UNIVERSITÄT SALZBURG

Analysis of JPEG 2000 Encryption with Key-dependent Wavelet Packet Subband Structures

Dominik Engel Thomas Stütz Andreas Uhl

Technical Report 2010-09 December 2010

Department of Computer Sciences

Jakob-Haringer-Straße 2
5020 Salzburg
Austria
www.cosy.sbg.ac.at

Technical Report Series

Analysis of JPEG 2000 Encryption with Key-dependent Wavelet Packet Subband Structures

Dominik Engel, Thomas Stütz, and Andreas Uhl

Abstract— **We analyze and discuss encryption schemes for JPEG 2000 based on the wavelet packet transform with a key-dependent subband structure. These schemes have been assumed to reduce the runtime complexity of encryption and compression. In addition to this "lightweight" nature, other advantages like encrypted domain signal processing have been reported.**

We systematically analyse encryption approaches based on key-dependent subband structures in terms of their impact on compression performance and the level of security they provide as compared to more classical techniques based on JPSEC. Furthermore, we analyse the prerequisites and settings in which the previously reported advantages actually hold and in which settings little to no advantages can be observed.

As a final outcome it has to be stated that this compression integrated encryption approach based on the idea of secret transform domains can be recommended for highly specialised application scenarios only. Previously reported advantages have turned out to apply only partially or only under specific circumstances, and do usually not justify to accept the obvious disadvantages associated with these approaches.

I. Introduction

For securing multimedia data – like any other type of data – full encryption with a traditional cipher, such as AES, is the most secure option. However, in the area of multimedia many applications do not require the level of security this option provides, and seek a trade-off in security to enable other requirements, including low processing demands, retaining bitstream compliance and scalability, and the support for increased functionality, such as transparent encryption [1]. Lightweight encryption aims at striking a balance between security and these other requirements.

JPEG 2000 is the most recent and comprehensive suite of standards for scalable coding of visual data [2], [3]. Although JPEG 2000 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 JPEG 2000 to really gain momentum, but recently JPEG 2000 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 JPEG 2000 as the compression standard in their specification for a unified Digital Cinema System [4]. As a second example, in 2002, the DICOM (Digital Imaging and Communications in Medicine) committee approved the final text of DICOM Supplement 61, marking the inclusion of Part 1 of JPEG 2000 in DICOM (ISO 12052). Further supplements (105 and 106) include the Part 2 multi-component transform syntax and JPIP, respectively. Furthermore, in the ISO/IEC 19794 standard on Biometric Data Interchange Formats JPEG 2000 is included for lossy compression, in the most recently published version (ISO/IEC FDIS 19794-6 as of August 2010) as the only format for iris image data, for example. Security techniques specifically tailored to the needs of scalable representation in general and JPEG 2000 in particular have been proposed recently, e.g., [5], [6], [7], [8], [9], [10]. An overview and discussion of the proposed approaches in the context of JPEG2000 can be found in [11].

JPEG 2000 security is discussed in JPEG 2000 Part 8 ([12], [13]). This part has the title "Secure JPEG 2000" and is referred to as JPSEC. It "intends to provide tools and solutions in 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 are discussed in this paper. JPSEC extends the codestream syntax to allow parts which are created by security tools, e.g., cipher or authentication tools. Furthermore, complementing the normative part, informative application examples are given.

The approaches discussed in this paper fall into the category of encryption by constructing a secret transform domain. 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 transparent encryption is the goal). Other than with bitstream-oriented methods, which operate on a finished media bitstream, these methods apply encryption integrated with compression. In terms of applicability they are therefore restricted to scenarios where the final media bitstream is not yet available (video

Manuscript received April 2009; revised October 2009. This work has been partially supported by the the European Commission through the IST Programme under contract IST-2002-507932 ECRYPT and the Austrian Science Fund (FWF) under project no. 15170 and project no. 19159.

The authors are with the Computer Sciences Department, Salzburg University, Jakob-Haringer-Str. 2, A–5020 Salzburg, Austria (e-mail: dengel@cosy.sbg.ac.at; tstuetz@cosy.sbg.ac.at; uhl@cosy.sbg.ac.at).

conferencing, live streaming, photo storage, transmission and storage of surveillance data etc.).

A focus is set on supporting transparent encryption by the discussed approaches. The term was introduced in the context of TV broadcasting ([14], [1]) 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 other media encryption schemes that guarantee a certain degree of distortion 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. [10], [15]). 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. The reason for considering transparent encryption as target application scenario is that it has been shown [16], that the lowest resolution contained in a JPEG 2000 file encrypted using the techniques discussed in this paper can always be decoded in JPEG 2000. This enables transparent encryption in an elegant manner, but is contra-productive for applications requiring a higher degree of confidentiality or different notion of security for the data.

The concept of key-dependent basis functions in multimedia security is introduced by [17] to protect a watermark from hostile attacks (at the cost of a significant increase in computational complexity). The method is developed further by [18] with the proposal of a faster method for the generation of key-dependent orthogonal patterns. A technique for data hiding using a key-dependent basis function in the tree structured Haar transform domain is proposed by [19]. A key-based choice of parameterized wavelet filters is suggested to establish watermark security in [20], [21], [22]. There are also some propositions that use secret Fourier transforms: the embedding of watermarks in an secret domain is discussed by [23], and [24] suggest to use this technique for encryption of visual data. Other proposals in the area of lightweight encryption [25] propose the encryption of the filter choice used for a wavelet decomposition. However, this suggestion remains vague and is not supported by any experiments, while [26], [27] propose encrypting the orthogonal filterbanks used for an non-stationary multi-resolution analysis (NSMRA) decomposition. The use of concealed biorthogonal parameterized wavelet filters for lightweight encryption is proposed by [28]. The use of key-dependent wavelet packet decompositions is proposed first by [29], [30]. The latter work [30] evaluates encryption based on key-dependent subband structures in a zerotree-based wavelet codec.

Other areas where key-dependent wavelet packet decompositions have been used are watermarking and image hashing. The main motivation for introducing key-dependent wavelet packets is increasing the security of existing schemes. In [31], [32] and [33], key-dependent wavelet packet decompositions are proposed for increasing the security of watermarking schemes. Successive watermarking employing different wavelet packet decompositions for each embedded mark is investigated in [34], where watermark inference in clearly reduced.

In the context of JPEG 2000, the degrees of freedom in the wavelet transform are a prime candidate for constructing a secret transform domain. JPEG 2000, Part 2, allows the definition of custom wavelet filters and user-defined isotropic and anisotropic wavelet packet subband structures [35]. Parameterized wavelet filters have been employed for JPEG 2000 lightweight encryption by [36], [37], however, this approach was shown to be insecure in later work [38]. Key-dependent parameterized wavelet filters as well as key-dependent isotropic wavelet packets have been used to propose a JPEG 2000-based secure image authentication scheme [39], [40]. The key-dependent wavelet transform schemes are used to transform the source data, subsequently a JPEG 2000-packet body data based hash is created in the secret transform domain.

