meta-information in the JPEG2000 codestream.

Note that only the last point, codec-specific attacks, is solely related to JPEG2000. The size of the keyspace only depends on the number of possible bases and not on the used codec. Also the keyspace quality is largely independent of the used codec. Compression performance will vary by codec, although it can safely be assumed that the principal results will stay the same for different codecs.

11.1 KEYSPACE SIZE

11.1.1 *Number of Isotropic Wavelet Packet Bases*

Uniform Distribution:

The total number of isotropic bases up to decomposition level n is equivalent to the number of quadtrees up to decomposition level n and is easily determined. We have already given one way in Chapter 8 (Equation (8.2) on page 109). Another way is proposed by Pommer and Uhl (2003). They denote the number of possible wavelet decompositions reaching up to level $n+1$ as $f(n)$, which is given as

$$f(n) = \sum_{i=0}^{4} \binom{4}{i} \cdot (f(n-1))^i \tag{11.1}$$

where $f(0) = 1$. Note that other than Equation (8.2), this formula does not count the case where the root is not decomposed (i.e., the original image).

Compression-oriented Distribution:

In order to determine the number of bases in the compression-oriented distribution, we need to take the additional parameters into account. In the following, we accommodate the parameters into the recursive Equation (8.2).

Most importantly, we want to reflect the minimum decomposition depth of the approximation subband. The maximum global decomposition depth will usually be the same for approximation and detail subbands and the minimum decomposition

depth for the detail subbands will be 0 in practical scenarios. To reflect the minimum decomposition of the approximation subband depth n we define N as

$$N(l) = \begin{cases} 0 & \text{for } l < n \\ 1 & \text{for } l \geq n \end{cases} \tag{11.2}$$

wherel l defines the current decomposition depth.

We redefine the number of isotropic wavelet packets Q (and we switch to a functional notation here, rather than using subscripts) as $Q(l, a)$ where l again is the current decomposition depth and a defines if the current subband is in the approximation tree ($a = 1$) or in the detail tree ($a = 0$).

$$Q(0, \cdot) = 1 \tag{11.3}$$
$$Q(l, 0) = Q(l-1, 0)^4 + 1 \text{ for } l > 0 \tag{11.4}$$
$$Q(l, 1) = Q(l-1, 1) \cdot Q(l-1, 0)^3 + N(l) \text{ for } l > 0 \tag{11.5}$$
$$\tag{11.6}$$

Table 11.1 lists the number of bases for different maximum decomposition depths and for the the minimum decomposition depth of the approximation subband set to, as suggested in the last chapter, the global maximum decomposition depth, i.e. $n = g$. It can be seen that setting a minimum level of decomposition for the approximation subband, which is important to eliminate uncompetitive compression results, does not have a significant effect on the number of bases (we lose a bit more than half of the possible bases). Generally, the impact of the compression-oriented parameters is low. This is different for the anisotropic case where the compression-oriented parameters result in a much larger cut in the keyspace, as we will discuss below.

It can also be seen that an exhaustive search for the wavelet decomposition structure is not feasible for sufficiently deep wavelet packet decompositions. For a maximum decomposition depth g, there are $f(g-1)$ possible decompositions. For $g = 5$, 2^{260} possible decompositions exist, and even with the parameter n set (which removes some decompositions), the complexity of an exhaustive search is higher than a brute-force attack against a 256-bit-key AES cipher. For $g = 6$, the number of possible wavelet decompositions is 2^{1045}. The attacker is therefore left with the option of trying to (partially) reconstruct the image from the unencrypted data.

11.1.2 *Number of Anisotropic Wavelet Packet Bases*

Uniform Distribution

We have already given the total number of anisotropic wavelet packet bases in Equations (9.2) – (9.6) on page 117. Table 11.2 compares the number of anisotropic wavelet

g	Compression-Oriented	Uniform (= all bases)
3	$\approx 2^{15}$	$\approx 2^{16}$
4	$\approx 2^{64}$	$\approx 2^{65}$
5	$\approx 2^{260}$	$\approx 2^{261}$
6	$\approx 2^{1045}$	$\approx 2^{1046}$
7	$\approx 2^{4184}$	$\approx 2^{4185}$

Table 11.1: Number of isotropic bases for compression-oriented vs. uniform distribution

g	Isotropic (max. depth g)		Anisotropic (max. depth $2g$)	
3	$\approx 10^{5}$	$\approx 2^{16}$	$\approx 10^{23}$	$\approx 2^{78}$
4	$\approx 10^{19}$	$\approx 2^{65}$	$\approx 10^{95}$	$\approx 2^{315}$
5	$\approx 10^{78}$	$\approx 2^{261}$	$\approx 10^{380}$	$\approx 2^{1263}$
7	$\approx 10^{1260}$	$\approx 2^{4185}$	$\approx 10^{6088}$	$\approx 2^{20225}$

Table 11.2: Comparison of number of anisotropic and isotropic wavelet packet bases

packet bases with $2g$ horizontal or vertical decompositions to the number of isotropic wavelet bases with g pairs of horizontal and vertical decomposition. It can be seen that the increase in keyspace size introduced by the use of anisotropic wavelet packets is substantial. This increase in keyspace size comes at no computational cost, as the computational complexity of the anisotropic wavelet packet transform is the same as the complexity of the isotropic wavelet packet transform. Formulated in another way, we can say that for the same size in keyspace randomized anisotropic wavelet packets need less complexity than isotropic wavelet packets. In the perspective of lightweight encryption anisotropic wavelet packets lower computational demands significantly.

It has to be noted that the numbers given here do not reflect the reduction by the parameter settings that control compression performance. This issue is covered in the following section.

Compression-oriented Distribution

The number of bases in the compression-oriented approach can be determined exactly. Before we do that, we give an approximation in form of a lower bound. The approxi-

mation leads to a formula that is far simpler than the formula determining the exact number of bases.

To obtain a lower bound for the suggested maximum decomposition depth of 12 for the approximation subband and 8 for the detail subbands, we only regard the number of bases induced by the decomposition of the detail subbands, as the restrictions that have most impact on the keyspace size pertain to the approximation subband (due to the proposed settings for minimum decomposition depth n and maximum degree of anisotropy q for the approximation subband). In correspondence to the suggested parameter settings, we assume that no minimum decomposition depth (e) or maximum degree of anisotropy (r) is set for the detail subbands (i.e., $e = 0, r = \infty$), and that the maximum decomposition depth for the approximation subband (m) is greater or equal the maximum decomposition depth for the details subbands (d). We then assume the root to be decomposed either horizontally or vertically (reflected by the factor 2 in the formula below). The approximation subband generated by this initial decomposition is decomposed up to the depth of d in alternating directions, starting with the inverse direction of the root subband. The resulting approximation subband does not exceed the maximum degree of anisotropy $q = 1$ (because the number of horizontal and vertical decomposition differ at most by 1). The decomposition leads to d detail subbands. We assume that each of these detail subbands is either not decomposed or further decomposed an arbitrary amount of times up to level d, with the first decomposition being in the inverse direction of the neighboring approximation subband to avoid isotropic decompositions. For the detail subband at level i this corresponds to at least $1 + (A_i - 1)/2$ possibilities (where A_i, as defined in Equation (9.1) is the number of anisotropic bases up to decomposition level i). The combination of the possibilities in the subtree of each of the d detail subbands gives a lower bound for the number of possible bases that can be obtained with $e = 0$, $r = 0$, $m \geq d$, and arbitrary settings for q and n:

$$2 \cdot \prod_{i=0}^{d-1} \left(1 + \frac{A_i - 1}{2}\right) \leq P_d. \tag{11.7}$$

For $d = 8$, the keyspace size is greater than 2^{302}, and thus above the full search complexity of AES with a 256-bit key.

To determine the exact size of bases available with the compression-oriented constraints in force, we need a little more work. We use the definition of the degree of anisotropy Υ given in Equation (9.7) on page 119. Furthermore, to reflect the minimum decomposition depth n we redefine N as

$$N(l, a) = \begin{cases} 1 & \text{for } a = 0 \\ 0 & \text{for } a = 1 \wedge l < n \\ 1 & \text{for } a = 1 \wedge l \geq n \end{cases} \tag{11.8}$$

where a, like in the isotropic case, defines if the current subband is in the approximation tree ($a = 1$) or in the detail tree ($a = 0$). We now adapt Equations (9.2) – (9.6) to take the compression parameters into account as follows:

$$U(l,h,v,a) = \begin{cases} 1 & \text{for } h+v+1 > d \\ R(0,h,v,a) & \text{for } l = 0 \\ R(l,h,v,a) & \text{for } a = 1 \wedge \\ & \Upsilon(h+1,v) > q \\ U(l-1,h+1,v,a)\cdot & \\ U(l-1,h+1,v,0)+ & \\ R(l,h,v,a) & \text{else} \end{cases} \tag{11.9}$$

$$R(l,h,v,a) = R_A(l) + R_B(l) + R_C(l) + R_D(l) \tag{11.10}$$

$$R_A(l,h,v,a) = N(l,a) \tag{11.11}$$

$$R_B(l,h,v,a) = \begin{cases} 0 & \text{for } l = 0 \\ 0 & \text{for } h+v+1 > d \\ 0 & \text{for } a = 1 \wedge \\ & \Upsilon(h,v+1) > q \\ R(l-1,h,v+1,a)\cdot & \\ R(l-1,h,v+1,0) & \text{else} \end{cases} \tag{11.12}$$

$$R_C(l,h,v,a) = \begin{cases} 0 & \text{for } l = 0 \vee l = 1 \\ 0 & \text{for } h+v+1 > d \\ 0 & \text{for } a = 1 \wedge \\ & \Upsilon(h,v+1) > q \\ U(l-2,h+1,v+1,a)\cdot & \\ U(l-2,h+1,v+1,0)\cdot & \\ R(l-1,h,v+1,0) & \text{else} \end{cases} \tag{11.13}$$

$$R_D(l,h,v,a) = \begin{cases} 0 & \text{for } l = 0 \vee l = 1 \\ 0 & \text{for } h+v+2 > d \\ 0 & \text{for } a = 1 \wedge \\ & \Upsilon(h,v+1) > q \\ R(l-1,h,v+1,a)\cdot & \\ U^2(l-2,h+1,v+1,a) & \text{else.} \end{cases} \tag{11.14}$$

This set of formulas reflects the minimum decomposition depth n of the approximation subband by checking against N. The maximum degree of anisotropy for each subband is handled by comparing Υ to the parameter q. The maximum degree of anisotropy of the detail subbands is not set for the suggested parameters, but it could easily be included in a similar way. The maximum decomposition of the detail sub-

	Compression-Oriented				Uniform
m	n	d	q	#Bases	#Bases
6	0	6	0.5	$\approx 2^{75}$	$\approx 2^{78}$
12	6	12	0.5	$\approx 2^{5048}$	$\approx 2^{5055}$
12	0	8	0.5	$\approx 2^{364}$	$\approx 2^{5055}$
12	6	8	0	$\approx 2^{371}$	$\approx 2^{5055}$
12	6	8	0.5	$\approx 2^{364}$	$\approx 2^{5055}$

Table 11.3: Number of bases for compression-oriented vs. uniform distribution

bands is handled by checking against e. Similarly the minimum decomposition depth of the detail subbands d could be accommodated.

We can now give the exact impact on keyspace size of the compression-oriented approach and compare it to the uniform distribution which uses all possible bases. Table 11.3 lists the numbers for some parameter settings. The last line represents the suggested settings for the compression-oriented approach, which only include bases that perform well for image compression.

It can be seen that the minimum decomposition depth of the approximation subband has little influence on keyspace size. Specifying a minimum squareness factor for the approximation subband does have a considerable impact on keyspace size. The biggest cut in the keyspace is due to setting the maximum decomposition depth d for the detail subbands. The bases with a complex decomposition of the higher frequency subbands are not suitable for the compression of natural images. As the subbands of high decomposition depth in the high frequency subbands are very numerous, a majority of bases is discarded by setting d to a low value.

From a practical point of view the keyspace size provided by the compression-oriented approach with the suggested parameter settings is sufficient. A brute-force attack that tries to find the correct basis has a complexity of 2^{363}. Compared to complexity of a brute-force attack on 256-bit AES, which is 2^{255}, this keyspace size is more than sufficient for applications.

11.2 KEYSPACE QUALITY

It should be noted that a full brute-force attack may not be necessary. If a basis can be found that is close enough to the basis that was used for transformation, there will only be very little distortion in the reconstructed image. Theoretically, such a basis

can be searched for incrementally, by trying to decode as much of the data as possible in resolution-progressive mode and then vary the higher frequency subbands and rerun the decoding. It has, however, been pointed out that the coding of the transform coefficients in JPEG2000 strongly relies on the associated subband structure (see below, and cf. Engel and Uhl, 2006a). Especially for the high frequency subbands, where there is a large number of possibilities it will be hard, if not impossible, to perform this incremental search.

Nevertheless, the question should be addressed, how dissimilar two decomposition structures need to be to ensure sufficient distortion for the wrong key. We perform a replacement attack: Two randomly generated subband structures are produced, one of which is used for encoding. Then the coefficients of the encoding process are decoded with the second subband structure. The distance of the trees is recorded along with a quality comparison of two reconstructed images: one reconstructed with the decomposition structure used for encoding and the other reconstructed with the second subband structure. This experimental setup aims at answering three questions: (i) how similar or dissimilar are the subband structures produced by the different selection methods, (ii) do the pairs of different subband structures result in significant quality distortion when one is used for encoding and the other is used for decoding, and (iii) is the distance metric a good predictor for quality deterioration.

We measure the distance of two isotropic or anisotropic subband structures s_1 and s_2 by comparing leaves common to both decomposition trees. First, all leaves of s_1 and s_2 are assigned a score. We define the score $L(b)$ as

$$L(b) = 2^{m-r(b)} \tag{11.15}$$

where b is a subband, $r(b)$ is the resolution level of b and m is the maximum resolution level in any of the two trees. The rationale for the weighting by resolution level is that differences that occur on the lower resolution levels have a greater impact on visual quality and therefore should be given a higher rating. The total score $L_t(s_1, s_2)$ is computed as the sum of the scores of all individual subbands contained in s_1 and s_2. The score L_u for unique subbands in s_1 and s_2 is defined as the sum of the scores of all subbands that are not contained in s_1 and s_2. The distance between s_1 and s_2 is defined as L_u/L_t. This measure is relatively crude but should suffice to give an idea of the distances the two distributions achieve on average.

The setup for our empirical study is as follows: we use the same 100 images that we used in Section 10.3. For each wavelet packet type and each selection type and each image we generate 100 pairs of bases consisting of B_c and B_f. B_c is used for encoding. We record the distance between the two bases, the PSNR obtained by decoding with the correct basis B_c, and the PSNR obtained when reconstructing with the incorrect basis B_f. For basis selection and encoding, we use exactly the same parameters that

we used in Section 10.3, as given in Table 10.4 on page 141. Again, we perform the test at the same five rates: 0.125, 0.25, 0.5, 1, and 2 bpp. In the following we present three plots for rates 2 and 0.125 bpp:

- The first plot shows the ratio of PSNR that was achieved by reconstruction with the incorrect basis B_f to the PSNR achieved by reconstruction with the correct basis B_c. This answers the question to what extent the different wavelet packet and selection methods reduce the quality of an image obtained with a random wavelet packet basis.
- The second plot shows the relative number of samples by the distance between B_c and B_f. This answers the questions how similar the subband structures are that are produced by the different wavelet packet and selection types.
- The third plot shows how ratio of achieved PSNR and distance are related. This tries to answer the question whether the crude distance metric defined above can serve as a predictor for image quality.

Figures 11.1 and 11.2 show the three result plots for a bitrate of 2 bpp in the isotropic and anisotropic setup, respectively. Figures 11.3 and 11.4 show the results for 0.125 bpp. For the rates between 0.125 the 2 bpp the results also range between the results of the extreme rates. For these rates we present only the plots of the first type. Figure 11.5 shows the plots for the isotropic setup and Figure 11.6 shows the plots for the anisotropic setup.

We start the discussion with rate 2 bpp. Here we have the situation that due to the high rate, the JPEG2000 packets which are created for the higher resolution are full of data. As the differences in the wavelet packets pertain to the higher resolutions, we have very good results for rate 2 for both, the isotropic and the anisotropic case. Figure 11.1 shows the results for isotropic wavelet packets, Figure 11.2 shows the anisotropic case. In the first line of each figure, plots (a) and (b) show the ratio of samples that achieve a certain ratio of the PSNR when decoded with B_f (the bins are set to 1/100). It can be seen that in the uniform selection method, two peaks exist in the isotropic case. Also for the anisotropic case, a second peak exists, but is hardly noticeable at this bitrate; for lower bitrates it becomes more pronounced. The two peaks are due to whether the dimensions of the approximation subband of B_f corresponds to the dimensions of the approximation subband of B_c. If this is the case, then the result will reside in the peak closer to 1, if not in the peak closer to 0. For the compression-oriented distribution we chose for the approximation subband to always be decomposed up to the maximum decomposition depth, therefore there is only one peak. It can be seen that overall, the reduction in visual quality achieved by the compression-oriented distribution is of course less than that achieved by the uniform distribution. However,

in terms of security it is important to note that the maximum quality score achieved by either distribution is 0.7 in the isotropic case. This means that for the randomly generated bases, the compression-oriented distribution provides the same security as the uniform distribution. The same can be said for anisotropic wavelet packets, even though the quality goes up a little for the compression-oriented approach compared to the uniform approach. When comparing the isotropic to the anisotropic setup it can be seen that the latter achieves significantly higher PSNR reduction.