Key-dependent wavelet packet structures in JPEG 2000 have been proposed for a lightweight encryption scheme in earlier work [16], [41], [42]. This approach is in the focus of interest in this work. The suggested scheme can be seen as a form of header encryption, as only the information pertaining to the transform 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 need to be kept secret, so the demands for the encryption stage are minimal as compared to a more traditional, bitstream-oriented encryption approach [16]. Due to the shift in complexity from actual encryption to the compression pipeline, the scheme has been termed “lightweight”. Another possible advantage is that these approaches are said to be suited for signal processing in the encrypted domain to a certain degree, a research area that has gained a lot of attention recently, simply because the encrypted domain *is* a transform domain in some sense.

In this paper, we evaluate, analyse, and discuss earlier proposed JPEG 2000 encryption techniques that use key-dependent wavelet packet subband structures (KDWPSSs) to establish a secret transform domain. In our analysis we assess the significance of disadvantages of this approach, and we evaluate potential advantages as follows:

- **Compression impact**: Using random KDWPSSs for compression instead of a pyramidal decomposition scheme obviously impacts the compression performance. In previous work on this topic the image test sets were restricted to very few images and only a proof of concept was conducted. In this paper we complement previous work by an evaluation with a large set of images. The evaluation is performed for both isotropic and anisotropic wavelet packets using two different methods for subband structure selection: a compression-oriented method and a method that selects uniformly from the set of all subband structures.
- **Security**: Since the techniques investigated leave the entire JPEG 2000 packet body data as well as packet header data in plaintext, a detailed security analysis is required in order to find out whether it might be possible to infer information from these plaintext data to reconstruct the actually employed KDWPSS (an investigation of attacks

that are specific to the used codec – JPEG 2000– is an important issue that has not been discussed before). A further question is which kind of data an attacker is able to reconstruct based on which computational effort involved. Recall, the lowest resolution can always be decoded from the bitstream generated by KDWPSS encryption [16], which is why KDWPSS-based approaches have been found to be unsuited for providing full confidentiality [42].

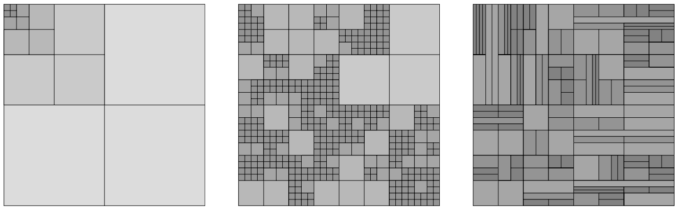
- **Computational demand**: The main argument for introducing the general concept of secret transform domains in encryption has always been the obvious reduction in *conventional encryption* effort, e.g., AES encryption. When replacing the pyramidal wavelet transform by KDWPSSs as being proposed by the analysed schemes, however, additional computations are introduced in the joint *compression and encryption* algorithm. Therefore, a careful analysis is required how these two contradictory properties affect the overall computational demand.
- **Encrypted domain processing**: Another potential advantage of the analysed scheme is the capability of performing several operations on the protected data (“encrypted domain processing”). We discuss which operations can actually be conducted and which cannot.

The organization of this paper is as follows. In Section II, after briefly discussing wavelet packets and their incorporation into the JPEG 2000 standard, we summarize work on the randomized generation of isotropic and anisotropic decomposition structures [30], [16], [41]. For each case we discuss two compression-oriented selection methods and a selection method that uniformly selects from the set of all possible subband structures (up to a specified decomposition level).

In Section III we validate the previously discussed work with a larger evaluation of the compression performance of randomized wavelet packets for natural images. The security of the proposed approaches is discussed in depth in Section IV. In particular, we recapitulate the size of the generated keyspaces (upper bound on the entropy) and give more accurate estimations on the entropy of the employed distributions of KDWPSSs allowing a more detailed security estimation compared to only considering the size of the keyspace. Further, we estimate an attackers’ effort to generate a certain quality based on a given resolution used for transparent encryption. Section V compares the computational complexity of the KDWPSSs approach to JPSEC based approaches analytically (i.e., in terms of actual operation counts) and experimentally. Section VI discusses the applicability of the analyzed KDWPSSs approach also in comparison to JPSEC based approaches. Section VII summarizes the results and concludes.

II. Wavelet Packets

The wavelet packet transform (WPT) [43] generalizes the pyramidal 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 1. This results in a large space of possible decomposition structures (of which the pyramidal decomposition structure is a single element). For a specific maximum decomposition depth, there are many possible WP-structures – the WPT is an overcomplete library of bases. Each decomposition structure represents represents a unique wavelet packet basis. These decompositions can be adapted to take the properties of the image to be transformed into account, for example by using the best basis algorithm [43], [44]. In this paper we refer to each such a wavelet packet basis by the terms “wavelet packet subband structure” (WPSS) or “decompostion structure”.



(a) Pyramidal wavelet decomposition

(b) Isotropic Wavelet packet decomposition

(c) Anisotropic wavelet packet decomposition

Fig. 1. Example decomposition structures

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 1(c).

Note that for the isotropic wavelet packet transform a single decomposition refers to both horizontal and vertical filtering and downsampling. For the anisotropic wavelet packet transform, a decompositions refers to filtering and downsampling in one direction (horizontal or vertical). Therefore, a decomposition depth of $2k$ in the anisotropic case is comparable to a decomposition depth of k in the isotropic case. This paper deals with both isotropic and anisotropic WPSS.

A. Wavelet Packets in JPEG 2000

Part 2 of the JPEG 2000 standard [35] allows arbitrary decomposition structures. New possible orientations are defined 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. The notion of resolutions in JPEG 2000 remains unchanged in Part 2, and is only determined by the all low-pass decomposition branch. However, whereas in part 1 of the JPEG 2000 standard, this branch can only contain the LL-subband, in Part 2, LX and XL are possible: “[s]ince 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” [35, p. 92]. Every subband resulting from a highpass filtering can be decomposed at most 2 more times (either horizontally, vertically or in both directions).

In order to maximize keyspace size for the proposed encryption scheme, we have implemented full support for arbitrary isotropic and anisotropic wavelet decomposition structures in JPEG 2000, based on the JJ2000 reference implementation.[1] In the more recent versions, Kakadu[2] is able to employ custom decompostion structures. The source code for the implementation underlying all results in this paper can be downloaded from `http://www.wavelab.at/sources`.

B. Randomized Generation of Isotropic Wavelet Packet Structures

Wavelet packet decomposition structures can be derived by a sequence of binary decomposition decisions. A naive approach for generating randomized wavelet packet structures is to use a fixed decomposition probability for each subband, e.g., 0.5. 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 far more probable than deep ones. Such a bias severely undermines the security of the proposed encryption scheme.

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

A problem with the method of uniformly selecting from the set of all structures will be that not every subband structure is well suited to be used for compression. Some of the subband structures lead to inferior compression performance. However, the method of selecting from all subband structures by uniform distribution is good in terms of comparison, as for security the uniform distribution forms an upper bound.

To overcome deficiencies in compression performance, a compression-oriented selection method has been introduced [16], [41], [42], which aims at discarding the subband structures that are not suitable for compression. Of course the remaining subband structures should still form a keyspace that is sufficiently large to provide lightweight security.

1) Uniform Distribution: In the isotropic case a uniform distribution for selecting a subband structure (of maximum decomposition depth g) 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 decomposition level for the node. Then the probability $p(l)$ for decomposition for this node is given by [45] as

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

where g is the maximum overall decomposition depth and Q_j is the number of possible subband structures with a decomposition depth up to j. Q_j can easily be determined, e.g., using the recursive formula proposed by [46]:

$$Q_j = Q_{j-1}^4 + 1 \tag{2}$$

where $Q_0 = 1$. One possible subband structure comes from the case where the node is not further decomposed. If the node is further decomposed then the number of possible subband structures is given by the combination of possible decompositions in the subtree for each subband. Subsequently, this distribution is denoted / abbreviated by *isouni*.

2) Compression-oriented Distribution: In order to limit the selection process to subband structures 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.

[30] introduces the following decision process for decomposition: a random number between 0,2 is divided by a weight, which is computed at every decomposition level. If the result is smaller than 1, no further decomposition is computed. In order to make it possible to influence the selection process, two parameters are introduced to compute the weight, which allows to favor the selection of deeper or more shallow decompositions: the *base value (b)* and the *change factor (c)*. 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 b determines the basic probability with which a subband is decomposed. The change factor c alters this probability based on the decomposition depth of the subband. The algorithm by [30] for deciding whether to decompose or not is given in Listing 1. Note that by the decomposition level we refer to the number of decompositions that have been conducted to obtain the subband. For the approximation subband, the notion of resolution is reciprocal to decomposition level. The higher the decomposition level, the lower the resolution. If c is negative, then the higher the level of the subband, the higher is the chance for it to be decomposed. If c is positive, 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. The “base value” gives the initial probability determining the decomposition decision. At a value of 1.0 the two possibilities are equal, when the number is lower than this the decision favours further decompositions (see [30]). The “change factor” alters the weight in a decomposition depth dependent way. If the change factor is 0, the “base value” stays the same on all decomposition levels. Otherwise it is added to the “base value” at every level of decomposition thereby increasing (or decreasing) the weight at each decomposition level. Generally, it is advisable to tune these parameters to produce a balanced distribution. [30] suggests to set the base value to 1 and the change factor to 0.

Another parameter is the seed s for the pseudo-random number generator (PRNG). The seed is used to initialize the PRNG.

To achieve transparent encryption, an additional parameter p is introduced by [16] 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

[1] `http://jj2000.epfl.ch/`

[2] `http://www.kakadusoftware.com`

Listing 1. Random Generation of Isotropic Wavelet Packets

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

R_{p+1}. With resolution-layer progressions in the final bitstream, standard JPEG 2000 codecs can be used to obtain resolutions R_0 to R_p. Note that a decoder compliant to JPEG 2000, part 1 is sufficient to decode the preview image.

Note that the definition of a subband structure only depends on the aforementioned parameters. Only these parameters need to be encrypted. Furthermore, as a random number generator is used to control the creation of the randomized subband structures, a minimal change in the seed (e.g. the change of a single bit) will result in a completely different subband structure. This results in an excellent diffusion property. Subsequently, these distributions are denoted / abbreviated by *iso* and further classified into a constrained (the LL is always further decomposed) and an unconstrained case.

C. Randomized Generation of Anisotropic Wavelet Packet Structures

The main motivation to introduce anisotropic wavelet packets in the context of lightweight encryption is a significant increase in keyspace size [41], [42]. This increase is due to the fact that the anisotropic transform has more degrees of freedom.

Even more than in the case of isotropic wavelet packets, there are anisotropic wavelet packet decompositions that are ill-suited for energy compaction. The compression-oriented selection method tries to eliminate these subband structures.

1) Uniform Distribution: We use the case distinction introduced by [45] to construct a uniform distribution for the selection of a random subband structure: the probability for any case to be chosen is the ratio of the number of subband structures contained in the case to the total number of subband structures. Subsequently, this distribution is often denoted / abbreviated by *anisouni*.

2) Compression-oriented Distribution: The basic algorithm for the compression-oriented generation of randomized anisotropic wavelet packet subband structures stays the same. However, the parameters for compression-oriented selection of anisotropic wavelet packet differ from the isotropic case in order to reflect the properties of the anisotropic wavelet packet transform.

Four parameters, n, m, e, d, determine the maximum and minimum decomposition depths for the approximation subband and the detail subbands, respectively. They influence both, compression performance and keyspace size.

Constraining the degree of anisotropy may be necessary for some subbands in order to prevent them from 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 transform domain for the other direction. Two parameters, q and r, are used to restrict the maximum degree of anisotropy for the approximation and detail subbands, respectively. For the degree of anisotropy Υ of a subband we use the following definition:

$$\Upsilon(h, v) = v - h \tag{3}$$

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., $|\Upsilon(h, v)|$. The definition Υ of the degree of anisotropy that we use here is also used in [45], 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 an anisotropic wavelet packet subband structure, 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 approximation and detail subbands influence both, compression performance and keyspace size.

Some parameters are used in the same way as in the isotropic case: the seed s initializes the PRNG. The base value b set the basic probability of decomposition and the change factor c alters this base probability depending on the current decomposition level.

The algorithm for the randomized generation of anisotropic wavelet packet structures is shown in Listing 2.

The accommodation of transparent encryption into this scheme is proposed by [41] by adding 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 JPEG 2000, part 1 can be used to decode the first p resolutions. Subsequently, this distribution is often denoted / abbreviated by *aniso* and