The high reduction in visual quality that can be achieved by the proposed encryption method at high rates is not present at lower rates. Figure 11.3 and 11.4 show the results for 0.125 bpp for the isotropic and anisotropic setup, respectively. In the first line of each figure, plot (a) again shows the uniform selection method and plot (b) shows the compression-oriented selection method. It can be seen that at a lower rate the reduction in visual quality is far less than for a higher bitrate. It should be mentioned that the same bases were used for each image at each rate. Therefore the achieved distance within the randomly generated pairs of bases is the same. The difference to the previous case is that with lower bitrates only the JPEG2000 packets of the lower resolutions carry information. The JPEG2000 packets of the higher resolutions are nearly empty. The differences of the wavelet packet bases in the higher resolutions therefore have only marginal influence on the resulting visual quality. As the higher resolution carry by far the most possibilities for different decompositions, the encryption scheme fails to provide high visual distortion at lower bitrates. For the compression-integrated selection method in the isotropic setup, some bases achieve nearly the full quality, as can be seen in figure 11.3(b). For the anisotropic case the highest visual quality score obtained is around 0.9, still far too high for practical application scenarios.

Figures 11.5 shows the reduction in visual quality for the isotropic case for the rates 0.25, 0.5 and 1 bpp. Figure 11.6 shows the same for the anisotropic case. It can be seen that the higher the bitrate, the better the comparative reduction in quality. The peaks move toward the regions of lower PSNR ratio for higher bitrates for all wavelet packet types and all selection methods. It can also be observed that the reduction in visual quality achieved by the anisotropic wavelet packet transform is superior to the reduction achieved by the isotropic wavelet packet transform. In terms of compression level we have to conclude that the proposed scheme needs a minimum bitrate to be effective. The required reduction in visual quality depends on the application.

Apart from the rate, we found that to some degree the frequency properties of the images influences the reduction in quality. For very smooth images, like pictures with a large portion of sky in it, the reduction in visual quality was far less than for textured images. This is not surprising as the textured images have significantly more energy in the highpass subbands than smooth images. Therefore a difference in the highpass details subbands (where the highest number of decomposition possibilities exist) has less impact on smooth images than on textured images. Of the 100 images, the im-

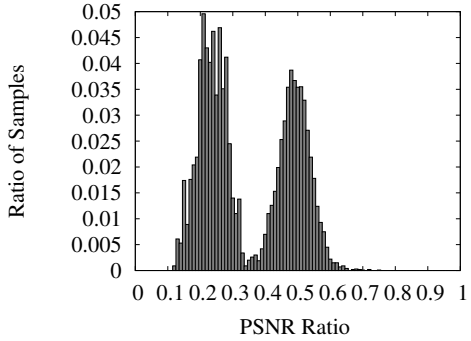
age shown in Figure 10.12(a) (on page 140) exhibited comparatively little reduction in visual quality for most test setups, whereas the image shown in Figure 10.12(b) was among the images with the largest reduction in visual quality.

We now turn to the question of achieved distance, and to the second and third line in Figures 11.1–11.4. Plots (c) and (d) of each figure show the ratio of sample pairs by the distance score they achieve (using the resolution-weighted tree-distance given in Equation (11.15)). As the same bases were used for each image at each rate, plots (c) and (d) are of course identical for the different bitrates. When comparing the two selection methods, it can be seen that the distance achieved by the uniform distribution is higher than the distance achieved by the compression-oriented distribution, for both isotropic and anisotropic wavelet packets. This is especially evident for uniform selection in the anisotropic setup, where the largest number of bases exist. When comparing the different wavelet packet types it is also not surprising that the distances achieved by the anisotropic wavelet packet transform are a lot higher than the distances achieved by the isotropic wavelet packet transform. This is of course due to the higher number of anisotropic bases that exist for a given decomposition depth.

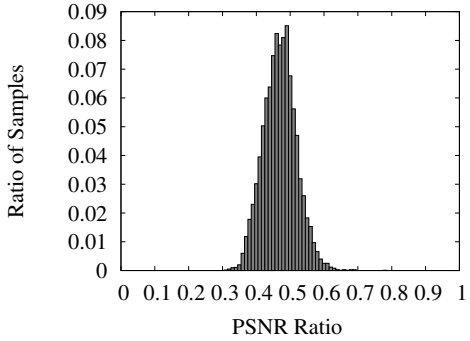
The last question, if the resolution-weighted distance between the decompositions used for encoding and decoding is a good predictor for the achieved PSNR reduction is answered in Plots (e) and (f). What is immediately evident is the fact that the relation between distance and PSNR differs for the uniform and the compression-oriented selection. Within each selection method a weak connection exists between distance and PSNR, but it is far from a conclusive correlation and exhibits a significant number of outlying bins. We can conclude that the crude distance measure can give an idea of the visual quality to be expected, but is not a strong predictor. Note that this is in contrast to the results we reported for only a couple of test images in Engel and Uhl (2007a). Future work in this direction could include a better distance metric and a detailed statistical analysis of the correlation between distance and visual quality.

In conclusion we can say that two key factors influence keyspace quality of isotropic and anisotropic wavelet packets:

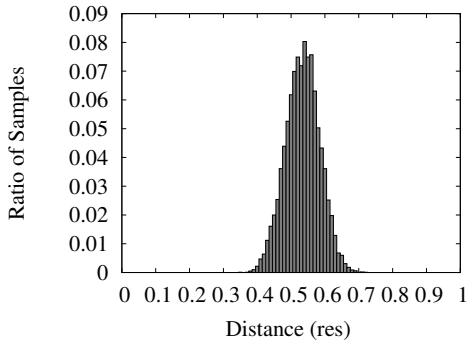
- *bitrate*: if the rate is very low, only the coefficients of the lower resolutions will be part of the generated bitstream and the visual reduction achieved by the proposed scheme will be minimal, and
- *frequency properties*: if the source image material is *very* smooth, then the differences in wavelet packet decompositions will not have a major impact on compression quality.



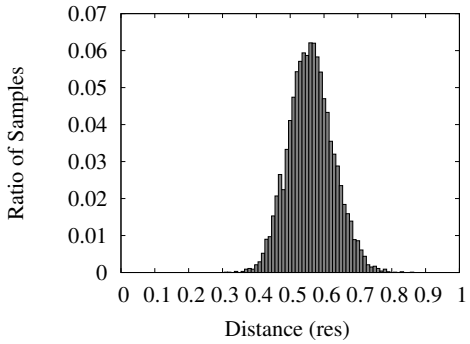
(a) Samples by PSNR ratio, uniform selection



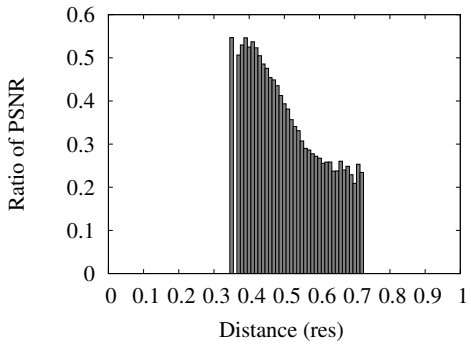
(b) Samples by PSNR ratio, compression-oriented selection



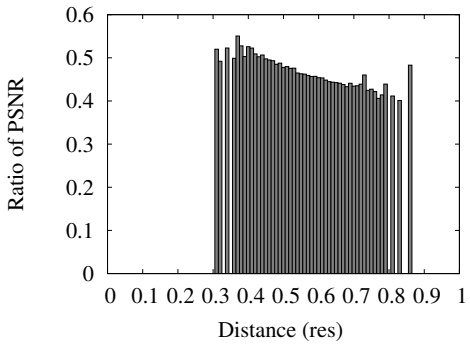
(c) Samples by distance (resolution-weighted), uniform selection



(d) Samples by distance (resolution-weighted), compression-oriented selection

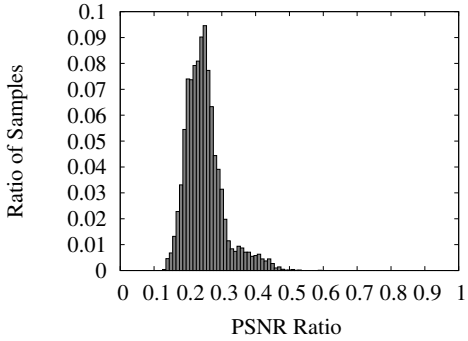


(e) Distance by PSNR ratio, uniform selection

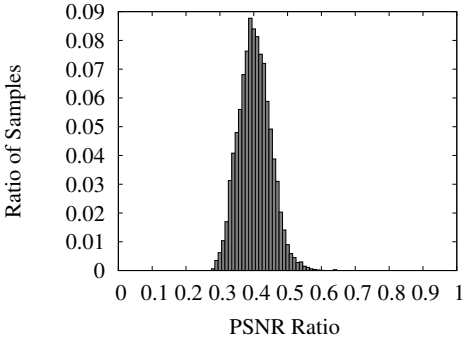


(f) Distance by PSNR ratio, compression-oriented selection

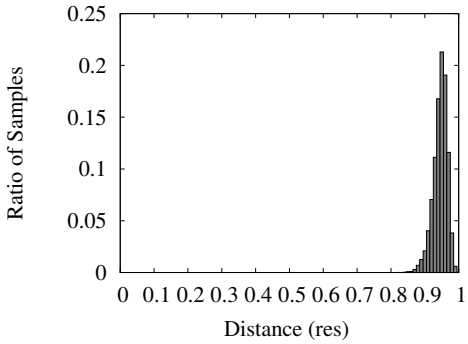
Figure 11.1: Evaluation of randomized isotropic wavelet packets at 2 bpp



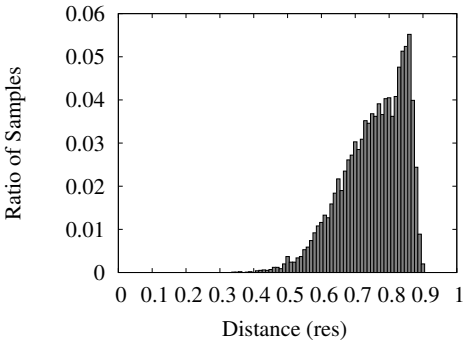
(a) Samples by PSNR ratio, uniform selection



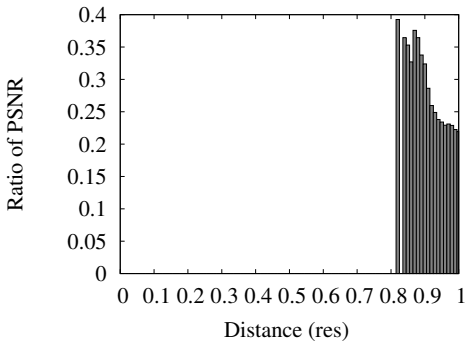
(b) Samples by PSNR ratio, compression-oriented selection



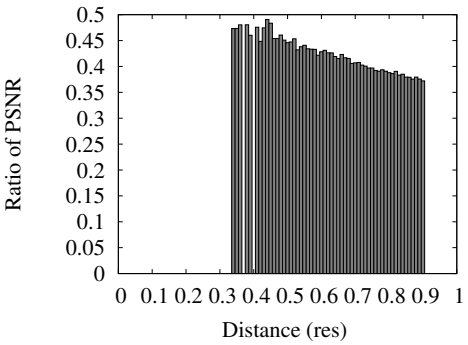
(c) Samples by distance (resolution-weighted), uniform selection



(d) Samples by distance (resolution-weighted), compression-oriented selection

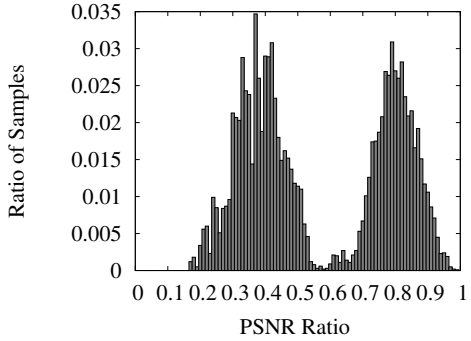


(e) Distance by PSNR ratio, uniform selection

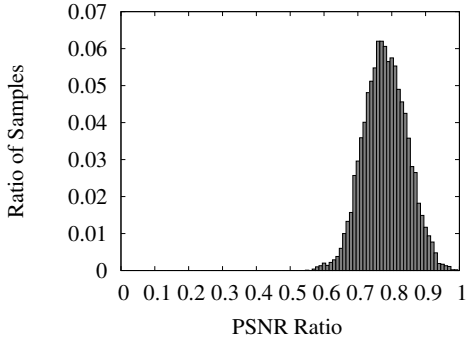


(f) Distance by PSNR ratio, compression-oriented selection

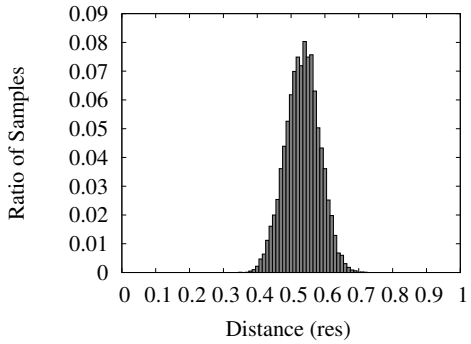
Figure 11.2: Evaluation of randomized anisotropic wavelet packets at 2 bpp



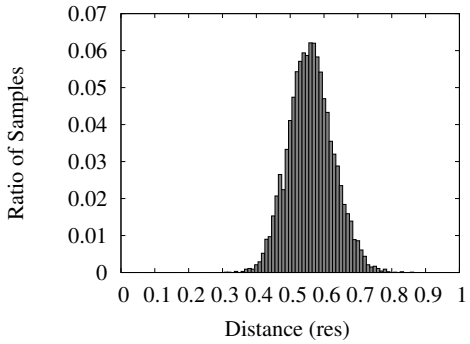
(a) Samples by PSNR ratio, uniform selection



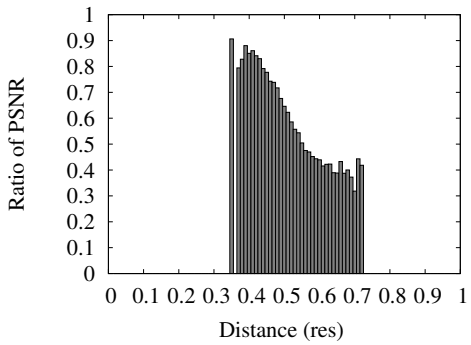
(b) Samples by PSNR ratio, compression-oriented selection



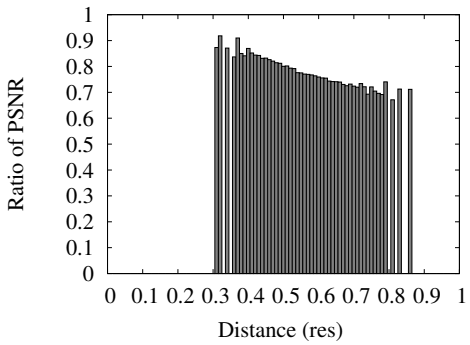
(c) Samples by distance (resolution-weighted), uniform selection



(d) Samples by distance (resolution-weighted), compression-oriented selection

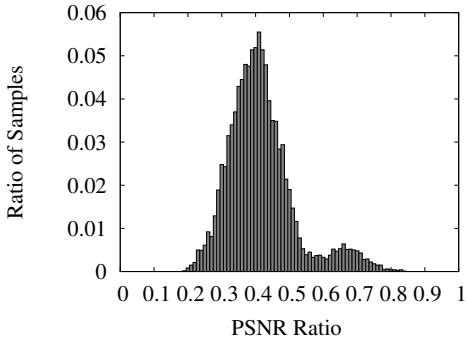


(e) Distance by PSNR ratio, uniform selection

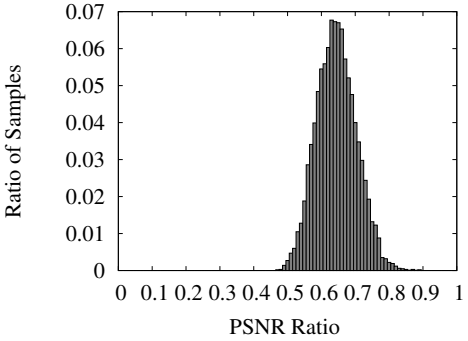


(f) Distance by PSNR ratio, compression-oriented selection

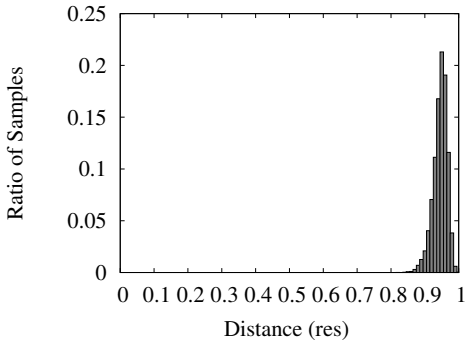
Figure 11.3: Evaluation of randomized isotropic wavelet packets at 0.125 bpp



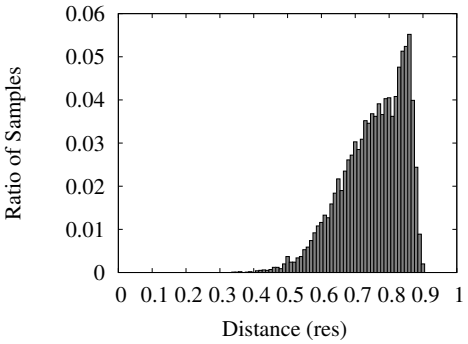
(a) Samples by PSNR ratio, uniform selection



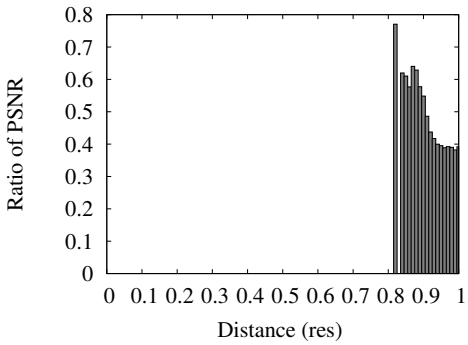
(b) Samples by PSNR ratio, compression-oriented selection



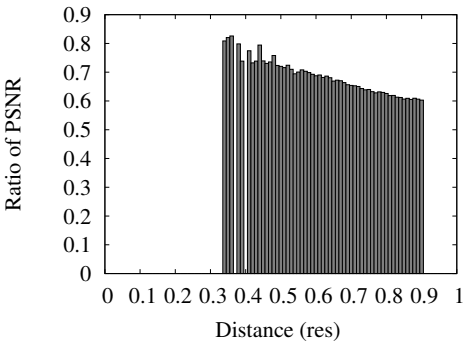
(c) Samples by distance (resolution-weighted), uniform selection



(d) Samples by distance (resolution-weighted), compression-oriented selection

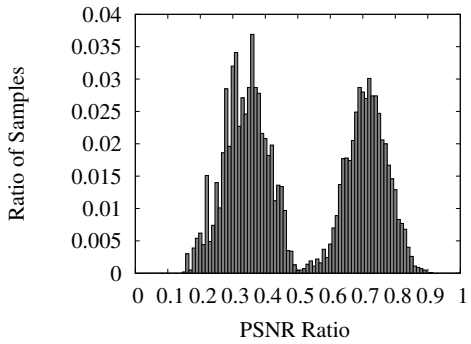


(e) Distance by PSNR ratio, uniform selection

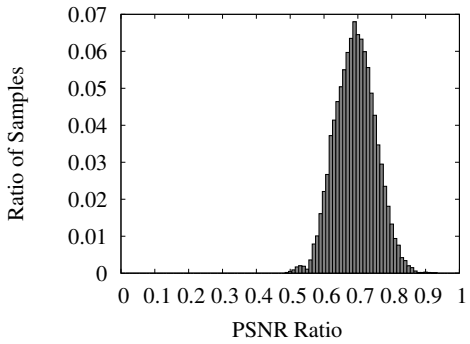


(f) Distance by PSNR ratio, compression-oriented selection

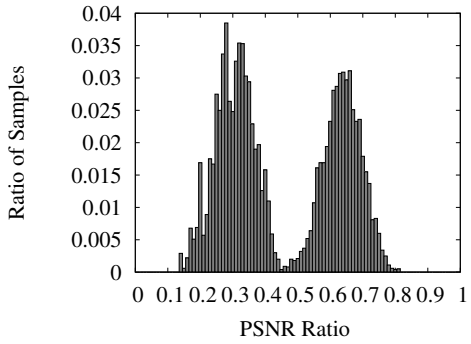
Figure 11.4: Evaluation of randomized anisotropic wavelet packets at 0.125 bpp



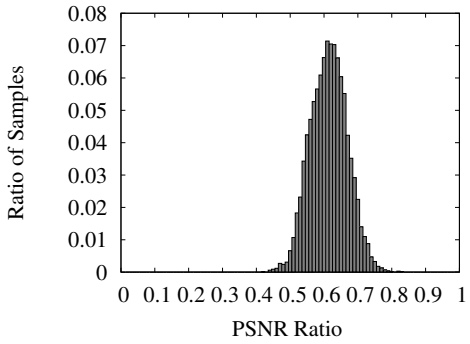
(a) Samples by PSNR ratio, uniform selection, rate 0.25



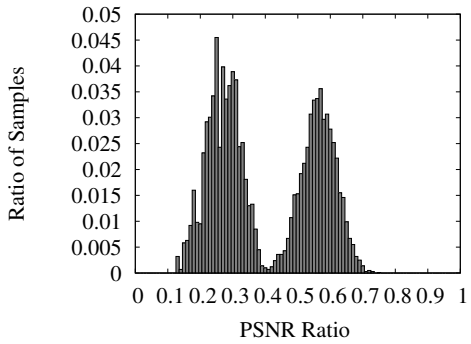
(b) Samples by PSNR ratio, compression-oriented selection, rate 0.25



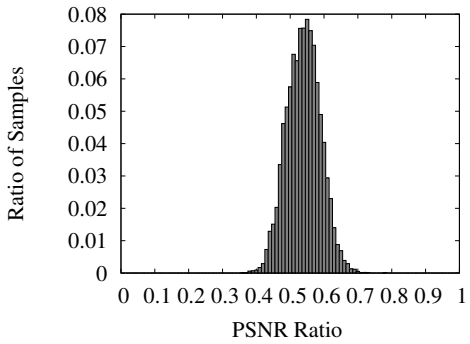
(c) Samples by PSNR ratio, uniform selection, rate 0.5



(d) Samples by PSNR ratio, compression-oriented selection, rate 0.5

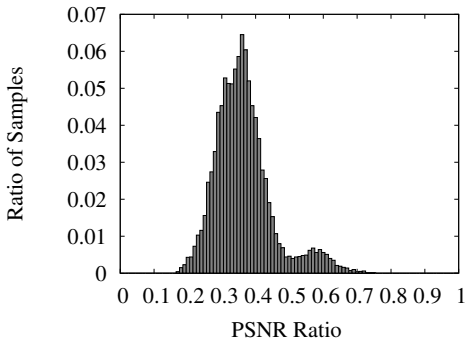


(e) Samples by PSNR ratio, uniform selection, rate 1

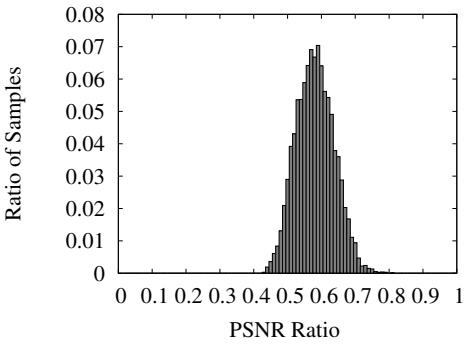


(f) Samples by PSNR ratio, compression-oriented selection, rate 1

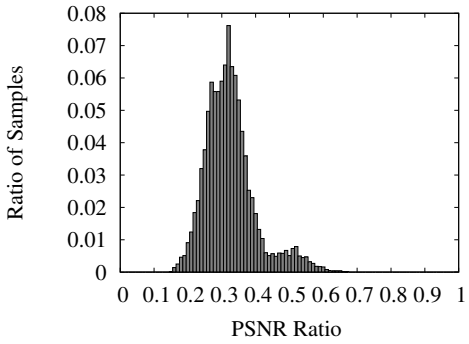
Figure 11.5: Evaluation of isotropic wavelet packets at rates 0.25, 0.5, and 1 bpp



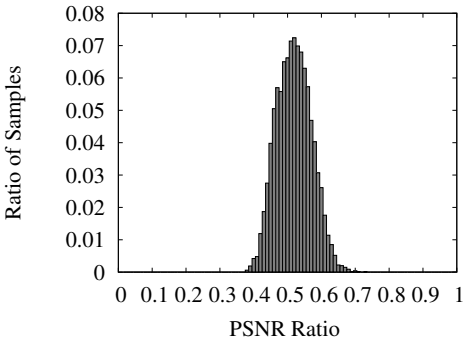
(a) Samples by PSNR ratio, uniform selection, rate 0.25



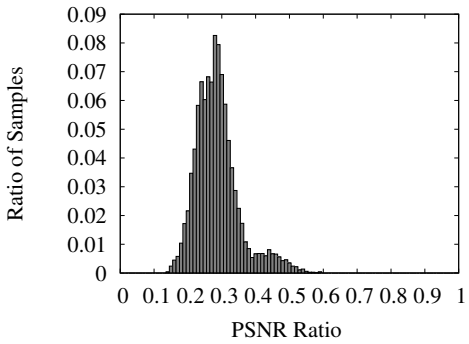
(b) Samples by PSNR ratio, compression-oriented selection, rate 0.25



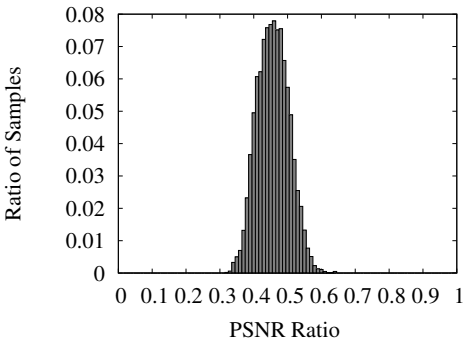
(c) Samples by PSNR ratio, uniform selection, rate 0.5



(d) Samples by PSNR ratio, compression-oriented selection, rate 0.5



(e) Samples by PSNR ratio, uniform selection, rate 1



(f) Samples by PSNR ratio, compression-oriented selection, rate 1

Figure 11.6: Evaluation of anisotropic wavelet packets at rates 0.25, 0.5, and 1 bpp

11.3 CODEC-SPECIFIC ATTACKS

In this section we discuss attacks that are not generally applicable. These attacks use the specific format in which the transform coefficients of the secret transform domain are encoded, in our case JPEG2000, to obtain information on the visual data, the key or the transform domain. We first discuss why in the context of JPEG2000 the proposed scheme is unsuited for providing full confidentiality.

11.3.1 *Full Confidentiality*

The packets of resolution R_0 of any wavelet packet decomposition are the same as the packets produced by a pyramidal decomposition of the same image. Without additional precautions, the first resolution is therefore accessible (in the case of anisotropic packets R_0 is only identical if the LL-subband is isotropic, but even if it is not, a potential attacker only has to take a few guesses to decode the lowest resolution).

The ease of accessibility can be weakened by additionally encrypting header information. If the number of resolutions is not known, then the attacker also lacks knowledge of the size of the approximation subband. However, due to the strong limitations on the minimum and maximum number of resolutions, this measure does not sufficiently increase search complexity. In a similar way, additionally hiding the number of quality layers makes the interpretation of the sequence of packets more difficult for the attacker, but fails to provide the level of security needed for full confidentiality. Because most of the energy is contained in the approximation subband (i.e., R_0), an attacker will be able to get a good idea of the visual data by forcing the data from the packets with the highest payload into a pyramidal decomposition.

Effectively, the presented scheme is not able to restrict access to the lower resolution levels in a manner adequate for providing confidentiality. The solution of encrypting the lower resolutions as such introduces additional computational overhead. Also, this would present a different protection scheme, and should take its bases rather from an existing scheme that uses selective encryption. In such a scenario, it would of course be an interesting experiment to try to add the security provided by secret wavelet packets in the higher resolution levels to approaches that selectively encrypt the lower resolutions, e.g., Norcen and Uhl (2003). As in the present work we are interested in analyzing the security brought by wavelet packets on their own, we will conclude that for full confidentiality the scheme is not suitable and concentrate on the transparent encryption scenario.

11.3.2 *Enforcement of Pyramidal Decomposition*

In contrast to encryption for full confidentiality, in a transparent encryption scheme the accessibility of R_0 is desired, full security is only required for the full quality version. Because hiding the number of resolution levels and quality layers falls more into the category of providing security through obscurity, but does not affect vulnerability of the proposed scheme in principle, in the following we assume that apart from the subband structure every detail about the packet stream is known to the attacker. In particular this means that we assume the attacker to know the size of the image and the number of resolution levels contained in the bitstream. Therefore the attacker is assumed to also know the size of the approximation subband. We also assume that no precinct partitioning is used.

Let p be the value of the parameter controlling the number of higher pyramidal resolutions. Then the packets of resolutions R_0 through R_p are the same as for the corresponding pyramidal decomposition and can be decoded by any JPEG2000 decoder compliant to part 1 without quality loss. For resolution levels higher than R_p, the data does not fit the pyramidal decomposition anymore, and decoding will eventually fail. In order to obtain an image of higher quality than R_p, an attacker could try to read a fraction of the coefficient data of R_{p+1} into the pyramidal structure and then attempt a full resolution reconstruction. However, with the suggested parameters, the intersection of the randomly generated decomposition structures and the pyramidal structure is far too small to obtain data that allows reconstruction at a substantial quality gain (compared to R_p). This is illustrated by Figure 11.7 for $p = 1$ and by Figure 11.8 for $p = 2$.

11.3.3 *Information in the Packet Header*

In this section we turn to the question how much information on the decomposition structure (and the wavelet coefficients) is leaked by meta-information contained in the JPEG2000 codestream.

Apart from the encrypted parameters, no explicit information on the anisotropic wavelet packet structure is contained in the header data.

However, we need to investigate the situation for implicit information. The inclusion information, which is contained in the packet headers, is of course the point where the decomposition structure is needed. The crucial questions are if (i) the decomposition structure is necessary to decode the inclusion information and if (ii) the decomposition structure can even be inferred from the inclusion information. A positive answer in any case implies a security threat for the proposed scheme.

(a) Decomposition structure, $p = 1$ (b) Reconstruction at full resolution (c) Result of attack

Figure 11.7: Reconstruction example for $p = 1$

(a) Decomposition structure, $p = 2$ (b) Reconstruction at resolution 2 (c) Result of attack

Figure 11.8: Reconstruction examples for $p = 2$

If the inclusion information can be decoded correctly without the decomposition structure, then also the wavelet coefficients can be correctly decoded. If the wavelet coefficients are known, then also the decomposition structure is revealed, as it can be easily determined from the coefficients by simple statistical methods, as has been shown by Pommer and Uhl (2003).

JPEG2000 employs so-called tag-trees (Taubman, 2000) to signal inclusion information (see also Chapter 3): In a highly contextualized coding scheme, the contributions of each code-block contained in a packet are linked to the subband structure. Thereby the subband structure is used as context to interpret the output of the tagtrees. If the subband decomposition structure is unknown, the attacker Mallory generally cannot correctly interpret this output (we will discuss the exceptions to this below): Mallory can only see the answer given to an inclusion question, but, lacking the decomposition structure (and therefore information like the size of the array the codeblock refers to), does not know the right question. Mallory cannot assign codeblocks to subbands. Furthermore, the fact that the inclusion information cannot be decoded eliminates access to the raw coefficient data, as an attacker cannot correctly associate the contributions of a code-block to the correct coefficients. Lacking the wavelet coefficients, Mallory cannot use statistical analysis to infer the wavelet packet structure from the coefficients. In this respect the security of the secret frequency domain is strongly dependent on the used codec: for other codecs the coefficients may well be available, see the discussion by Pommer and Uhl (2003).

If the packet headers were encoded with an arithmetic coder and no side information existed, then the statement of the above paragraph would hold in any case. However, the packet headers do not use all possible bitpatterns. Therefore, a sequence of bits cannot be interpreted in any arbitrary way. Furthermore, there is side information, namely that the possible interpretation has to fit the packet body data. The crucial question therefore is how many interpretations are there for a given packet header (in the context of a wavelet packet scenario), given the size of the packet body data. As discussed in Chapter 3, the packet header contains header information for each codeblock in each subband in the packet (provided that no precinct partitioning is used). For each codeblock, first the inclusion information is contained. For included codeblocks this is followed by the number of leading zero bitplanes, the number of coding passes and the length of the codeblock contribution to the packet. This is illustrated by Figure 11.9. The inclusion information for codeblocks that have never been included before is coded in form of tag-tree output. For codeblocks that have been included before, a 0 or a 1 is encoded to signal non-inclusion or inclusion. The leading zero bitplanes are coded as tag trees. The number of coding passes are coded using fixed code words and the length of the codeblock contributions are coded by a length indicator that can be incremented by number k by coding k ones and then a zero.