Listing 2. Random Generation of Anisotropic Wavelet Packets

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

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

 function direction_decision:
      x = random([0,1[)
      if (x < 0.5) then
              direction = vertical
      else
              direction = horizontal
   if (degree_of_anisotropy(direction) >=
    max_approx_degree_of_anisotropy) then
              invert direction
      decompose (direction)
```

are further classified in a constrained case (the degree of anisotropy is restricted) and an unconstrained case.

III. Compression Performance

In the comparison of compression quality of randomized wavelet packet decompositions and the pyramidal wavelet decomposition it is noteworthy that there are wavelet packet decomposition structures that produce better results than the pyramidal structure (cf. [16], [47]). This effect is stronger for images with oscillatory patterns that facilitate energy compaction with wavelet packets. However, without restrictions, the wavelet packet transformation is outperformed by the pyramidal wavelet transformation. Therefore, settings for the aforementioned parameters have to be determined that discard the subband structures that do not produce good compression results.

In previous work ([16], [41]) parameter settings for the compression-oriented distribution have been determined for a small number of test images. For the isotropic wavelet packet transform it is proposed to force maximum decomposition depth for the approximation subband. For the detail subbands it is proposed to leave as much degrees of freedom as possible. For the anisotropic wavelet packet transform, in addition to forcing on maximum decomposition depth, the maximum degree of anisotropy for the approximation subband needs to be restricted: if the approximation subband is kept as close to an isotropic decomposition as possible, compression results can be produced that have the same quality as the compression results of the isotropic wavelet packet transform.

A. Empirical Evaluation of Compression Performance

The parameters proposed by [16], [41] were obtained empirically by a number of experiments. A large number of different parameter settings were used, but only on three test images. The parameters that were obtained for these three test images are given in Table I. We use these parameters in the empirical setup discussed below, and evaluate their performance for a larger set of test images. Recall 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.

We verify the compression performance of the compression-oriented selection method by an empirical study based on an extended set of images. For this purpose we use a set of 100 grayscale images of 512×512 pixels (taken with four different camera models).

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 JPEG 2000 compression tests at each of these bitrates:

- Pyramidal (1 subband structure, level 5),
- Isouni: randomized isotropic wavelet packets with uniform distribution (100 randomly selected subband structures),
- Iso (constrained): randomized isotropic wavelet packets with compression-oriented distribution (100 randomly selected subband structures),
- Anisouni: randomized anisotropic wavelet packets with uniform distribution (100 randomly selected subband structures), and
- Aniso (constrained): randomized isotropic wavelet packets with compression-oriented distribution (100 randomly selected subband structures).

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 standard CDF 9/7 biorthogonal wavelet was used for transformation in all experiments. The results of our empirical study are summarized for the 5 categories and all bitrates in Figure 2. The exact results are given in tabular form in Table II, which lists the average PSNR and the number of samples for each category and bitrate. Note that the wavelet packet decomposition structure was not counted towards the overall bitrate. A detailed analysis of the coding demand for decomposition structures as well as efficient representations can be found in [45].

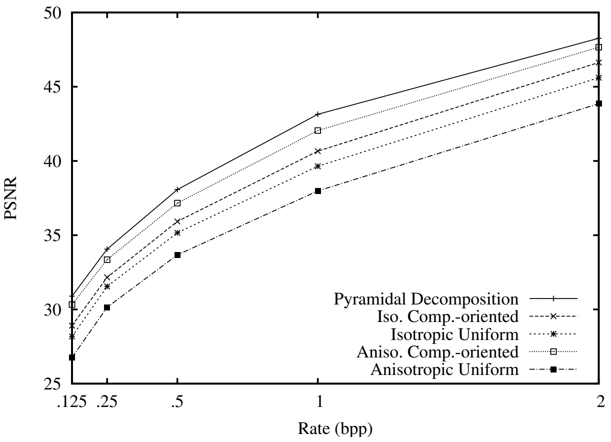


Fig. 2. Empirical results: average compression performance (100 images)

For the compression-oriented setup, the loss in compression performance is smaller for the anisotropic randomized decomposition method (below 1 dB). Due to the fact that randomized 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. In part this is due to the overhead in header data that is introduced in the JPEG 2000 bitstream by increasing the number of subbands (as is usually the case in KDWPSSs). Table III shows the average ratio of header data to packet data for all test images at different bitrates. As the header size is less affected by bitrate it can be seen that the ratio increases when the bitrate decreases and can make up a substantial part of the bitstream.

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, 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 subband structures 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 subband structures there are many that yield inferior compression results. In terms of compression performance it is therefore important to limit the number of admissible subband structures. Only the compression-oriented approach ensures good compression results. In the following we will evaluate if it also yields acceptable security.

Parameter Name	Isotropic		Anisotropic	
	isouni	iso	anisouni	aniso
Max. global decomposition depth (g)	5	5	10	10
Max. approx. decomposition depth (m)	5	5	10	10
Min. approx. decomposition depth (n)	5	5	10	10
Max. detail. decomposition depth (d)	5	5	10	10
Min. detail. decomposition depth (e)	0	0	0	0
Max. degree of anisotropy approx. subband (q)	n/a	n/a	n/a	1
Max. degree of anisotropy detail subbands (r)	n/a	n/a	n/a	∞
Base value (b)	n/a	0.25	n/a	0.25
Change factor (c)	n/a	0.1	n/a	0.1

TABLE I
PARAMETERS USED FOR THE EMPIRICAL STUDY

IV. Security Evaluation

In order to assess the security of a multimedia encryption approach, we first need to define "security" of a multimedia cryptosystem more precisely. Conventional notions of security for cryptosystems require that the ciphertext does not leak any information (information-theoretic approach [48]) or any efficiently computable information (the approach of modern cryptography [49]) of the plaintext. This kind of security notions are also referred to as MP-security (message privacy) [50]. As most of the plaintext data is preserved by only keeping the WPSS secret, the conventional cryptographic security notions are obviously not met.

In lack of applicability of the conventional MP-security notion, multimedia encryption is often analyzed with respect to a full message (in our case image) recovery. This type of security notion is referred to as MR-security (message recovery) [50]. However, a reconstruction of a multimedia datum (on the basis of the ciphertext) may have excellent quality and even be perceived as identical by a human observer, while the perfect recovery of the entire message remains impossible.

Thus in the context of multimedia encryption it is required to take the quality of a reconstruction (by an adversary on the basis of the ciphertext) into account. An adversary, who tries to break a multimedia encryption system, is successful if she can efficiently compute a "high quality" reconstruction of the original multimedia datum. Which quality constitutes a security threat highly depends on the targeted application scenario [11]. In [51] this multimedia-specific security notion is termed MQ-security (message quality), similar concepts can be found in the multimedia encryption literature [52], [53]. A sensible definition of MQ-security for KDWPSSs is proposed in section IV-A. This security notion gives rise to a specific class of attacks in which an adversary tries to improve the quality of her reconstruction.