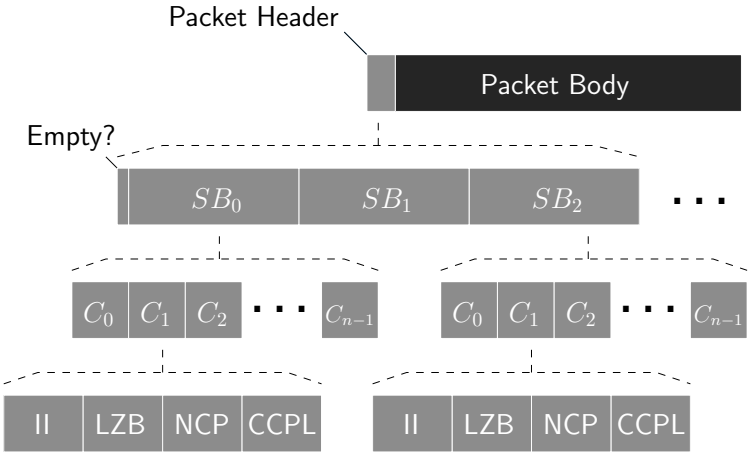


Figure 11.9: JPEG2000 packet header for wavelet packets

Given a packet header, Mallory's problem is to find all possible interpretations of this header that assign values to all classes of header information that are consistent with the packet body data without knowing the subband structure. As he does not know the subband structure, he does not know where the codeblock headers for one subband stop and the headers continue for the next subband. Therefore it will be hard for Mallory to assign refinement passes of later packets to the correct codeblocks. Mallory could start by scanning for the fixed codewords to form a hypothesis. He could look for runs of ones and interpret them as length indicators. He could try to assume inclusion information for a hypothetical number of codeblocks per subband (which will be difficult as he knows neither number of subbands nor the number of tagtrees, but he does have a lower and upper limit on the number of codeblocks in the packet).

The extent of Mallory's success will depend on a number of factors, like the number of quality layers. For fewer quality layers, fewer choices exist and Mallory will have a higher chance of deciphering the header. Furthermore, Mallory's success will also depend on whether he knows the packet boundaries (through SOP markers) and if he additionally knows the end of packet header positions (through EPH markers).

In summary we can state that the structure in the packet header means that information is leaked, which in some cases may lead Mallory to all coefficient data and therefore the decomposition structure and the full quality image.

The security of the scheme can be restored, at the cost of additional computational complexity. A different wavelet packet structure can be used for writing the header. Any decomposition structure can be used for encoding the packet headers. In this scenario apart from the structure used for transformation another structure is written to

the codestream, which is used for encoding the packet headers. After the codeblocks have been decoded the structure used for transformation can be used to restore the coefficient. This prevents Mallory from inferring (parts of) the correct wavelet packet structure from the packet headers. However, the threat of Mallory being able to decode some or all coefficients remains. In order to counteract this threat, the packet headers need to be protected from access. This can be done by conventional encryption (to retain bitstream compliance a format-compliant scheme can be used), or by using the header protection scheme proposed in Chapter 3.

11.4 CONCLUSION

The compression-oriented approach for lightweight encryption with wavelet packets discards a number of possible bases to produce good compression results with randomly selected bases. The achieved compression performance is better for anisotropic wavelet packets than for isotropic wavelet packets, as they need a smaller number of decompositions for the same set of bases.

By comparing compression performance to a method that randomly selects from all possible bases, we have shown that pruning the set of all bases is necessary especially in case anisotropic wavelet packets are used, as compression results vary immensely. If all bases were used, the approach would not be suitable for application. In order to assess the loss in keyspace size we formally determined the total number of bases and the number of bases for the compression-oriented approach for isotropic and anisotropic wavelet packets. A comparison shows that the number of bases that are suitable for image compression is small compared to the total number of possible bases but is still far above the complexity of a brute-force attack against 256-bit AES.

A quality assessment of the set of available bases showed that the expected distance is higher for the uniform distribution. We have also shown that the expected distance is higher in the anisotropic case.

As regards reduction in visual quality, we have show that the maximum visual quality retained was not significantly increased by the compression-oriented distribution. However, our results clearly showed that the lower the bitrate the lower the reduction in visual quality. For very low bitrates, the scheme becomes ineffective.

We have found the level of security to be too low for applications that demand full confidentiality. In terms of transparent encryption the scheme is successful, as it can naturally be used to grant access to lower levels of resolution while keeping the higher resolution levels secure.

We have also analyzed to what extent JPEG2000 meta-information can be used to recover the decomposition structure and have identified the packet headers as a

potential source of leakage. As the packet headers have structure and leak information that can potentially be used in an attack, they should be secured.

An advantage of the presented schemes is that the amount of data that needs to be encrypted is extremely small and remains confined to the packet headers and the description of the wavelet packet decomposition structure. Furthermore, the schemes are naturally format-compliant. This is a significant advantage of the presented approach compared to other suggestions. In the following final chapter of this part we discuss possible application scenarios.

12 APPLICATION SCENARIOS

As discussed in Chapter 1, scenarios for the employment of encryption in multimedia environments can be divided into bitstream-oriented and compression-integrated encryption, depending on where in the compression pipeline encryption takes place. Furthermore we can divide application scenarios into on-line and off-line scenarios, distinguishing whether the data is given as plain image data (i.e., not compressed) or in form of a bitstream resulting from prior compression. In this chapter we discuss possible application scenarios of the compression-integrated approach for transparent encrpytion that is based on a key-dependent wavelet transform compared to bitstream-oriented approaches.

There are many application scenarios for transparent encryption for which both, bitstream-oriented and compression-integrated approaches can be used. An example is TV broadcasting: a preview video is available for all users, but the full-quality version remains exclusive to the paying subscribers. Also in image databases, the availability of a thumbnail is of advantage as an incentive for buying the full-quality version. The same is true for online video databases, of course.

For video surveillance it is sometimes desirable to show the video feed in order to discourage theft. However, privacy should be protected. A possible solution is to only show the poor quality as a deterrent. If an incident should occur then the full quality version can be accessed by security personnel.

The main difference between compression-integrated and bitstream-oriented approaches is the fact that bitstream-oriented approaches can be used for on-line and off-line scenarios, whereas compression-integrated approaches are mainly restricted to on-line scenarios (because if compression has already been done, transcoding would become necessary to apply the encryption in this case). This makes bitstream-oriented approaches more flexible in comparison. On the other hand, compression-integrated encryption also offers some advantages: (a) only very little data has to be encrypted even compared to selective encryption schemes (of course this comes at the cost of increased complexity in the compression pipeline), (b) format-compliance comes naturally and does not need to be taken care of explicitly (as it is the case for bitstream oriented approaches), and (c) signal processing is to some extent possible in the en-

crypted domain, which allows secure watermarking and intrinsic fingerprinting, as discussed below.

It should be noted for (c) that signal processing in the encrypted domain is also possible for partial bitstream-oriented encryption to some extent, because only small amounts of data are encrypted and even the encrypted data can be processed on a packet or even CCP basis, e.g., enabling efficient transcoding and cropping. The advantage of the compression-integrated approach is that the full transform coefficients are available for signal processing. In the case of JPEG2000, for example, key-dependent wavelet packet subband structures allow direct access to the transform coefficients and with a codec compliant to JPEG2000, part 2, all features of JPSearch can be used. In the context of key-dependent wavelet packet subband structures in JPEG2000, the fact that a part II compliant decoder is needed to access the full quality version can be seen as another advantage: Even if a key is compromised, the visual data is only accessible with such a decoder (if no transcoding is performed of course), which might hinder dispersion of the full quality version. A traitor-tracing scheme can be employed that uses the unique wavelet packet structure to determine the source of pirated visual data. This works in the second and third scenario and is discussed below.

For both compression-oriented and bitstream-oriented approaches we can combine symmetric transparent encryption with public key encryption to facilitate distribution and key-management. For the bitstream-oriented approach the visual data is first encrypted symmetrically (using one of the schemes discussed previously). Then the key of the symmetric encryption scheme is encrypted with a public-key scheme and sent to the (paying) user. In the case of the compression-integrated approach, the visual data is first encoded with a randomly selected basis. For a paying user, the description of this basis is then encrypted with the user's public key and sent to the user. This basically amounts to a hybrid encryption scenario with combined symmetric and asymmetric encryption: The description of the basis used for encoding is the key to the symmetric encryption scheme, key management and distribution is realized by classical public key encryption. Both kinds of approaches can be used for protection on three different levels: (a) individual protection for each image (or groups of images) regardless of the associated user, (b) individual protection for each user (for all images associated with the user) or (c) individual protection for each image and each user. We use wavelet packets in the following discussion as a representative of compression-integrated encryption, but it should be noted that the reflections are valid for other key-dependent transform domains as well.

12.1 PROTECTION FOR EACH IMAGE

This approach presents a pay-per-item implementation. Each image (e.g., in a database) is encrypted with an individual key or decomposed with a unique wavelet packet subband structure respectively. Any non-paying subscriber can get a preview image.

If we allow more than one image to share the same symmetric key (or the same subband structure), then a channel-based subscription service can be implemented: if, for example, all sports pictures share the same symmetric key, a person paying for the sports key can access all of the pictures in this group.

For compression-integrated encryption, the unique frequency domain of each image allows copy-tracing. For example, a crawler could be employed to search for images with a particular subband structure. This can be done by matching the coefficients in the transform domain, or by reconstructing the image and matching in the spatial domain. Even if transcoding is employed, there can be some traces of the subband structure left, as outlined in the following scenario.

12.2 PROTECTION FOR EACH USER

In this scenario, a unique symmetric key (or subband structure) is assigned to each user. In an image database with personal albums, for example, this key would be used for each item the user uploads.

A nice feature of key-dependent transform domains in this context is that the transform domain can be used to trace the user's data if it gets redistributed. A crawler could, for example, look for images coded in a particular user's subband structure. Even if transcoding was used, e.g., if the JPEG2000 image has been transformed to a JPEG image, due to quantization there could be traces left in the visual data that allow the linking to the subband structure of a particular user. To make the tracing scheme more reliable, a watermark can be embedded in the wavelet packet domain (Dietl and Uhl, 2003).

Another feature provided by the use of user-dependent wavelet packet structures are secure annotation watermarks. As these watermarks can be embedded in the wavelet packet domain to enhance security, only a person knowing the subband structure can extract the contained information (Dietl and Uhl, 2004). Any person lacking the subband structure can only access the preview image, but not the annotations. As the secret transform domain is already available, apart from embedding no additional computational costs are introduced.

12.3 PROTECTION FOR EACH USER AND EACH IMAGE

This scenario is similar to the previous scenario, but a new symmetric key (or subband structure) is created for each of the user's images. This is important if one or some of the images get sold and access should not be given to the other images. This is especially interesting for personalized visual data, for example, personalized greeting cards. Each personalized card is encrypted using a unique symmetric key, only the user who ordered the card may access the full quality version. Another example are blueprints, which an architect creates for a specific customer. The customer can get a preview image, but to access the blueprints in full detail, he has to obtain the symmetric key. A different key is used for the next set of blueprints from a different order, even if it is the same customer. Also in this scenario the possibilities of copy-tracing and secure annotation watermarks come for free with the use of key-dependent transform domains.

12.4 CONCLUSION

Like bitstream-oriented partial / selective encryption schemes, the compression-integrated approaches based on key-dependent transforms succeeds in reducing the computational demand for encryption significantly as compared to traditional transparent encryption methods.

While a bitstream-based encryption technique is more flexible as it can be employed in both on-line and off-line scenarios, the secret wavelet packet approach requires a smaller amount of data to be encrypted (which has to be paid for with a higher complexity in the compression pipeline) and does not have to obey any restrictions imposed by the requirement for format compliance as required for bitstream packet data encryption. Therefore, for specific application scenarios, e.g., such that require public-key cryptography to be applied (where a minimal data amount subjected to encryption is a must and bitstream compliance is harder to achieve), the compression integrated wavelet packet technique is an interesting alternative to bitstream-based transparent encryption.

To conclude this part, it should be noted again that the discussed schemes are specimen of lightweight encryption. Full confidentiality in the sense of cryptographic security cannot be provided. For this the only option is to use a traditional cipher on the complete source data.

Part IV

ANALYSIS OF BITPLANE-BASED IMAGE ENCRYPTION

13 ANALYSIS OF BITPLANE-BASED LIGHTWEIGHT FINGERPRINT ENCRYPTION

In the following two chapters (cf. Engel and Uhl, 2007b; Engel et al., 2008a) we analyze security schemes that are based on bitplane extraction. The context of our analysis is provided by a specific scheme proposed for the encryption of fingerprint images. We will analyze the bitplane properties of the 3200 images in contained in the database of the fingerprint verification contest 2004 (FVC2004).[1] We will then present attacks on the "image-based selective bitplane encryption protocol" (Moon et al., 2006).

13.1 INTRODUCTION

The increasing use of biometric systems raises the question of how to store and handle biometric data in a secure way. In this respect, sample data, i.e., data acquired by the sensor, is more sensitive than template data. Whereas template data contains a description of the biometric features that pertain only to the used system, sample data can potentially be used on other systems as well. As generally the revocation of biometric data is extremely problematic, a compromise of this data has severe security implications. Therefore, the secure handling and storage of biometric sample data may become imperative due to the security and privacy concerns of the users.

Usually, classical cryptographic techniques are suggested to be used for biometric sample data Maltoni et al. (2003). A small number of specific techniques has been developed for fingerprint sample images. A Fourier-type transform-based private encryption scheme for fingerprints is proposed in Soutar et al. (1998). The concept of "cancelable biometrics" is proposed in Ratha et al. (2001), where the acquired biometric signal (i.e., the sample) is distorted with an intentional repeatable transform before the extraction of the template. In case of a compromise, the transformation can simply be changed. A very similar approach are the so-called "biometric cryptosystems" proposed in Uludag et al. (2004) and Ratha et al. (2001): A secret transformation is applied to the biometrics templates to render them useless for intruders. Matching can be performed in the encrypted domain.

[1] http://biometrics.cse.msu.edu/fvc04db/index.html

(a) DB1_A (b) DB2_A (c) DB3_A (d) DB4_A

Figure 13.1: Example fingerprint images from the FVC2004 database

Controlling the computational demand is important, especially in distributed scenarios with weak and low-power sensor devices. Classical encryption techniques can be too demanding to be employed, therefore a careful but significant reduction of encryption complexity is required for this type of applications. The limited computational resources in embedded processors are addressed in recent work by Moon et al. (2006), where an approach involving selective encryption of fingerprint images employing XOR on a bitplane basis is suggested. We will analyze this approach in detail in the next chapter and demonstrate several shortcomings and a computationally efficient attack.

Before we turn to the analysis of these specific approaches, we will investigate the bitplane properties of fingerprint images. Random properties have been attributed to these bitplanes, e.g., by Droogenbroeck and Benedett (2002b) and Moon et al. (2006).

13.2 RANDOMNESS OF FINGERPRINT BITPLANES

The four databases of the fingerprint verification contest 2004 (FVC2004)[2] contain samples from different sensors. Databases 1 and 2 contain images of two different optical sensors (DB1, DB2), database 3 originates from a thermal sweeping sensor (DB3), and database 4 consists of synthetically generated prints (DB4). Figure 13.1 displays an example fingerprint from each database (the frames for DB1 and DB2 are only drawn to show the image dimensions, and are not an actual part of the fingerprint images).

In order to find a measure for quantifying discernibility from noise in the bitplanes of these images, there are a couple of possible approaches. A direct approach is to perform an statistical analysis on the bitplanes. Another approach that is discussed in Engel et al. (2008a) is to count the number of consecutive runs and how well the bit-

[2] http://biometrics.cse.msu.edu/fvc04db/index.html

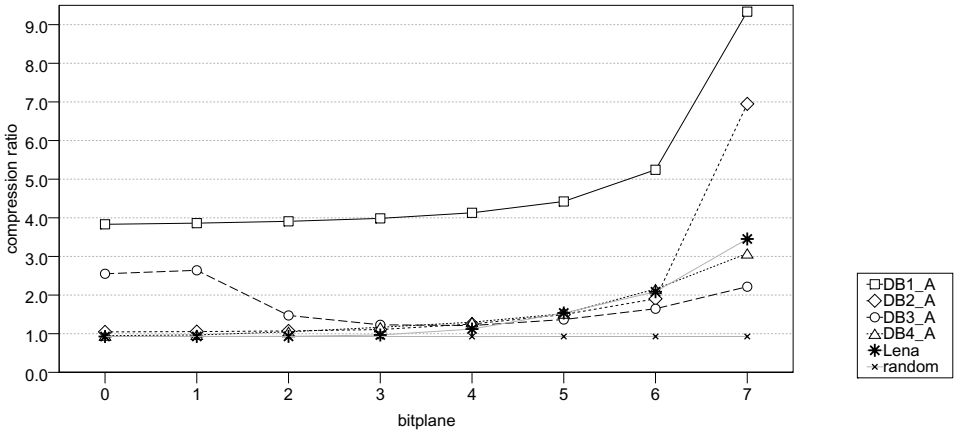


Figure 13.2: Compression performance of arithmetic coding by bitplane

planes can be compressed by an arithmetic coder. We will recount the latter approach here.

From each set, 100 fingers with 8 prints each are used, the plots show average compression rates. For the sake of a comparison to a "classical" digital image we also present the values of bitplanes of the Lena image. As a reference, we have also added the results for an image which only contains random noise. The image was generated by using the Mersenne Twister PRNG (Matsumoto and Nishimura, 1998) to produce a random byte for each pixel value.

The idea for this test setup is that if the bitplane can be compressed well, it cannot be very close to a random field. We use the arithmetic coder proposed by Moffat et al. (1998) in a mode where it accepts a sequence of bits without a specific background model. Figure 13.2 shows the compression ratio of each bitplane for the fingerprint images in the four databases and, as a reference again, for Lena and the randomly generated image. It can be seen that the arithmetic coder is successful at compressing all bitplanes of the images in DB1, including the LSB-plane. An interesting phenomenon can be observed for the images in DB3: the thermal sensor yields images for which the medium bitplanes exhibit more noise than the LSB-plane. The compression ratios for the LSB-planes of the images in DB2 and DB4 and also Lena are nearly as low as the results for the randomly generated image. For these images the lower bitplanes are closer to random than for the other images.

13.3 CORRELATION WITHIN THE BITPLANES OF FINGERPRINT IMAGES

Another point apart from the randomness of the LSB-plane is the correlation of each bitplane with the other bitplanes. To assess this correlation, we compute the sample

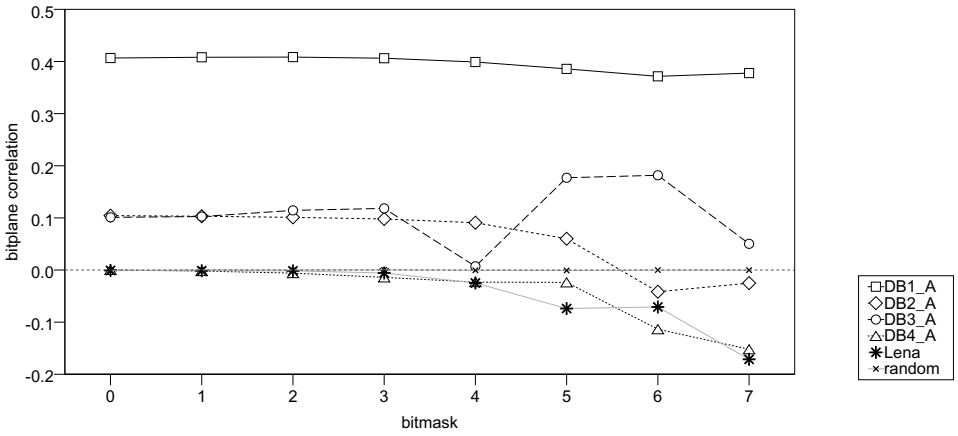


Figure 13.3: Correlation of each bitplane with the other bitplanes

correlation of the bitplane under consideration with each other bitplane, and then average the correlation values. The results for each bitplane are shown in Figure 13.3.

For the images of DB1, which have a uniform background, the correlation is naturally high. The phenomenon of more noise in the medium bitplanes in the case of DB3 can again be observed in the lower correlation of these bitplanes, especially for the fifth bitplane. For DB2 a little correlation can be observed at lower bitplanes. The lower bitplanes of DB4 and Lena exhibit no correlation to the other bitplanes, just like the randomly generated image.

13.4 CONCLUSION

It should be noted that better sensors, due to inherent (thermal) noise, may produce images with more random LSB-planes as compared to weaker sensors. Furthermore, if post-processing is applied, it may influence the randomness of the LSB-plane in one or the other direction. What our observations show is that the assumption that all scanners always produce images for which the LSB-plane is (close to) a random field that shows no correlation to the other bitplanes is not true and is, in fact, a dangerous assumption with regard to security.

14 AN ATTACK AGAINST IMAGE-BASED SELECTIVE BITPLANE ENCRYPTION

In this chapter, we analyze a recently published lightweight encryption scheme for fingerprint images and discuss several shortcomings. A low-cost attack on this scheme is proposed, which allows access to the full plaintext for most given ciphertexts. We give some recommendations for improvements of the encryption scheme, but conclude that the analyzed scheme remains insecure.

14.1 IMAGE-BASED SELECTIVE BITPLANE ENCRYPTION

In recent work Moon et al. (2006), a lightweight fingerprint image encryption technique has been proposed, which has been denoted as "image-based selective bitplane encryption protocol". The approach, which is essentially a Vigenère cipher, constructs a keystream from the least significant bit (LSB) plane of the input image. This keystream is XORed with the binary representation of the plaintext data and then encrypted with AES. The aim of the approach is a reduction in computational complexity to facilitate real-time processing on low-end processors.

Let I be the original 8 bpp fingerprint image with a width of w pixels and a height of h pixels. s denotes the size of the image in bits, $s = h \cdot w \cdot 8$. Consider now the binary representation of the image I being given as

$$I = \{b_0, b_1, \cdots, b_6, b_7, b_8, \cdots, b_{s-1}\}$$

where $b_{m \cdot 8}$, $0 \leq m \leq h \cdot w - 1$ is the MSB of the binary representation of pixel $m + 1$, whereas $b_{m \cdot 8-1}$, $1 \leq m \leq h \cdot w$ is the LSB of pixel m. We extract a set of key bits

$$K_0 = \{k_0, \cdots, k_{h \cdot w-1}\}$$

where the m-th keybit k_m of key K_0 is constructed by taking the LSB of each pixel m, i.e., $k_m = b_{m \cdot 8-1}$, $1 \leq m \leq h \cdot w$. Subsequently, to obtain the encrypted data c_i, we apply an exclusive-or operation (XOR) between I and K_0:

$$c_i = b_i \oplus k_i \ , \ 0 \leq i \leq s - 1 \ .$$

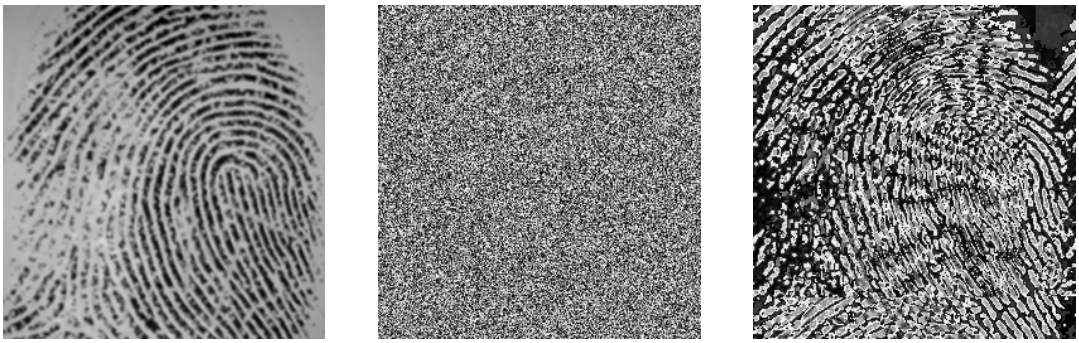
Since this operation only processes 1/8 of the binary representation of I, it is repeated for the remaining binary data of I 7 times using the identical key K_0. Finally, K_0 is encrypted using AES and transmitted to the receiver together with the encrypted data c_i, $0 \leq i \leq s-1$.

Compared to a full (i.e., 100%) AES encryption, the approach reduces the AES encryption effort to 12.5% and introduces only little additional overhead (XOR and extraction of the binary image data – compare Table 3 in Moon et al. (2006)).

14.2 VULNERABILITIES

14.2.1 *Key-length*

Encryption with simple XOR is only secure if the keystream is truly random and of the same length as the plaintext, i.e., if it is a one-time pad (e.g., Schneier (1996)). Both conditions are violated for the proposed approach. The key-length, which is an eighth of the message length in the proposed approach, gives an attacker the possibility to shift the ciphertext by the size of the key and XOR it with itself. This operation removes the key and leaves the attacker with the plaintext XORed with a version of itself that has been shifted by the key-length Schneier (1996). Figure 14.1 illustrates this for a fingerprint image. As can be seen, the image obtained by this operation yields a lot more information of the original fingerprint than the ciphertext.

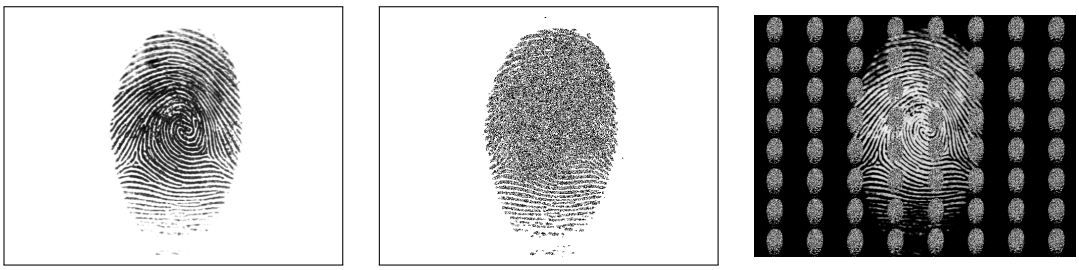


(a) Original image (b) Ciphertext C (c) $C \oplus (C >> key\_length)$

Figure 14.1: Illustration of shifting a ciphertext by key-length (part of DB4_1_8)

14.2.2 *Randomness*

Another problem lies with the assumption that for fingerprint sensors the LSB-plane is sufficiently random and not correlated with the other bitplanes. The authors of Moon et al. (2006) argue that the LSB-plane is "not correlated with other bitplanes if the images are acquired by various sensors such as a digital camera, scanner and other devices" and that the LSB plane "looks similar to a random number field". We have discussed both the randomness of the bitplanes and the correlation among the bitplanes in the previous chapter. Figure 14.2 shows another counter-example for the randomness of the LSB plane. The image is from database DB1 and has been acquired with an optical fingerprint sensor. As can be seen, the LSB does not generally behave like a random number field for all fingerprint sensors. The ciphertext – if the term is indeed appropriate in this case – for this fingerprint image is shown in Figure 14.2(c).



(a) Original image (b) LSB-plane (c) Ciphertext

Figure 14.2: Random LSB-plane: a counterexample (DB1_1_3)

14.2.3 *Key XORed with Itself*

Even if the LSB-plane produced by the used sensor is assumed to be sufficiently random, the scheme is not secure. The authors propose to XOR *all* bits of the plaintext with the LSB-plane. That means that the LSB-plane is XORed with itself at some positions. We show below that this is a fundamental problem. This critical mistake in the design of the encryption scheme could have easily been avoided, as we will discuss in Section 14.5.

14.2.4 *Data Expansion*

The ciphertext is 112.5% of the size of the plaintext. Since the transmission is intended for weak network links, this property is highly undesired. Similar to the issue of XORing

the key with itself, also data expansion could have been avoided, which will be discussed in Section 14.5 as well.

14.3 ATTACK

We attack the scheme at its most vulnerable point: the key being XORed with itself. Let C be the bits of the ciphertext that are obtained by encrypting I with key K_0. During encryption, each of the keybits (being the LSB of plaintext pixels) is XORed with an element of K_0. We subsume these elements as K_1. Note that $K_1 \subseteq K_0$. We introduce the operation $\hat{\oplus}$ with the meaning of $m \hat{\oplus} n$ as "the bit at position m gets encrypted by the bit at position n". We can conceive the operation as a "mapping" from K_0 to K_1, where

$$K_1 = \{k_i \in K_0 \mid \exists k_j \in K_0 : k_j \hat{\oplus} k_i\}.$$

If one or more of the keybits in K_0 are mapped to the same position, then $K_1 \subset K_0$, i.e., the mapping reduces the number of key positions. As the ciphertext is known, K_0 can be reconstructed from K_1, if the correct settings for the keybits in K_1 can be determined.

We can further investigate the mappings of the keybits in K_1. All of the elements of K_1 are mapped to an element of K_1. This can easily be shown: let k_j be an element of K_1, then also $k_j \in K_0$, because $K_1 \subseteq K_0$. If we now assume that k_j is mapped to $k_i \in K_0 \backslash K_1$, i.e., $k_j \hat{\oplus} k_i$, then by the definition of K_1 and because $k_j \in K_0$ and $k_i \in K_0$, it follows that $k_i \in K_1$, which contradicts the assumption. Therefore the set K_1 can be mapped to a set K_2 with $K_2 \subseteq K_1$.

This process can be applied repeatedly. We can map the keybits in K_i to a set K_{i+1}, $K_{i+1} \subseteq K_i$:

$$K_{i+1} = \{k_i \in K_i \mid \exists k_j \in K_i : k_j \hat{\oplus} k_i\}.$$

As long as one or more bits from K_{i+1} are mapped to the same bits in K_i, K_{i+1} is a proper subset of K_i: $K_{i+1} \subset K_i$, i.e., we reduce the number of referenced keybits. It can easily be seen that after a number of iterations N no more reduction is possible:

$$\exists N \geq 0 : K_{i+1} = K_i \text{ for } i \geq N.$$

If the correct settings for the bits in K_N are known, then K_{N-1} can be reconstructed. As generally the correct settings of K_{i+1} can be used to reconstruct K_i, the correct settings of the bits in K_N are sufficient to get the settings for all bits in the key.

It can be shown that for key-lengths of a power of 2, $|K_N| = 1$, i.e., the whole key depends on the setting of a single bit. In this case, the plaintext can be easily reconstructed by testing the two possible settings of this bit and then reconstructing the key. After decryption, one setting will yield the original plaintext, the other setting will yield the original plaintext with its pixels inverted.

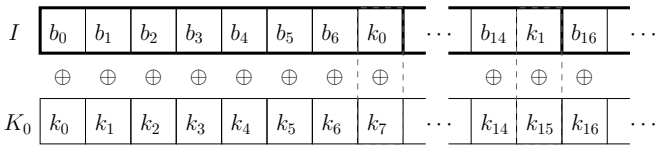
For images of other sizes, more positions will remain in the set K_N. Generally, K_N will be too large for a brute force search. Note that if s is coprime to 8, then $K_N = K_0$, i.e., no reduction is possible. However, the existing mappings of the bits in K_N can be used to further reduce the number of relevant keybits. The elements of K_N are either mapped to themselves or to another element of K_N. During the encryption process, only a limited number of elements are actually mapped to themselves. For the rest of the elements we can define circular chains of keybits. For the keybits in each of these sequences a mapping exists between each element and its successor, with the successor of the last element being the first element. E.g., for a key of length 73, k_{30} is mapped to k_{28} which is mapped to k_{12} which is mapped to k_{30} again. The elements of K_N which are mapped onto themselves form a chain of length 1. The number of chains that exist in K_N and their lengths depend on the length of the key.

If the correct setting for one bit in the chain is known, then the complete chain can be reconstructed. So we can choose a single element from each chain as a representative. Each of these representative elements influences a multitude of bit positions in the plaintext.

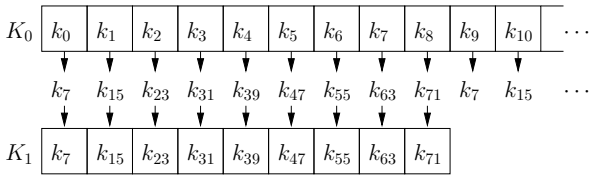
The procedure of reducing the keybits is illustrated in Figure 14.3 for an input image of 72 pixels. During encryption each element of K_0 is XORed with an element of K_0 (a). The referenced elements are collected in K_1 (b). For $n = 72$, no further reduction is possible after this step, so $N = 1$. In the final step, chains of mappings are found in K_1 (c). The number of representative bits for the 72-bit key is reduced to 5 bits.

As an example, we investigate fingerprint images used by Moon et al. (2006): for two different sensors, they obtain images of 320×440 and 248×292, respectively. For the first sensor this leads to a set $K_{N=3}$ with 275 elements. These elements can be grouped into 16 chains of varying length. During encryption, each chain influences between 4096 and 81920 bit positions in the plaintext. For the second sensor the reduced set of keybits $K_{N=2}$ has 2263 elements which can be organized into 175 chains. Each chain influences between 256 and 3840 bits in the plaintext.

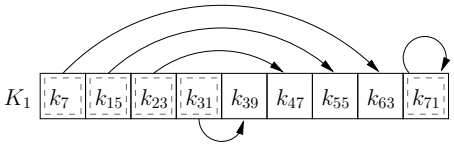
For a brute-force search, the number of chains is still too large. But we can formulate some conditions that should hold for the plaintext. Because the representative bits influence so many positions in the image, the condition needs not be overly sophisticated. For natural images, the sample variance can be used as a simple measure, for fingerprint images we introduce a more suitable measure below. A hypothesis for the value of the representative bit of each chain is formed and iteratively tested. We start by setting each representative bit to zero. Then the chains are reconstructed to form the set K_N. A hypothetical key is created by reconstructing K_{N-1} through K_0 from K_N. K_0 is used to reconstruct an image. In the next step one of the representative bits is flipped. Again, an image is reconstructed, and its sample variance is compared to the previous run. If the sample variance has decreased, then the bit is left at 1 otherwise it is flipped back to 0. This process is repeated over the whole set of representative bits,



(a) Mapping of keybits to keybits



(b) Construction of K_1 from K_0



(c) Construction of chains in K_1

Figure 14.3: Reduction of keybits for $n = 72$, $N = 1$

until the variance no longer changes. For images with a sufficient degree of smoothness the result will be the original image (or an inverted version of it, depending on the initial setting of the representative bits). This iterative refinement of a hypothetical key is similar to the method for cryptanalysis of substitution ciphers proposed by Jakobsen (1995).