Basically, KDWPSSs could be considered a symmetric

Rate	Sel. Method	WP Type	Avg. PSNR	Samples
0.125	pyr.	pyr.	30.92	100
0.125	compr.	aniso	30.33	10000
0.125	compr.	iso	28.91	10000
0.125	compr.	$\sum$	29.62	20000
0.125	uniform	aniso	26.76	10000
0.125	uniform	iso	28.18	10000
0.125	uniform	$\sum$	27.47	20000
0.25	pyr.	pyr.	34.08	100
0.25	compr.	aniso	33.35	10000
0.25	compr.	iso	32.16	10000
0.25	compr.	$\sum$	32.76	20000
0.25	uniform	aniso	30.14	10000
0.25	uniform	iso	31.55	10000
0.25	uniform	$\sum$	30.85	20000
0.5	pyr.	pyr.	38.08	100
0.5	compr.	aniso	37.16	10000
0.5	compr.	iso	35.92	10000
0.5	compr.	$\sum$	36.54	20000
0.5	uniform	aniso	33.67	10000
0.5	uniform	iso	35.16	10000
0.5	uniform	$\sum$	34.41	20000
1	pyr.	pyr.	43.15	100
1	compr.	aniso	42.05	10000
1	compr.	iso	40.66	10000
1	compr.	$\sum$	41.36	20000
1	uniform	aniso	37.98	10000
1	uniform	iso	39.64	10000
1	uniform	$\sum$	38.81	20000
2	pyr.	pyr.	48.27	100
2	compr.	aniso	47.66	10000
2	compr.	iso	46.63	10000
2	compr.	$\sum$	47.15	20000
2	uniform	aniso	43.86	10000
2	uniform	iso	45.61	10000
2	uniform	$\sum$	44.73	20000
$\sum$	$\sum$	$\sum$	36.38	200500

TABLE II

EMPIRICAL RESULTS: COMPRESSION PERFORMANCE (100 IMAGES)

Rate	pyramidal	iso	aniso
0.125	15.9%	32.8%	20.8%
0.25	10.0%	20.8%	15.1%
0.5	6.1%	13.6%	9.6%
1	3.7%	9.2%	5.9%
2	2.5%	6.4%	3.9%

TABLE III

RATIO OF HEADER DATA TO PACKET DATA FOR DIFFERENT COMPRESSION RATES (16 QUALITY LAYERS)

cryptosystem, the secret key is the employed WPSS. An important aspect for the security analysis is the fact that the WPSS selection schemes introduce a distribution on the set of all decompostion structures with a specified maximum decompostion depth. If the ciphertext does not leak information on the WPSS, an attacker that tries to break the scheme in the sense of MR-security, i.e., a full recovery, has to guess the key-dependent WPSS. The complexity she has to face can be derived from the entropy of the WPSS distribution (resulting from the WPSS selection method). The entropy of the distributions has not been considered so far in the security analysis of KDWPSSs, but merely the size of the key space, i.e., the number of possible WPSSs, has been taken into account [16], [41], [42]. The logarithm dualis of the key space cardinality constitutes an upper bound for the entropy of a distribution on the key space, which is reached in the case of a uniform distribution on the key space. The entropy of the distributions for the proposed WPSS selection schemes is in-depth analyzed in section IV-B.

However, the actual threat for multimedia encryption is more appropriately captured by the MQ-security notion, i.e., an adversary tries to compute a high quality reconstruction. The complexity she has to face is discussed in section IV-C.

It is highlighted [16], that the lowest resolution of a WPSS can always be decoded in JPEG 2000, which enables transparent encryption. The application of the coding framework of JPEG 2000 puts constraints on the lowest quality achievable by KDWPSS, which are in detail discussed in section IV-D. We have to point out that JPEG 2000 Part 2 only allows a subset of WPSSs. In section IV-E we investigate whether the subset of admissible WPSSs in JPEG 2000 Part 2 is still large enough to be employed for encryption purposes. If this would be the case an arbitrary JPEG 2000 Part 2 encoder could be employed for encryption.

A. MQ-Security for KDWPSSs and a Possible Attack

It is important to highlight that the goal of an adversary is not the full recovery of the image, even a “high quality” reconstruction of the image is a security threat. Therefore we need to define “quality” in the context of KDWPSSs; a natural quality indicator is the resolution on which the reconstruction is based, i.e., we assign an image I the quality $\frac{1}{1+l}$ if it is equal to the reconstruction on the basis of the resolution at decomposition depth l (and 0 otherwise). E.g., the original image is assigned a quality of 1, a reconstruction on the basis of the “first” resolution, i.e., at decomposition depth 1, is assigned a quality of 0.5, and reconstruction that are not based on any resolution of the original image are assigned a quality of 0. The definition applies to both the isotropic and the anisotropic case, the difference is in the definition of decomposition depth and resolution. For the isotropic case one decomposition depth consists of horizontal and vertical filtering, while for the anisotropic case a decomposition depth is either horizontal or vertical filtering and a resolution is always the resulting low frequency band.

This definition of MQ-security, i.e., quality, is further justified in the transparent encryption application scenario, where a low quality version (based on a low resolution in the case of

KDWPSSs) has to be available. The goal of an adversary is to subsequently decode the higher resolutions. Thus the security of KDWPSSs is measured in the complexity an adversary has to face in order to compute an reconstruction based on a certain resolution.

In the following we investigate whether the JPEG 2000 codestream leaks information on the employed KDWPSSs, that can be employed for attacks in the MQ-security sense. As already mentioned, the coding framework of JPEG 2000 ensures that the lowest resolution is accessible, as it is contained in the first contributions of the tile stream, i.e., right after the main header. Therefore, the central questions for assessing the security of KDWPSSs is whether the next resolution can be decoded from the codestream (independently of the higher resolutions) and whether it is decidable that the employed subband decomposition structure is the correct one, i.e., the one used for encoding. The answer to the first question is yes: in the coding framework of JPEG 2000, a resolution can be decoded independently from the remaining higher resolutions (definitely for resolution progression and at least at the lowest quality for layer progression and always if SOP and EPH markers are employed which signal packet borders). It is also highly likely that it can be decided whether the correct decomposition structure has been employed in the decoding of a resolution: Firstly, the wavelet resolutions are not independent, i.e., statistical cross-resolution dependencies are highly likely to identify the correct decomposition. Secondly, the codestream syntax and semantics must also be met while decoding with a subband decomposition structure, i.e., decoding errors clearly indicate an incorrect decomposition structure. Thus the decomposition structure of a resolution can be determined independently of the higher resolutions in JPEG 2000. As a result the security analysis has to take this into account and ask the question how hard is it for an adversary to decode a certain resolution. The complexity an adversary has to face is given by the entropy of the distribution on the WPSS on a certain resolution, i.e., given a distribution on KDWPSSs, what is resulting distribution of subband decomposition structures of a certain resolution (see section IV-C for in-depth analysis).

B. Entropy of Distributions on WPSSs