This process is illustrated for a version of the DB2_3_3 fingerprint image of size 248 × 292 in Figure 14.4. Each image represents a whole run over the 175 chain bits. After run number 2 the variance does not change anymore and the image is found.

The variance for testing the hypothesis does not only work for most natural images but also for many of the tested fingerprint images. However, some of the fingerprint images exhibit strong oscillatory patterns. An example image, which was captured by a thermal sweeping sensor, is shown in Figure 14.5(a). In such cases, the minimum variance fails as a condition for the correct plaintext image, as shown in Figure 14.5(b). Therefore we use a more local measure that reflects the properties of fingerprint images in a better way: For each pixel in the image decrypted with the hypothetical key, we measure the difference of this pixel to all pixels surrounding it. The sum of these differences should be minimized. We found that considering the eight immediately

(a) Initial hypothesis (b) Run #1 (c) Run #2

Figure 14.4: Variance attack on DB2_3_3

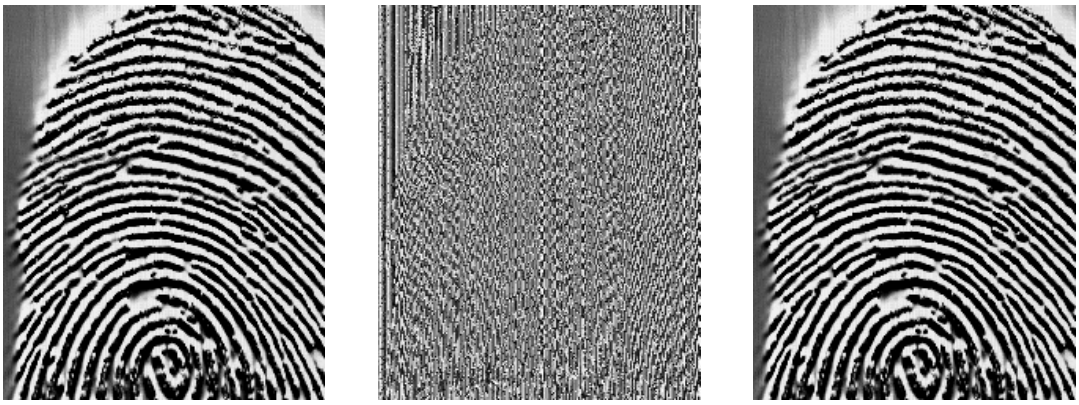
surrounding pixels is sufficient. With the neighborhood measure we can decrypt both, fingerprint and natural images. Figure 14.5(c) shows the attack result for Figure 14.5(a).

14.4 EVALUATION

If the proposed attack is successful for a given ciphertext, the decrypted image will generally be bitwise identical to the original image. The attack does not depend on the randomness of the LSB-plane: even for a truly random LSB-plane the encrypted image can be easily decrypted without knowing the key. We verified this by successfully attacking images for which the LSB was replaced by a pseudo-random number field. Furthermore, the attack is not restricted to fingerprint images, but also works for natural images. The attack can be adapted to work with any type of plaintext, if a suitable measure can be found to be used in the iteration for this type of plaintext.

There are some key-lengths that produce a large number of short chains, each of which only influences relatively few positions in the image. In these cases, the measures for the plaintext are too crude to produce a successful attack. To quantify the success rate of the proposed attack, we investigate the ratio of the number of chains to the number of bits in the key for image sizes ranging from 64^2 to 512^2. Figure 14.6 shows the ratio of key-lengths on the ordinate that achieve a certain ratio of number of chains to total key bits, given on the abscissa.

We found that the attack reliably produces the original plaintext for ratios that are below approximately 0.02. It can be seen that for 95.5% of the key-lengths the ratio lies below 0.01. 98.2% of the key-lengths lie below 0.02. That means that for 98.2% of the possible image sizes between 64^2 to 512^2 the proposed attack will work reliably



(a) Original image (b) Result of variance attack (c) Result of neighborhood attack

Figure 14.5: Variance versus neighborhood attack (part of DB3_84_2)

for natural and fingerprint images. For the rest, the reliability of the attack decreases with the number of representative bits. The time the attack needs increases with the number of bits. A more sophisticated measure could help to improve reliability in the range above 2%, at the expense of higher computational demands.

Note that the proposed attack cannot be transferred to Vigenère encryption in general (i.e., encryption with keystreams shorter than the plaintext where the keystream is not XORed with itself). Even for very short keys with a couple of hundred bits, for which each of the bits in the key influences many bits in the image, the iterative attack is unsuccessful. Figure 14.7 illustrates this for the Lena image: a 700-bit pseudo-random one-time pad is used for encryption (a), neither variance (b) nor local neighborhood (c) can retrieve the plaintext. A blocked version of the variance leads to slightly better results (d). The proposed attack fails, because it utilizes the fact that the reduction of the key results in an irregular influence of the representative bits on the bits in the image, which facilitates the use of very simple plaintext measures. For general Vigenère encryption, a more suitable and possibly more complex measure has to be used. Substitution ciphers with short keys have been shown to be easy to crack Schneier (1996), especially for plaintexts with low entropy like natural language texts and images.

The computational demands of the attack depend on the size of the image and the number of representative bits. Table 14.1 shows some timing results, which were obtained with a Java implementation running on an AMD Athlon 1.6 GHz with 2 GB of RAM. It can be seen that the costs for the attack are low. Using the sample variance for an attack on DB3_84_2 was unsuccessful (marked by † in the table), all other attacks produced the original plaintext image. Some of the images were cropped to a certain

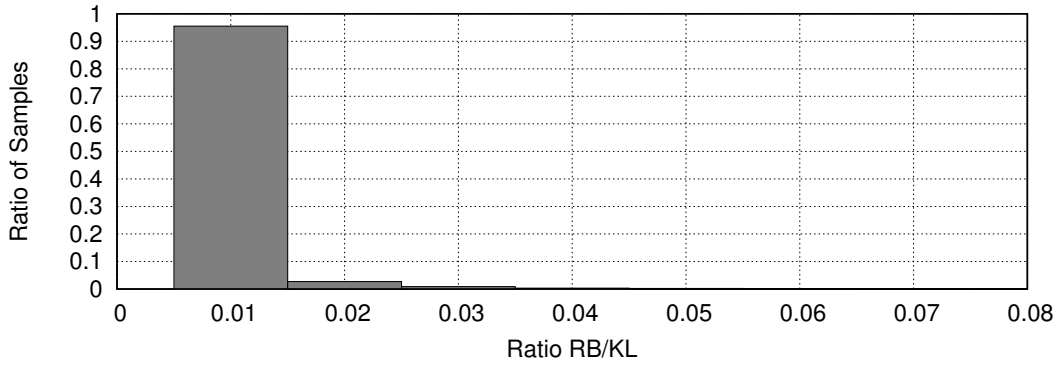


Figure 14.6: Ratio of samples by ratio of (representative bits)/(total keybits)

size (marked by $\star$) to reflect different key-lengths (and to remove the scanner background in some cases). The last image represents a special case for which the keybits can be reduced to a single representative bit. In this case, the attack is extremely fast and no measure is needed.

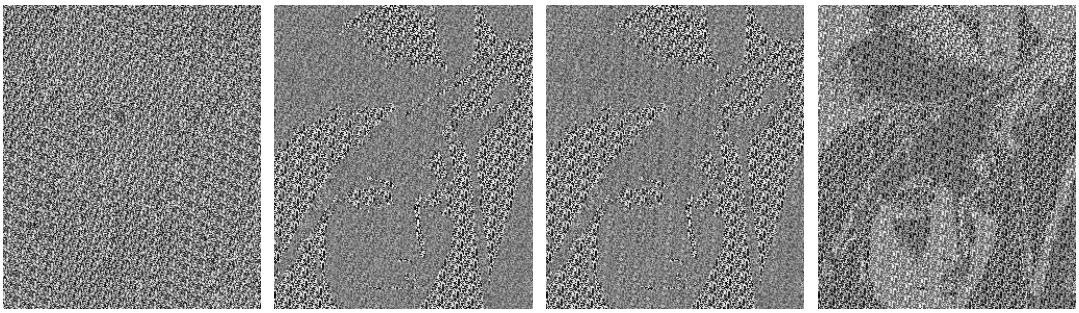
14.5 IMPROVEMENTS

14.5.1 *Key XORed with Itself*

In this respect, the scheme can be designed in a more secure way: only XOR the bitplanes apart from the LSB-plane with the LSB-key, i.e., bitplanes 7 through 1, but leave the LSB untouched. The original scheme proposes to encrypt the LSB-plane with AES anyway. The encrypted version can be inserted into the ciphertext at the LSB positions. Apart from enhancing security by avoiding the key being XORed with itself, this modification brings another advantage: unlike in the original scheme, the LSB-plane information is not transmitted twice, therefore also solving the *data expansion* problem.

14.5.2 *Randomness*

In order to produce a keystream that exhibits more properties of a random number field, we suggest to extract the LSB-plane first (or any other bitplane), subject it to AES encryption, and finally use the resulting data as the keystream for the XOR operation. Of course, AES ciphertext is not truly random, but at least it passes several strong statistical tests for randomness (Hellekalek and Wegenkittl, 2003). This proce-



(a) Ciphertext (b) Variance (c) Neighborhood (d) Blocked Variance

Figure 14.7: Unsuccessful attacks for general Vigenère encryption

dure also invalidates the proposed attack. It has to be noted, however, that with this approach, the encrypted LSB-plane has to be regarded as proper key material and has to be transferred over a secure channel.

14.5.3 *Key-length*

The length of the key remains restricted to 1/7 of the data size even if implementing the improvements as suggested so far. This is a major obstacle. A possible solution would be to additionally introduce 6 different permutations of the key data at the cost of additional key material. It is doubtful (and of course depends on the type of permutations applied) if such a scheme would still be more efficient and equally secure as compared to full encryption with a fast stream cipher (e.g., Rose and Hawkes (2003); Ekdahl and Johansson (2003); Halevi et al. (2002)).

14.6 CONCLUSION

We have analyzed a published encryption scheme for fingerprint images. With the computationally undemanding attack we presented, access to the full plaintext can be obtained for most given ciphertexts. We have shown that although some improvements to the original scheme are possible, it remains insecure.

Simplistic schemes used to secure fingerprint image data may be a severe threat to the security of the biometric data. It has to be pointed out that fundamental knowledge in the cryptographic area has to be obeyed as well, when designing lightweight encryption schemes.

Image	Size	R. Bits	Variance	Neighborhood
DB1_1_3	640 × 480	8	13.4 s	7.5 s
DB4_1_8	233 × 384	8	4.4 s	4.8 s
DB3_84_2	300 × 480	32	†	14.8 s
DB3_84_2*	205 × 251	527	†	96.9 s
DB2_3_3	328 × 364	210	67 s	107 s
DB2_3_3*	248 × 292	175	52 s	54 s
DB2_3_3*	187 × 279	1556	433.2 s	529.3 s
DB2_3_3*	256 × 256	1	1.2 s	

Table 14.1: Timing results

EPILOGUE

15 CONCLUSION

We proposed, discussed, and evaluated a number of different methods for encryption of still visual data. Many of the proposed approaches present alternatives to full encryption. It should be stated one more – final – time that all of the approaches trade in security for something else. Therefore, in terms of security, the most secure option remains full encryption. However, if functionality is desired rather than the highest possible security, then the proposed approaches can be used to cover a large variety of different functionalities.

In the area of compression-integrated methods, we discussed two families of approaches that use secret transform domains: parameterized wavelet filters and key-dependent subband structures in the form of randomized wavelet packet decomposition structures. We presented an attack against encryption approaches that are based on filter parameterization. This attack makes encryption methods that are based on a linear transform in general only suitable for very lightweight encryption. In the context of wavelet packets, the security is much higher. We have shown that the negative effect of randomized wavelet packets on compression performance can be kept at a minimum by introducing constraining parameters. We have also shown that because the anisotropic wavelet packet transform provides a higher degree of freedom than the isotropic wavelet packet transform, higher security can be achieved at a lower price (in terms of loss in quality). We discussed possible attacks on the wavelet-packet-based encryption schemes and have shown that configurations exist that leak information on the decomposition structure through the JPEG2000 packet header. The header protection scheme we proposed can be used to restore security for wavelet-packet-based encryption methods. In the discussion of application scenarios for key-dependent subband structures we have shown that because so little information needs to be encrypted, there are diverse scenarios in which the encryption method can be used.

The second major focus of this thesis has been the analysis of existing schemes and fixing discovered security flaws where possible. In the context of format-compliant JPEG2000 encryption, we have been able to show that the encryption of JPEG2000 packet bodies alone, i.e., leaving the packet headers in plaintext, is not secure for applications that require full confidentiality. We proposed a header protection scheme that

solves this problem. We also discussed a method for transparent encryption based on the header protection scheme.

In the last part of the thesis we turned to the security of fingerprint images and the analysis of existing schemes in this context. We first discussed properties of the bitplanes of fingerprint images and have shown that, contrary to some publications, the lowest bitplanes do not generally exhibit random properties. We then presented an attack against a previously published lightweight encryption scheme for fingerprint images. The presented attack is independent of the properties of the bitplanes and would work even if the least significant bitplane were truly random. Although some of the security flaws of the encryption scheme can be fixed, overall the approach has been shown to be beyond repair.

In this thesis we covered different fields of the large area of media encryption. It is natural that some open questions remain. In the context of header encryption, more methods could and should be developed, possibly methods that are not based on permutation. In compression-integrated encryption, different parameterized transform domains could be investigated and evaluated with regard to their compression performance and security. The attack presented against parameterized wavelet filters should be implemented efficiently and tested empirically against a large number of images. In the context of key-dependent wavelet packet subband structures, the extent of information leakage that is present in the JPEG2000 headers should be investigated in more detail and quantified. An implementation of the attack could prove insightful. In the context of the security of biometric data, further analysis of the statistical properties of fingerprint images could be conducted.

On a more general note, all of the topics covered provide much room for future research. In the context of JPEG2000 encryption, flexible encryption schemes that can be adapted to the needs of an application will become more important. From the practical point of view, the detailed investigation of application scenarios in which transparent encryption is employed is still missing. Applications like digital libraries are likely to profit from transparent encryption techniques. With the further dissemination of mobile devices with diverse display capabilities, format-compliant and scalable encryption of visual content will increase in importance. JPEG2000 as a standard for scalable still image coding is a promising starting point in this context, and security techniques tailored to specific applications should be investigated.

Many of the techniques discussed here can be transferred and adapted to video encryption. There has been a lot of research in the area of scalable video coding recently. Therefore, a promising research area will be the development of reliable bitstream-oriented approaches for the scalable extension of H.264/AVC, H.264/SVC (Schwarz et al., 2006, 2007). Proposals for security techniques for H.264/SVC have recently begun to appear more numerously, see, e.g., Magli et al. (2008).

The use of secret transform domains for video encryption should be assessed in the context of wavelet-based scalable video coding. A good overview of scalable video coding is given by Ohm (2005). A number of proposals and implementations exist, for each of which security techniques need to be designed. For instance, André et al. (2007) propose a video codec based on JPEG2000 that uses motion-compensated temporal filtering (MCTF) and report very competitive compression results. A less sophisticated approach for MCTF with JPEG2000 is described in Eder et al. (2007). A wavelet-based codec for which the compression-integrated methods should also be investigated is MC-EZBC (Hsiang and Woods, 2001).

Media encryption is a young and thriving topic with many possible directions for future research, of which we have covered only a small selection. Many more research directions will come up in the future. They will on the one hand entail the continued investigations of specialized topics, while on the other hand they will have to look at the bigger picture and research the integration and the interplay of different areas of security, e.g., encryption, watermarking, perceptual hashing, and fingerprinting, in practical systems and also take legal and social issues into account.

APPENDIX

BIBLIOGRAPHY

T. Acharya and A. K. Ray. *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*. Wiley, New Jersey, 2004. (Cited on page 19.)

T. André, M. Cagnazzo, M. Antonini, and M. Barlaud. JPEG2000-compatible scalable scheme for wavelet-based video coding. *EURASIP Journal on Image and Video Processing*, 2007. (Cited on page 195.)

B. Bhargava, C. Shi, and Y. Wang. MPEG video encryption algorithms. *Multimedia Tools and Applications*, 24(1):57–79, 2004. (Cited on page 6.)

M. Brachtl, W. M. Dietl, and A. Uhl. Key-dependency for a wavelet-based blind watermarking algorithm. In J. Dittmann and J. Fridrich, editors, *ACM Multimedia and Security Workshop*, pages 175–179, Magdeburg, Germany, Sept. 2004. (Cited on pages 66 and 107.)

J. N. Bradley, C. M. Brislawn, and T. Hopper. The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression. In *SPIE Proceedings, Visual Information Processing II*, volume 1961, pages 293–304, Orlando, FL, USA, Apr. 1993. (Cited on page 123.)

R. Buckley. JPEG 2000 – a practical digital preservation standard? Technical Report 708-01, DPC Technology Watch Series, Feb. 2008. URL http://www.dpconline.org/docs/reports/dpctw08-01.pdf. (Cited on pages 21 and 22.)

J. But. Limitations of existing MPEG-1 ciphers for streaming video. Technical Report CAIA 040429A, Swinburne University, Australia, Apr. 2004. URL http://caia.swin.edu.au/cv/jbut/publications.html. (Cited on page 6.)

M. Cancellaro, M. Carli, K. Egiazarian, and J. Astola. Perceptual data hiding in tree structured haar transform domain. In E. J. Delp and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, Proceedings of SPIE, pages 65051Q1–65051S11, San Jose, CA, USA, Jan. 2007. SPIE. (Cited on page 12.)

M. Caramma, R. C. Lancini, and M. Marconi. A perceptual PSNR based on the utilization of a linear model of HVS, motion vectors and DFT-3d. In *Proceedings*

of the 10th European Signal Processing Conference, EUSIPCO '00, Tampere, Finland, Oct. 2000. (Cited on page 16.)

M. Carnec, P. L. Callet, and D. Barba. A new method for perceptual quality assessment of compressed images with reduced reference. In *Picture Coding Symposium 2003 (PCS'03)*, pages 343–348, Saint Malo, France, Apr. 2003. (Cited on page 16.)

C. Christopoulos, A. N. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system: an overwiew. *IEEE Transactions on Consumer Electronics*, 46 (4):1103–1127, Nov. 2000. (Cited on page 19.)

A. Cohen, I. Daubechies, and J. Feauveau. Bi-orthogonal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 45:485–560, 1992. (Cited on page 66.)

V. Conan, Y. Sadourny, K. Jean-Marie, C. Chan, S. Wee, and J. Apostolopoulos. Study and validation of tools interoperability in JPSEC. In A. G. Tescher, editor, *Applications of Digital Image Processing XXVIII*, volume 5909, page 59090H. SPIE, 2005. (Cited on page 10.)

I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis Applications*, 4(3):245–267, 1998. (Cited on pages 65, 66, 67, and 92.)

E. J. Delp. Multimedia security: the 22nd century approach. *Multimedia Systems*, 11(2):95–97, Dec. 2005. (Cited on page 5.)

R. H. Deng, W. S. Di Ma, and Y. Wu. Scalable trusted online dissemination of JPEG2000 images. *Multimedia Systems*, 11(1):60 – 67, Nov. 2005. (Cited on page 10.)

A. Descampe, P. Vandergheynst, C. D. Vleeschouwer, and B. Macq. Coarse-to-fine textures retrieval in the JPEG 2000 compressed domain for fast browsing of large image databases. In B. Günsel, A. K. Jain, A. M. Tekalp, and B. Sankur, editors, *Proc. Multimedia Content Representation, Classification and Security, MRCS 2006*, volume 4105 of *Lecture Notes in Computer Science*, pages 282–289, Berlin, Heidelberg, New York, Tokyo, Sept. 2006. Springer-Verlag. (Cited on pages 32, 46, and 56.)

W. Dietl and A. Uhl. Watermark security via secret wavelet packet subband structures. In A. Lioy and D. Mazzocchi, editors, *Communications and Multimedia Security. Proceedings of the Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, volume 2828 of *Lecture Notes on Computer*

Science, pages 214–225, Turin, Italy, Oct. 2003. Springer-Verlag. (Cited on pages 107 and 171.)

W. Dietl, P. Meerwald, and A. Uhl. Watermark security via high-resolution wavelet filter parametrization. In K. Stanislav and P. Miron, editors, *Proceedings of 7th International Scientific Conference, Section 1: Applied Mathematics*, pages 21–28, Košice, Slovakia, May 2002. (Cited on page 66.)

W. Dietl, P. Meerwald, and A. Uhl. Key-dependent pyramidal wavelet domains for secure watermark embedding. In E. J. Delp and P. W. Wong, editors, *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, pages 728–739, Santa Clara, CA, USA, Jan. 2003a. SPIE. (Cited on page 66.)

W. Dietl, P. Meerwald, and A. Uhl. Protection of wavelet-based watermarking systems using filter parametrization. *Signal Processing (Special Issue on Security of Data Hiding Technologies)*, 83:2095–2116, 2003b. (Cited on page 66.)

W. M. Dietl and A. Uhl. Robustness against unauthorized watermark removal attacks via key-dependent wavelet packet subband structures. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004. (Cited on pages 107 and 171.)

Digital Cinema Initiatives (DCI), LLC. Digital cinema system specification v1.1, Apr. 2007. URL http://www.dcimovies.com. (Cited on page 21.)

J. Dittmann and R. Steinmetz. Enabling technology for the trading of MPEG-encoded video. In *Information Security and Privacy: Second Australasian Conference, ACISP '97*, volume 1270, pages 314–324, July 1997a. (Cited on page 9.)

J. Dittmann and R. Steinmetz. A technical approach to the transparent encryption of MPEG-2 video. In S. K. Katsikas, editor, *Communications and Multimedia Security, IFIP TC6/TC11 Third Joint Working Conference, CMS '97*, pages 215–226, Athens, Greece, Sept. 1997b. Chapman and Hall. (Cited on page 9.)

J. Dittmann, K. Nahrstedt, and P. Wohlmacher. Approaches to multimedia and security. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '00*, pages 1275–1278, New York, NY, USA, July 2000. (Cited on page 5.)

J. Dittmann, C. Vielhauer, and J. Hansen, editors. *New Advances in Multimedia Security, Biometrics, Watermarking and Cultural Aspects*. Logos Verlag Berlin, 2006. (Cited on page 5.)

I. Djurovic, S. Stankovic, and I. Pitas. Digital watermarking in the fractional fourier transformation domain. *Journal of Network and Computer Applications*, 24:167–173, 2001. (Cited on page 12.)

M. V. Droogenbroeck. Partial encryption of images for real-time applications. In *Proceedings of the 4th 2004 Benelux Signal Processing Symposium*, pages 11–15, Hilvarenbeek, The Netherlands, Apr. 2004. (Cited on page 9.)

M. V. Droogenbroeck and R. Benedett. Techniques for a selective encryption of uncompressed and compressed images. In *Proceedings of ACIVS (Advanced Concepts for Intelligent Vision Systems)*, pages 90–97, Ghent University, Belgium, Sept. 2002a. (Cited on page 9.)

M. V. Droogenbroeck and R. Benedett. Techniques for a selective encryption of uncompressed and compressed images. In *Proceedings of ACIVS (Advanced Concepts for Intelligent Vision Systems)*, pages 90–97, Ghent University, Belgium, Sept. 2002b. (Cited on page 176.)

F. Dufaux and T. Ebrahimi. Securing JPEG2000 compressed images. In A. G. Tescher, editor, *Applications of Digital Image Processing XXVI*, volume 5203, pages 397–406. SPIE, 2003. (Cited on page 10.)

P. Eder, D. Engel, and A. Uhl. Jpeg2000-based scalable video coding with mctf. Technical Report 2007-04, Department of Computer Sciences, University of Salzburg, Austria, Oct. 2007. URL http://www.cosy.sbg.ac.at/research/tr/2007-04_Eder_Engel_Uhl.pdf. (Cited on page 195.)

P. Ekdahl and T. Johansson. *A New Version of the Stream Cipher SNOW*, volume 2595 of *LNCS*, pages 47–61. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 2003. (Cited on page 188.)

D. Engel and A. Uhl. Adaptive object-based image compression using wavelet packets. In M. Grgic, editor, *Proceedings of the 4th International Symposium on Video/Image Processing and Multimedia Communications (VIPromCom 2002)*, pages 183–187, Zadar, Croatia, June 2002. (Cited on page 106.)

D. Engel and A. Uhl. Adaptive image compression of arbitrarily shaped objects using wavelet packets. In *Picture Coding Symposium 2003 (PCS'03)*, pages 283–288, Saint Malo, France, Apr. 2003. (Cited on page 106.)

D. Engel and A. Uhl. Parameterized biorthogonal wavelet lifting for lightweight JPEG2000 transparent encryption. In *Proceedings of ACM Multimedia and Security*

Workshop, MM-SEC '05, pages 63–70, New York, NY, USA, Aug. 2005a. (Cited on pages 12 and 65.)

D. Engel and A. Uhl. Security enhancement for lightweight JPEG2000 transparent encryption. In *Proceedings of Fifth International Conference on Information, Communication and Signal Processing, ICICS '05*, pages 1102–1106, Bangkok, Thailand, Dec. 2005b. (Cited on pages 12 and 81.)

D. Engel and A. Uhl. Secret wavelet packet decompositions for JPEG2000 lightweight encryption. In *Proceedings of 31st International Conference on Acoustics, Speech, and Signal Processing, ICASSP '06*, volume V, pages 465–468, Toulouse, France, May 2006a. IEEE. (Cited on pages 12 and 152.)

D. Engel and A. Uhl. Lightweight JPEG2000 encryption with anisotropic wavelet packets. In *Proceedings of International Conference on Multimedia & Expo, ICME '06*, pages 2177–2180, Toronto, Canada, July 2006b. IEEE. (Cited on pages 12 and 113.)

D. Engel and A. Uhl. An evaluation of lightweight JPEG2000 encryption with anisotropic wavelet packets. In E. J. Delp and P. W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, Proceedings of SPIE, pages 65051S1–65051S10, San Jose, CA, USA, Jan. 2007a. SPIE. (Cited on pages 12, 113, 119, and 155.)

D. Engel and A. Uhl. An attack against image-based selective bitplane encryption. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '07*, volume II, pages 141–144, San Antonio, TX, USA, Sept. 2007b. IEEE. (Cited on page 175.)

D. Engel, R. Kutil, and A. Uhl. A symbolic transform attack on lightweight encryption based on wavelet filter parameterization. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '06*, pages 202–207, Geneva, Switzerland, Sept. 2006. (Cited on page 91.)

D. Engel, T. Stütz, and A. Uhl. Format-compliant JPEG2000 encryption in JPSEC: Security, applicability and the impact of compression parameters. *EURASIP Journal on Information Security*, (Article ID 94565), 2007a. (Cited on pages 8, 26, and 31.)

D. Engel, T. Stütz, and A. Uhl. Format-compliant JPEG2000 encryption with combined packet header and packet body protection. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '07*, pages 87–95, New York, NY, USA, Sept. 2007b. ACM Press. (Cited on pages 31 and 49.)

D. Engel, T. Stütz, and A. Uhl. Efficient transparent JPEG2000 encryption with format-compliant header protection. In *Proceedings of IEEE International Conference on Signal Processing and Communications, ICSPC '07*, pages 1067–1070, Dubai, UAE, Nov. 2007c. (Cited on page 49.)

D. Engel, E. Pschernig, and A. Uhl. An analysis of lightweight encryption schemes for fingerprint images. *IEEE Transactions on Information Forensics and Security*, 3(2):173–182, June 2008a. (Cited on pages 175 and 176.)

D. Engel, T. Stütz, and A. Uhl. Efficient transparent JPEG2000 encryption. In C.-T. Li, editor, *Multimedia Forensics and Security*, chapter 16. IGI Global, Hershey, PA, USA, 2008b. to appear. (Cited on page 10.)

A. Eskicioglu, P. Fisher, and S. Chen. Image quality measures and their performance. Technical report, University of North Texas, Department of Computer Sciences, 1993. (Cited on page 15.)

M. M. Fisch, H. Stögner, and A. Uhl. Layered encryption techniques for DCT-coded visual data. In *Proceedings (CD-ROM) of the European Signal Processing Conference, EUSIPCO '04*, Vienna, Austria, Sept. 2004. paper cr1361. (Cited on page 9.)

D. H. Foos, E. Muka, R. M. Slone, B. J. Erickson, M. J. Flynn, D. A. Clunie, L. Hildebrand, K. S. Kohm, and S. S. Young. JPEG 2000 compression of medical imagery. In G. J. Blaine and E. L. Siegel, editors, *Medical Imaging 2000: PACS Design and Evaluation: Engineering and Clinical Issues*, volume 3980 of *Proc. of SPIE*, pages 85–96. SPIE, 2000. (Cited on page 21.)

J. Fridrich. Key-dependent random image transforms and their applications in image watermarking. In *Proceedings of the 1999 International Conference on Imaging Science, Systems, and Technology, CISST '99*, pages 237–243, Las Vegas, NV, USA, June 1999. (Cited on page 12.)

J. Fridrich, A. C. Baldoza, and R. J. Simard. Robust digital watermarking based on key-dependent basis functions. In D. Aucsmith, editor, *Information hiding: second international workshop*, volume 1525 of *Lecture notes in computer science*, pages 143–157, Portland, OR, USA, Apr. 1998. Springer Verlag, Berlin, Germany. (Cited on page 12.)

B. Furht and D. Kirovski, editors. *Multimedia Security Handbook*. CRC Press, Boca Raton, Florida, 2005. (Cited on page 5.)

M. Grangetto, E. Magli, and G. Olmo. Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, 8 (5):905–917, 2006. (Cited on pages 10 and 12.)

R. Grosbois, P. Gerbelot, and T. Ebrahimi. Authentication and access control in the JPEG2000 compressed domain. In A. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, pages 95–104, San Diego, CA, USA, July 2001. (Cited on pages 7 and 10.)

S. Halevi, D. Coppersmith, and C. Jutla. *Scream: A Software-Efficient Stream Cipher*, volume 2365 of *LNCS*, pages 195–209. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 2002. (Cited on page 188.)

F. Hartenstein. Parametrization of discrete finite biorthogonal wavelets with linear phase. In *Proceedings of the 1997 International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, Apr. 1997. (Cited on page 66.)

P. Hellekalek and S. Wegenkittl. Empirical evidence concerning AES. *ACM Trans. Model. Comput. Simul.*, 13(4):288–302, 2003. (Cited on page 187.)

T. Hopper. Compression of gray-scale fingerprint images. In H. Szu, editor, *Wavelet Applications*, volume 2242 of *SPIE Proceedings*, pages 180–187, San Diego, CA, Mar. 1994. (Cited on page 106.)

S.-T. Hsiang and J. W. Woods. Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank. *Signal Processing: Image Communication*, 16(8):705–724, May 2001. (Cited on page 195.)

J. Huang, J. Hu, D. Huang, and Y. Q. Shi. Improve security of fragile watermarking via parameterized wavelet. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapore, Oct. 2004. IEEE Signal Processing Society. (Cited on pages 66 and 69.)

ISO/IEC 15444-1. Information technology – JPEG2000 image coding system, Part 1: Core coding system, Dec. 2000. (Cited on pages 11, 19, 22, and 36.)

ISO/IEC 15444-2. Information technology – JPEG2000 image coding system, Part 2: Extensions, May 2004. (Cited on pages 24 and 107.)

ISO/IEC 15444-4. Information technology – JPEG2000 image coding system, Part 4: Conformance testing, Dec. 2004. (Cited on page 24.)

ISO/IEC 15444-8. Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000, Apr. 2007. (Cited on pages 24 and 31.)

ITU-T T.800. Information technology – JPEG2000 image coding system, Part 1: Core coding system, Aug. 2002. (Cited on pages 19 and 22.)

ITU-T T.801. Information technology – JPEG2000 image coding system, Part 2: Extensions, Feb. 2002. (Cited on page 24.)

ITU-T T.803. Information technology – JPEG2000 image coding system, Part 4: Conformance testing, Nov. 2004. (Cited on page 24.)

ITU-T T.807. Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000, Mar. 2006. (Cited on page 24.)

T. Jakobsen. A fast method for the cryptanalysis of substitution ciphers. *Cryptologia*, 19(3):265–274, 1995. (Cited on page 184.)

H. Kiya, D. Imaizumi, and O. Watanabe. Partial-scrambling of image encoded using JPEG2000 without generating marker codes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, volume III, pages 205–208, Barcelona, Spain, Sept. 2003. (Cited on page 10.)

T. Köckerbauer, M. Kumar, and A. Uhl. Lightweight JPEG2000 confidentiality for mobile environments. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004. (Cited on pages 12 and 65.)

T. Kunkelmann. Applying encryption to video communication. In *Proceedings of the Multimedia and Security Workshop at ACM Multimedia '98*, pages 41–47, Bristol, England, Sept. 1998. (Cited on pages 6 and 9.)

T. Kunkelmann and U. Horn. Partial video encryption based on scalable coding. In 5th *International Workshop on Systems, Signals and Image Processing (IWSSIP'98)*, pages 215–218, Zagreb, Croatia, 1998. (Cited on page 9.)

R. Kutil. A significance map based adaptive wavelet zerotree codec (SMAWZ). In S. Panchanathan, V. Bove, and S. Sudharsanan, editors, *Media Processors 2002*, volume 4674 of *SPIE Proceedings*, pages 61–71, Jan. 2002. (Cited on page 106.)

R. Kutil. Zerotree image compression using anisotropic wavelet packet transform. In T. Ebrahimi and T. Sikora, editors, *Visual Communications and Image Processing 2003 (VCIP'03)*, volume 5150 of *SPIE Proceedings*, pages 1417–1427, Lugano, Switzerland, July 2003a. SPIE. (Cited on page 113.)