The number of possible WPSSs, both isotropic and anisotropic, grows tremendously with the maximum decomposition depth [42]. Even if the set of possible WPSSs is constrained as in the case of the compression oriented distributions (iso and aniso), the number of possible KDWPSSs remains sufficiently large to render brute-force attacks infeasible. Table IV gives the number of all possible WPSSs (isouni) and the number of possible KDWPSSs if the LL subband is always further decomposed (iso) in dependency of the maximum decomposition depth, as the latter approach enables better compression for the isotropic decomposition case. For a maximum decomposition depth in excess of 4 the number of possible WPSSs is more than sufficient to generate large enough key sets (a prerequisite for security) for both cases. The resulting key sets are larger than those of state-of-the-art symmetric ciphers, e.g., AES with a key set size of 2^{128}. Table V gives an overview of the number of possible anisotropic WPSSs with uniform and with compression oriented distribution; again the number of possible KDWPSSs is more than sufficiently large for both cases. When the number of anisotropic WPSSs is compared to the number of isotropic WPSS, it can be seen that the increase in keyspace size introduced by the use of the anisotropic wavelet packet transform is substantial.

g	iso (constrained)	iso (unconstrained)
3	$\approx 2^{15}$	$\approx 2^{16}$
4	$\approx 2^{64}$	$\approx 2^{65}$
5	$\approx 2^{260}$	$\approx 2^{261}$
6	$\approx 2^{1045}$	$\approx 2^{1046}$

TABLE IV
NUMBER OF ISOTROPIC WPSSS FOR THE COMPRESSION-ORIENTED CASE (ISO, LL IS ALWAYS DECOMPOSED) VS. THE UNCONSTRAINED CASE OF THE COMPRESSION ORIENTED-DISTRIBUTION OF WPSSS

	aniso (constrained)				aniso (unconstrained)
m	n	d	q	#WPSSs.	#WPSSs.
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 V
NUMBER OF ANISOTROPIC WPSSS WITH CONSTRAINTS AND WITHOUT CONSTRAINTS

The number of WPSSs only gives an upper bound for the complexity an adversary has to face in a brute-force attack. If the distribution on WPSSs is uniform, this upper-bound is reached, i.e., the expected number of trials in a brute-force attack is actually half of the number of KDWPSSs, which corresponds to $2^{H(X)-1}$, where H is the entropy of the discrete random variable X which is distributed uniformly on the set of all possible KDWPSSs. Thus the uniform distributions are secure (assuming that no information on the WPSS is leaked from the plaintext data).

In case of the compression-oriented distribution, not all WPSSs are equally probable and thus an adversary can take advantage of the non-uniformity of the distribution (i.e., by testing KDWPSSs with higher probability first), a standard measure for that end is the entropy of the distribution. As the number of possible WPSSs is large even for moderate maximum decomposition depths the computation of the entropy for a distribution on the set of possible KDWPSSs is complex. Basically for each possible WPSS ψ, its probability p has to be determined and the value of $-p_\psi \, \mathtt{ld} \, p_\psi$ computed (the entropy is the sum of this value of all ψ). A futile approach given the number of possible WPSSs. However, if we consider the algorithm for randomized generation of isotropic

g	Entropy (iso)	Entropy (isouni)
2	2.4	4.1
3	9.1	16.3
4	32.4	65.4
5	114.0	261.6
6	399.5	1046.4

TABLE VI
ENTROPY OF THE ISOTROPIC COMPRESSION-ORIENTED DISTRIBUTION (b=1/4, c=0) AND THE UNIFORM DISTRIBUTION

g	Entropy (anisouni)
4	19.7
5	40.3
6	81.7
7	164.4
8	329.7
9	660.5
10	1321.9
11	2644.9
12	5290.7

TABLE VII
ENTROPY OF THE ANISOTROPIC UNIFORM DISTRIBUTION

WPSSs without constraints (no mandatory LL decomposition), its simple structure can be exploited to reduce the complexity of the computation of the entropy (details are explained in appendix I). This allows us to compute the entropy for decomposition depths, which have high entropy values (see table VI). Since the entropy values in excess of state-of-the-art ciphers key set sizes (128 bit) can be considered secure, security if sufficient for $g > 5$ in the compression oriented case and for $g > 4$ in the case of uniform WPSSs distribution.

The computation of the entropy for the anisotropic compression-oriented distribution is even more complex, as there are substantially more WPSSs to consider and also because the anisotropic wavelet packet transform does not result in decomposition trees, but rather in less structured graphs, called "bushes" in [45]. The simplification of the computation of the entropy relies on tree structures and thus we can only give the upper bound of the entropy, which is the entropy of the uniform distribution (given in table VII). The high entropy values indicate that even a severe reduction of the entropy for the compression oriented distribution does not harm the MR-security (full message recovery is threat) of the scheme.

C. The Entropy of Distributions on Lower Resolution Decomposition Structures

As outlined, the goal of an adversary is to reconstruct the image with a "higher quality", where in the case of KDWPSSs it is sensible to define quality in terms of the resolution on which the reconstruction is based. The complexity she has to face is given by the entropy of the distribution on the decomposition structures of the targeted resolution. Details on the computation of the entropy of the KDWPSSs schemes on the WPSSs of a resolution are given in the appendix II.

Res@depth	Quality	Security/Entropy (iso)	Security/Entropy (isouni)
0	1	114.0	261.6
1	1/2	28.7	65.4
2	1/3	7.5	16.3
3	1/4	2.2	4.1
4	1/5	0.8	1.0
5	1/6	0.0	0.0

TABLE VIII
ENTROPY FOR THE DISTRIBUTION ON RESOLUTIONS OF THE ISOTROPIC COMPRESSION-ORIENTED DISTRIBUTION (g=5, b=1/4, c=0) AND THE UNIFORM DISTRIBUTION

The algorithms for random WPSS generation also introduce a distribution on the decomposition structures of a certain resolution. The question we want to answer is, how good the subsequent resolutions are protected for both the uniform and the compression-oriented distribution on WPSSs.

The tables VIII and IX summarize the results for the isotropic and the anisotropic case, respectively. As we can see, the lower resolutions (at higher depth) are quite easily accessible for an adversary, while the higher resolutions (at lower depth) remain well-protected. If we consider a 512x512 image as source image, the 64x64 image can be decoded with only $2^{H(X)-1} = 2^{2.2-1} \approx 2$ trials on average in a brute-force attack for the compression-oriented isotropic case. The 128x128 image can be decoded with only $2^{H(X)-1} = 2^{7.5-1} \approx 45$ trials on average. The 256x256 image can be decoded with only $2^{H(X)-1} = 2^{28.7-1} = 2^{27.7}$ trials on average. The full resolution image can be considered secure as the entropy is 114, which would result in 2^{113} trials in a brute-force attack. The lower resolutions are better protected for the uniform isotropic case, where the 256x256 image requires already $2^{64.4}$ trials on average, large enough for most adversaries, as each trial is rather expensive. The full resolution is even protected with $2^{261.6}$ bit. Even higher are the entropy values for the anisotropic case, as there are substantially more possible WPSSs. Images with a quality of $\frac{1}{5}$, i.e. only $\frac{1}{16}$ the number of the original pixels (comparable to a 128x128 image in the isotropic case) already have an entropy value of 81.7. The higher resolutions can be considered secure.

D. The Lowest Possible Quality with KDWPSSs in the JPEG 2000 Part 1 Coding Framework

The analysis focuses on isotropic wavelet packet bases, the anisotropic case is analogous, though with a larger number of possible decomposition structures. In JPEG 2000 the smallest subband size is 4x4 (limited by the minimum size of a codeblock). Thus the smallest LL-subband is 4x4 and the corresponding high-pass bands can not be decomposed further as well, i.e., an reconstruction on the basis of at least an 8x8 LL-subband is always possible, i.e., an 8x8 image is always decodeable. For a reconstruction on the basis of 16x16 LL-subband, one needs to determine the next resolution's subbands (LH, HL, and HH) each have two possibilities (decomposed or not decomposed) which results in 8 possibilities