R. Kutil. Anisotropic 3-D wavelet packet bases for video coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, Barcelona, Spain, Sept. 2003b. IEEE. (Cited on page 113.)

R. Kutil and D. Engel. Methods for the anisotropic wavelet packet transform. *Applied and Computational Harmonic Analysis*, 2008. to appear. (Cited on pages 113, 114, 119, and 121.)

M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In P. W. Wong and E. J. Delp, editors, *Proceedings of the 11th SPIE Annual Symposium, Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, pages 226–239, San Jose, CA, USA, Jan. 1999. (Cited on pages 14 and 15.)

G. Laimer and A. Uhl. Improving security of JPEG2000-based robust hashing using key-dependent wavelet packet subband structures. In P. Dondon, V. Mladenov, S. Impedovo, and S. Cepisca, editors, *Proceedings of the 7th WSEAS International Conference on Wavelet Analysis & Multirate Systems (WAMUS'07)*, pages 127–132, Arcachon, France, Oct. 2007. (Cited on page 107.)

G. Laimer and A. Uhl. Key dependent JPEG2000-based robust hashing for secure image authentication. *EURASIP Journal on Information Security*, (Article ID 895174):doi:10.1155/2008/895174, 19 pages, 2008. (Cited on page 66.)

C. Lambrecht and O. Verscheure. Perceptual quality measure using a spatio-temporal model of the human visual system. In *Proceedings of SPIE*, volume 2668, pages 450–461, Jan. 1996. (Cited on page 15.)

A. Langley and D. S. Bloomberg. Google Books: making the public domain universally accessible. In X. Lin and B. A. Yanikoglu, editors, *Proceedings of SPIE, Document Recognition and Retrieval XIV*, volume 6500, San Jose, CA, USA, Jan. 2007. to appear. (Cited on page 21.)

C. Liu and M. Mandal. Fast image indexing based on JPEG2000 packet header. In *Proceedings of the 2001 ACM workshops on Multimedia: Multimedia Information Retrieval*, pages 46–49, New York, NY, USA, 2001. ACM Press. (Cited on pages 32, 46, and 56.)

J.-L. Liu. Efficient selective encryption for jpeg 2000 images using private initial table. *Pattern Recogn.*, 39(8):1509–1517, 2006. ISSN 0031-3203. (Cited on pages 10 and 12.)

T. D. Lookabaugh, D. C. Sicker, D. M. Keaton, W. Y. Guo, and I. Vedula. Security analysis of selectiveley encrypted MPEG-2 streams. In *Multimedia Systems and Applications VI*, volume 5241 of *Proceedings of SPIE*, pages 10–21, Sept. 2003. (Cited on page 6.)

X. Lu and A. M. Eskicioglu. Selective encryption of multimedia content in distribution networks: Challenges and new directions. In *Proceedings of the IASTED International Conference on on Communications, Internet and Information Technology (CIIT 2003)*, Scottsdale, AZ, USA, Nov. 2003. (Cited on page 6.)

B. Macq and J. Quisquater. Digital images multiresolution encryption. *The Journal of the Interactive Multimedia Association Intellectual Property Project*, 1(1): 179–206, Jan. 1994. (Cited on page 9.)

B. M. Macq and J.-J. Quisquater. Cryptology for digital TV broadcasting. *Proceedings of the IEEE*, 83(6):944–957, June 1995. (Cited on pages 9 and 99.)

E. Magli, M. Grangetto, and G. Olmo. Conditional access techniques for H.264/AVC and H.264/SVC compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008. to appear. (Cited on page 194.)

D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Verlag, 2003. (Cited on page 175.)

Y. Mao and M. Wu. Security evaluation for communication-friendly encryption of multimedia. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapore, Oct. 2004. IEEE Signal Processing Society. (Cited on pages 16 and 50.)

D. Marpe, H. Cycon, and W. Li. Complexity constrained best-basis wavelet packet algorithm for image compression. *IEE Proceedings Vision, Image, and Signal Processing*, 145(6):391–398, 1998. (Cited on page 106.)

M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998. (Cited on page 177.)

P. Meerwald and A. Uhl. Watermark security via wavelet filter parametrization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, volume 3, pages 1027–1030, Thessaloniki, Greece, Oct. 2001. IEEE Signal Processing Society. (Cited on page 66.)

A. Meixner and A. Uhl. Security enhancement of visual hashes through key dependent wavelet transformations. In F. Roli and S. Vitulano, editors, *Image Analysis and Processing - ICIAP 2005*, volume 3617 of *Lecture Notes on Computer Science*, pages 543–550, Cagliari, Italy, Sept. 2005. Springer-Verlag. (Cited on page 66.)

F. G. Meyer, A. Z. Averbuch, and J.-O. Strömberg. Fast adaptive wavelet packet image compression. *IEEE Trans. on Image Process.*, 9(5):792–800, May 2000. (Cited on page 106.)

A. Moffat, R. M. Neal, and I. H. Witten. Arithmetic coding revisited. *ACM Trans. Inf. Syst.*, 16(3):256–294, 1998. (Cited on page 177.)

D. Moon, Y. Chung, S. B. Pan, K. Moon, and K. I. Chung. An efficient selective encryption of fingerprint images for embedded processors. *ETRI Journal*, 28(4):444–452, Aug. 2006. (Cited on pages 175, 176, 179, 180, 181, and 183.)

R. Norcen and A. Uhl. Selective encryption of the JPEG2000 bitstream. In A. Lioy and D. Mazzocchi, editors, *Communications and Multimedia Security. Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, CMS '03*, volume 2828 of *Lecture Notes on Computer Science*, pages 194 – 204, Turin, Italy, Oct. 2003. Springer-Verlag. (Cited on pages 7, 10, 50, and 162.)

R. Norcen and A. Uhl. Encryption of wavelet-coded imagery using random permutations. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapore, Oct. 2004a. IEEE Signal Processing Society. (Cited on pages 7, 12, and 22.)

R. Norcen and A. Uhl. Robust visual hashing using JPEG2000. In D. Chadwick and B. Preneel, editors, *Eighth IFIP TC6/TC11 Conference on Communications and Multimedia Security (CMS'04)*, pages 223–236, Lake Windermere, GB, Sept. 2004b. Springer-Verlag. (Cited on page 56.)

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*, 33(3):277 – 292, 2003. (Cited on page 21.)

J.-R. Ohm. Advances in scalable video coding. *Proceedings of the IEEE*, 93(1):42–56, 2005. (Cited on page 195.)

Y. Ou, C. Sur, and K. H. Rhee. Region-based selective encryption for medical imaging. In *Proceedings of the International on Frontiers in Algorithmics*, Lecture Notes in Computer Science, pages 62–73, Lanzhou, China, Aug. 2007. Springer-Verlag. (Cited on page 7.)

A. Pommer and A. Uhl. Wavelet packet methods for multimedia compression and encryption. In *Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 1–4, Victoria, Canada, Aug. 2001. IEEE Signal Processing Society. (Cited on pages 12 and 74.)

A. Pommer and A. Uhl. Selective encryption of wavelet packet subband structures for secure transmission of visual data. In J. Dittmann, J. Fridrich, and P. Wohlmacher, editors, *Multimedia and Security Workshop, ACM Multimedia*, pages 67–70, Juan-les-Pins, France, Dec. 2002. (Cited on pages 12 and 107.)

A. Pommer and A. Uhl. Selective encryption of wavelet-packet encoded image data — efficiency and security. *ACM Multimedia Systems (Special issue on Multimedia Security)*, 9(3):279–287, 2003. (Cited on pages 12, 78, 97, 107, 108, 110, 111, 127, 146, and 165.)

L. Qiao and K. Nahrstedt. Comparison of MPEG encryption algorithms. *International Journal on Computers and Graphics (Special Issue on Data Security in Image Communication and Networks)*, 22(3):437–444, 1998. (Cited on page 6.)

N. M. Rajpoot, R. G. Wilson, F. G. Meyer, and R. R. Coifman. A new basis selection paradigm for wavelet packet image coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, pages 816–819, Thessaloniki, Greece, Oct. 2001. IEEE. (Cited on page 106.)

K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. on Image Process.*, 2(2):160–175, 1993. (Cited on page 106.)

N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001. (Cited on page 175.)

G. Rose and P. Hawkes. *Turing: a fast stream cipher*, volume 2887 of *LNCS*, pages 290–306. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 2003. (Cited on page 188.)

Y. Sadourny and V. Conan. A proposal for supporting selective encryption in JPSEC. *IEEE Transactions on Consumer Electronics*, 49(4):846– 849, Nov. 2003. (Cited on page 10.)

A. Said. Measuring the strength of partial encryption schemes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'05)*, volume 2, Sept. 2005. (Cited on pages 7 and 52.)

A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–249, June 1996. (Cited on pages 21 and 22.)

T. Schell and A. Uhl. New models for generating optimal wavelet-packet-tree-structures. In *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium (SPS 2002)*, pages 225–228, Leuven, Belgium, Mar. 2002a. IEEE Benelux Signal Processing Chapter. (Cited on page 106.)

T. Schell and A. Uhl. Wavelet packet image coding revisited. In *CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002)*, Tromso-Trondheim, Norway, Oct. 2002b. IEEE Norway Section. file cr1090.pdf. (Cited on page 106.)

T. Schell and A. Uhl. Optimization and assessment of wavelet packet decompositions with evolutionary computation. *EURASIP Journal on Applied Signal Processing*, 2003(8):806–813, 2003. (Cited on page 106.)

J. Schneid and S. Pittner. On the parametrization of the coefficients of dilation equations for compactly supported wavelets. *Computing*, 51:165–173, May 1993. (Cited on page 65.)

B. Schneier. *Applied cryptography (2nd edition): protocols, algorithms and source code in C*. Wiley Publishers, 1996. (Cited on pages 180 and 186.)

H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable H.264/MPEG4-AVC extension. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '06*, pages 161–164, Atlanta, GA, USA, Oct. 2006. IEEE. (Cited on page 194.)

H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, Sept. 2007. ISSN 1051-8215. (Cited on page 194.)

Y.-S. Seo, M.-S. Kim, H.-J. Park, H.-Y. Jung, H.-Y. Chung, Y. Huh, and J.-D. Lee. A secure watermarking for JPEG-2000. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 2001. (Cited on page 66.)

J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Process.*, 41(12):3445–3462, Dec. 1993. (Cited on pages 21 and 22.)

C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. Kumar. Biometric encryption using image processing. In *Optical Security and Counterfeit Deterrence Techniques II*, volume 3314 of *Proceedings of SPIE*, pages 178–188, 1998. (Cited on page 175.)

T. Stütz and A. Uhl. Transparent image encryption using progressive JPEG. In S. Katsikas et al., editors, *Information Security. Proceedings of the 9th Infomation Security Conference (ISC'06)*, volume 4176 of *Lecture Notes on Computer Science*, pages 286–298. Springer Verlag, Sept. 2006a. (Cited on page 9.)

T. Stütz and A. Uhl. On format-compliant iterative encryption of JPEG2000. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM'06)*, pages 985–990, Los Alamitos, CA, USA, 2006b. IEEE Computer Society. ISBN 0-7695-2746-9. (Cited on page 10.)

T. Stütz and A. Uhl. On efficient transparent JPEG2000 encryption. In *Proceedings of ACM Multimedia and Security Workshop, MM-SEC '07*, pages 97–108, New York, NY, USA, Sept. 2007. ACM Press. ISBN 978-1-59593-857-2. (Cited on page 9.)

W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal*, 3(2):186–200, 1996. (Cited on page 65.)

A. Tabesh, A. Bilgin, K. Krishnan, and M. W. Marcellin. JPEG2000 and motion JPEG2000 content analysis using codestream length information. In *Proc. Data Compression Conference, DCC 2005*, pages 329–337. IEEE Computer Society Press, Mar. 2005. (Cited on pages 32, 46, and 56.)

D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158 – 1170, 2000. (Cited on pages 23 and 165.)

D. Taubman and M. Marcellin. *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002. (Cited on pages 19, 20, 22, 34, and 36.)

A. Uhl. Image compression using non-stationary and inhomogeneous multiresolution analyses. *Image and Vision Computing*, 14(5):365–371, 1996. (Cited on page 74.)

A. Uhl and C. Obermair. Transparent encryption of JPEG2000 bitstreams. In P. Podhradsky et al., editors, *Proceedings EC-SIP-M 2005 (5th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services)*, pages 322–327, Smolenice, Slovak Republic, 2005. (Cited on page 9.)

A. Uhl and A. Pommer. Are parameterised biorthogonal wavelet filters suited (better) for selective encryption? In J. Dittmann and J. Fridrich, editors, *Multimedia and Security Workshop 2004*, pages 100–106, Magdeburg, Germany, Sept. 2004. (Cited on pages 12 and 66.)

A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005. (Cited on pages 5, 6, and 13.)

U. Uludag, S. Pankanti, and S. Prabhakar. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004. (Cited on page 175.)

G. Unnikrishnan and K. Singh. Double random fractional fourier-domain encoding for optical security. *Optical Engineering*, 39(11):2853–2859, Nov. 2000. (Cited on page 12.)

L. Vorwerk, T. Engel, and C. Meinel. A proposal for a combination of compression and encryption. In *Visual Communications and Image Processing 2000*, volume 4067 of *Proceedings of SPIE*, pages 694–702, Perth, Australia, June 2000. (Cited on page 12.)

Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3), Mar. 2002. (Cited on page 15.)

S. Wee and J. Apostolopoulos. Secure scalable video streaming for wireless networks. In *Proceedings of the 2001 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, USA, Apr. 2001a. invited paper. (Cited on page 10.)

S. Wee and J. Apostolopoulos. Secure scalable streaming enabling transcoding without decryption. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 2001b. (Cited on page 10.)

M. Wickerhauser. *Adapted wavelet analysis from theory to software*. A.K. Peters, Wellesley, Mass., 1994. (Cited on pages 84, 85, 105, and 106.)

S. Winkler. Quality metric design: A closer look. In *Proceedings of SPIE Human Vision and Electronic Imaging*, volume 3959, pages 34–44, San Jose, CA, USA, Jan. 2000. (Cited on page 15.)

S. Winkler. A perceptual distortion metric for digital color images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'98)*, volume 3, pages 399–403, Chicago, IL, USA, Oct. 1998. (Cited on page 15.)

S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt. Vision and video: models and applications. In C. J. van den Branden Lambrecht, editor, *Vision Models and Applications to Image and Video Processing*, chapter 10. Springer-Verlag, 2001. (Cited on page 15.)

H. Wu and D. Ma. Efficient and secure encryption schemes for JPEG2000. In *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, pages 869–872, May 2004. (Cited on page 10.)

M. Wu and V. Mao. Communication-friendly encryption of multimedia. In *Proceedings of the IEEE Multimedia Signal Processing Workshop, MMSP '02*, St. Thomas, Virgin Islands, USA, Dec. 2002. (Cited on page 10.)

Y. Wu and R. H. Deng. Progressive protection of JPEG2000 codestreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapure, Oct. 2004. IEEE Signal Processing Society. (Cited on pages 7 and 10.)

Z. Xiong, K. Ramchandran, and M. T. Orchard. Wavelet packet image coding using space-frequency quantization. *IEEE Transactions on Image Processing*, 7(6):892–898, June 1998. (Cited on page 106.)

D. Xu and M. N. Do. Anisotropic 2-D wavelet packets and rectangular tiling: theory and algorithms. In M. A. Unser, A. Aldroubi, and A. F. Laine, editors, *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing X*, volume 5207 of *SPIE Proceedings*, pages 619–630, San Diego, CA, USA, Aug. 2003. SPIE. (Cited on pages 109, 113, and 115.)

Y. Yang, B. B. Zhu, Y. Yang, S. Li, and N. Yu. Efficient and syntax-compliant JPEG2000 encryption preserving original fine granularity of scalability. *EURASIP Journal on Information Security*, 2007. (Cited on page 10.)

C. Yuan, B. B. Zhu, M. Su, Y. Wang, S. Li, and Y. Zhong. Layered access control for MPEG-4 FGS. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, Barcelona, Spain, Sept. 2003. (Cited on page 9.)

G. Zhong, L. Cheng, and H. Chen. A simple 9/7-tap wavelet filter based on lifting scheme. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, pages 249–252, Oct. 2001. (Cited on page 91.)

H. Zhong and L. Jiao. Novel secret key watermarking system. In *Proceedings of SPIE, International Symposium on Multispectral Image Processing and Pattern Recognition, Image Compression and Encryption Technologies*, volume 4551, Wuhan, China, Oct. 2001. (Cited on pages 66 and 67.)

COLOPHON

This thesis was typeset with LaTeX 2ε. The *TEX Live* system was used with the collection of *KOMA-Script* packages and numerous other valuable packages from various authors, acquired through the CTAN network. The layout is inspired by and based on the layout of the *ClassicThesis* package by André Miede (available through CTAN).

The main typeface is Adobe *Minion Pro*, designed by Robert Slimbach. The mathematical symbols are set with *MnSymbol* by Achim Blumensath. The chapter numbers use the *Euler* typeface, designed by Hermann Zapf (assisted by Donald Knuth).