Res@depth	Quality	Security / Entropy (anisouni)
0	1	1321.9
1	1/2	660.5
2	1/3	329.7
3	1/4	164.4
4	1/5	81.7
5	1/6	40.3
6	1/7	19.7
7	1/8	9.3
8	1/9	4.2
9	1/10	1.5
10	1/11	0.0

TABLE IX

Entropy for the distribution of decomposition structures of a resolution as induced by the anisotropic uniform distribution (g=10, b=1/4, c=0)

for the decomposition structures of the subbands. Thus an adversary is capable to decode a 16x16 image. For a reconstruction based on a 32x32 LL-subband, the next resolution's subbands can be decomposed up to a decomposition depth of two, i.e., every subband has 17 possibilities, which adds up to $17^3 = 4913$ possible decomposition structures. If the correct decomposition structure can be identified, a 32x32 image can be decoded (4913 checks are still computationally feasible).

The next resolution (reconstruction of a 64x64 image), each subband has $Q_3 = 17^4 + 1$ possible decomposition structures which results in approximately 2^{49} possibilities, each requires a significant computational effort to test. Thus the reconstruction of a 64x64 image is not efficiently possible.

Every subsequent resolution becomes more complex to decode, i.e., 128x128 image may already require approximately 2^{196} checks, which is currently computationally infeasible.

For all WPSS selection schemes, a 32x32 image can be efficiently reconstructed by an adversary.

E. JPEG 2000 Part 2: Applicable for Encryption with KDWPSSs

JPEG 2000 Part 2 does only allow a two-depth decomposition of a high-frequency subband. In the case of isotropic wavelet packets, i.e., every subband of a resolution is decomposed in both directions, every next resolution may only be secured by $17^3 = 4913$ possibilities. In the case of only horizontal or vertical decompositions of the subbands of a resolution (every decomposition is counted as one level), a subband has only 18 possibilities to be further decomposed, i.e., $18^3 = 5832$ possible decomposition structures at most (for 3 subbands). However, JPEG 2000 Part 2 allows to specify either horizontal, vertical or horizontal and vertical decomposition for a subband, i.e., there are 4 different subband structures for a single subband (no decomposition, vertical, horizontal, or both) which yields less than $4^4+4^2+4^2+1 = 289$ different subband decomposition structures for a depth 2 decomposition (4^4 is the number of possibilities after a decomposition in both directions and 4^2 is the number of possibilities after either a horizontal or a vertical decomposition). There are at most three subbands to consider, which sums up to at most $289^3 = 24137569$ possibilities for three subbands. Though about 2^{24} checks are quite an effort, this number of checks is still computationally feasible.

Thus the application of a JPEG 2000 Part 2 encoder and KDWPSSs can not be considered secure (neither MR-secure nor MQ-secure) due to the restrictions in terms of admissible WPSSs.

V. Computational Demand Comparison to Implementations in JPSEC

JPEG 2000 encryption with key-dependent wavelet packet subband structures can be compared to implementations in JPSEC with similar functionalities [54], [10], [55]. JPSEC allows to apply encryption at several granularity levels. The JPSEC syntax element "granularity" defines the processing order (independently of the actual progression order of the JPEG 2000 codestream) and the granularity level. The granularity level may be component, resolution, layer, precinct, packet, subband, codeblock. JPSEC supports various approaches for transparent encryption [10], [56], [57], [58], the traditional approach is to secure the refinement layers (higher quality layers and higher resolutions), cf. [10], [58], while a base layer in target quality is preserved. With JPSEC it is also possible to implement encryption in a fully transparent fashion, i.e., the encrypted JPEG 2000 codestream remains format-compliant. The format-compliant encryption within JPSEC offers the advantage that the entire encrypted codestream is decodeable with standard JPEG 2000 Part I decoders, while encryption with key-dependent wavelet packet subband structures does not produce format-compliant Part I codestreams. The JPSEC approaches are bitstream-oriented approaches that apply encryption after compression has been performed. The wavelet-packet based approach discussed here is a compression-integrated approach that applies encryption during compression.

The aim of the subsequent experimental comparison is to compare the computational demand of JPEG 2000 based transparent encryption with comparable functionality, on the one hand implemented with key-dependent wavelet packet subband structures, on the other hand implemented in JPSEC. We focus on a scenario where the final bitstream is not yet available, i.e. both stages, compression and encryption need to be conducted.

The following KDWPSSs are evaluated:

- Isotropic:
  - isouni
  - iso
- Anisotropic
  - anisouni
  - aniso

The following processing pipeline is conducted for key-dependent wavelet packet subband structures:

1) Compression with secret wavelet packet subband structure
2) Encryption of wavelet packet subband structure specification (negligible)

3) Decryption of wavelet packet subband structure specification (negligible)
4) Decompression with secret wavelet packet subband structure

Transparent encryption with JPSEC is implemented in two different ways:

- JPSEC with a conventional cipher (further denoted JPSEC1)
- JPSEC and format-compliant encryption routines (further denoted JPSEC2)

JPSEC1 employs conventional encryption (e.g., AES in CTR-mode) and the JPSEC syntax to signal the encrypted parts. The encrypted JPEG 2000 codestream is no longer format-compliant and a JPSEC decoder is needed.

JPSEC2 employs JPEG 2000 specific encryption routines and is capable of preserving format-compliance. The packet body data is encrypted with JPEG 2000 bitstream compliant encryption algorithms [11]. We use the encryption routine sketched in the technology examples of the JPSEC standard and discussed in detail in [54]. An extensive evaluation of format-compliant JPEG 2000 encryption routines can be found in [11].

The processing pipeline for both JPSEC schemes is summarized in the following:

1) Compression
2) Parsing of relevant codestream portions
    - during compression (insignificant runtime complexity)
    - after compression
3) Encryption of relevant codestream portions
4) Parsing of the relevant codestream portion
    - explicitly signalled (insignificant runtime complexity)
    - implicitly signalled (for format compliance)
5) Decryption of relevant codestream portions

A comparable JPSEC-implementation of transparent encryption and encryption with key-dependent wavelet packet subband structures differ in the following:

- Wavelet transform stage (a pyramidal wavelet transform and a wavelet packet transform are employed)
- Post-processing stage (for the JPSEC approaches AES encryption is employed)

All the other processing steps have similar complexity. As long as the subband size does not become smaller than the code-block size the number of codeblocks remains unchanged, and thus the coding and rate allocation have the same complexity.

A. Theoretical performance analysis

In order to compare the computational complexity of these two basic approaches (JPSEC encryption and KDWPSSs) it is sufficient to assess the difference of computational complexity of the pyramidal wavelet transform and the expected complexity of a wavelet packet transform, as well as the complexity of AES encryption. Therefore we determine the computational complexity difference in the transform and encryption stage on a random access machine with an instruction set of basic operations, such as `^`, `&`, `+`, `*` and `%`.

1) Comparison of the Wavelet Transform Stage: The wavelet transform stage consists of iterated filter operations, the number of filter operations depends on the WPSS and the number of pixels. For every input pixel and decomposition level two filter operations are required for the isotropic case and one filter operation is required for the anisotropic case. A single filter operation with an n-length filter consists of n multiplications, n additions, n MemReads and a single MemWrite, which yields 25 operations for $n = 8$ (JPEG 2000's 9/7 irreversible filter) and 13 operations for $n = 4$ (JPEG 2000's 5/3 reversible filter).

In our further analysis we will determine the expected average decomposition depth of a WPSS drawn according to one of the proposed distributions, which enables us to determine the expected average number of filter operations per pixel, i.e., the expected computational complexity. For the isotropic wavelet packet transform two filter operations are required for one decomposition step (per pixel / coefficient). For the anisotropic wavelet packet transform one filter operation is required for one decomposition step (per pixel / coefficient).

For the pyramidal wavelet decomposition with decompostion depth g, the average decomposition depth D_g^p is given by:

$$D_g^p = \sum_{i=0}^{g} 1/4^i =^{g \to \infty} 4/3 \approx 1.33333$$

The expected average decomposition depth for the uniform distribution on isotropic WPSS with a maximum decomposition depth g, D_g can be derived recursively:

$$D_g = \frac{Q_g - 1}{Q_g}(D_{g-1} + 1)$$

with $D_0 = 0$. Most notably, the following holds:

$$g - 1 \leq D_g \leq g - 1/2$$

$$D_{k+1} \approx k + 0.41174$$

.

The expected decomposition depth for the compression-oriented distribution is denoted D_g^c. The distribution is generated by a randomized algorithm (see listing 1), which is determined by the parameters b (base value) and c (change factor) which can be used to adjust the probability p_l, which gives the probability of the event that a subband at level l is further decomposed.

$$p_l = 1 - \frac{b + cl}{2}$$

The probability of the converse event is $q_l = 1 - p_l$. $D_{l,g}^c$ gives the expected decomposition depth of a subband at level l with a maximum decomposition depth of g.

$$D_g^c = D_{0,g}^c$$

$$D_{g,g}^c = 0$$

$$D_{l,g}^c = (D_{l+1,g}^c + 1)p_l = \sum_{i=l}^{g-1} \prod_{k=l}^{i} p_k$$

If $c = 0$ and $1 - \frac{b}{2} < 1$, then:

$$D_g^c =^{g\rightarrow\infty} \frac{2}{b} - 1$$

If a minimal decomposition depth n of the approximation subband is also requested, the calculation of the expected average decomposition depth becomes more complex. We only consider the case, where n equals the maximum decomposition depth g, and denote the corresponding expected average decomposition depth D_g^f.

$$D_g^f = \frac{g}{4^{g-1}} + \sum_{i=1}^{g-1} \frac{3}{4^i}(D_{i,g}^c + i)$$

If $c = 0$ and $0 < 1 - \frac{b}{2} < 1$, then:

$$D_g^f =^{g\rightarrow\infty} \frac{2}{b} + 1/3$$

The proposed settings of section II-B.2 ($b = 1/4$, $c = 0.1$) yield $D_5^f = 3.39925$.

The expected decomposition depth for a uniform distribution on anisotropic WPSSs is denoted D_g^a and can be determined by the following recursion:

$$\forall i \in \mathbb{N} : A_{-i} = 0$$

$$A_g = 1 + 2A_{g-1}^2 - A_{g-2}^4$$

$$\forall i \in \mathbb{N} : D_{-i}^a = 0$$

$$D_g^a = \frac{1}{A_g}(2A_{g-1}^2(D_{g-1}^a + 1) - A_{g-2}^4(D_{g-2}^a + 2))$$

$$D_g^a \approx g - 1 + 0.5301$$

The expected decomposition depth for the compression-oriented distribution is denoted D_g^{ac}. $D_{l,g}^{ac}$ gives the expected decomposition depth of a subband at level l with a maximum decomposition depth of g.

$$D_g^{ac} = D_{0,g}^{ac}$$

$$D_{g,g}^{ac} = 0$$

$$D_{l,g}^{ac} = (D_{l+1,g}^{ac} + 1)p_l = \sum_{i=l}^{g-1} \prod_{k=l}^{i} p_k$$

If $c = 0$ and $0 < 1 - \frac{b}{2} < 1$, then:

$$D_g^{ac} =^{g\rightarrow\infty} \frac{2}{b} - 1$$

In summary, if a WPSS is chosen according to a uniform distribution on the WPSSs, one can expect a high average decomposition depth, which introduces additional computational complexity compared to the pyramidal wavelet transform. The compression-oriented distribution on isotropic WPSSs only slightly reduces the decomposition depth, while the compression oriented distribution on anisotropic WPSSs greatly reduces the decomposition depth (see tables X and XI).

g	D_g	D_g^c	D_g^f
1	0.5	0.857	1
2	1.41176	1.6406	1.90625
3	2.41174	2.3105	2.70703
4	3.41174	2.89673	3.40967
5	4.41174	3.40946	4.02496

TABLE X

THE EXPECTED AVERAGE DECOMPOSITION DEPTH FOR ISOTROPIC WAVELET TRANSFORM DISTRIBUTIONS ($b = 1/4$, $c = 0$)

g	D_g^a	D_g^{ac}
2	1.55556	1.64063
4	3.53125	2.89673
6	5.53020	3.85843
8	7.53014	4.59474
10	9.53014	5.15847

TABLE XI

THE EXPECTED AVERAGE DECOMPOSITION DEPTH FOR ANISOTROPIC WAVELET TRANSFORM DISTRIBUTIONS ($b = 1/4$, $c = 0$)

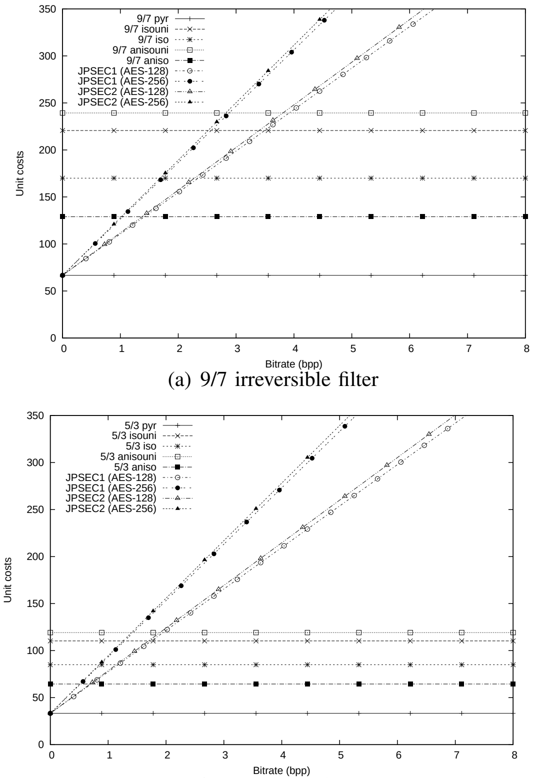
2) Cost of AES-Encryption: Conventional encryption approaches, e.g., implemented in JPSEC, rely on a secure cipher, the current state-of-the-art is AES. Most commonly AES will be employed in a stream-cipher mode of operation, such as CTR-mode. Thus in order to encrypt the JPEG 2000 codestream, a key stream will be generated with AES, which is then XORed with plaintext, which introduces two additional MemRead and one additional MemWrite operation for each encrypted byte. Table XII summarizes the computational complexity of the AES, where we consider every memory write and read operation as well as basic operations, such as `^`, `&`, `+`, and `%` [59]. In conclusion there are 352.625 operations necessary for the encryption of a single byte with AES with a 128-bit key in CTR-mode, 425.625 for AES with a 192-bit key, and 479.625 for AES with a 256-bit key.

3) Comparison of Wavelet Transform Stage and AES Encryption: The KDWPSSs approach has constant complexity for all bit rates, as the bitrate is determined by quantization after the wavelet transform, while the computational complexity of the JPSEC1 (conventional encryption) and JPSEC2 (format-compliant encryption) approach directly depends on the bit rate, i.e., the bit per pixel (bpp). In order to compare

name	`MemRead`	`MemWrite`	`^ & + %`
`Substitute`	48	16	0
`ShiftRows`	80	32	32
`MixColumns`	136	32	144
`AddKey`	32	16	16
128 bit key, total	2858	944	1792
192 bit key, total	3450	1136	2176
256 bit key, total	4042	1328	2304

TABLE XII

NUMBER OF BASIC OPERATIONS FOR AES ENCRYPTION [59]



(a) 9/7 irreversible filter

(b) 5/3 reversible filter

Fig. 3. Comparison of computational complexity

filter	pyr	isouni	iso	anisouni	aniso
9/7	66.6	220.6	170.0	238.3	129.0
5/3	33.3	110.3	85.0	119.2	64.5

TABLE XIII

EXPECTED COMPUTATIONAL COMPLEXITY OF THE PYRAMIDAL AND KDWPSSS PER PIXEL (IN UNIT COSTS)

approach	4bpp	2bpp	1bpp	0.5bpp
JPSEC1 (AES-128)	176.31	88.16	44.08	22.04
JPSEC1 (AES-192)	212.81	106.41	53.20	26.60
JPSEC1 (AES-256)	239.81	119.91	59.95	29.98
JPSEC2 (AES-128)	181.31	90.66	45.33	22.67
JPSEC2 (AES-192)	217.81	108.91	53.45	27.23
JPSEC2 (AES-256)	244.81	122.41	61.20	30.61

TABLE XIV

COMPUTATIONAL COMPLEXITY OF JPSEC APPROACHES WITH AES ENCRYPTION PER PIXEL (IN UNIT COSTS)

the computational complexity of the JPSEC approaches and KDWPSSs, we assign all operations, e.g., addition, multiplication, MemRead, and MemWrite, the same complexity cost. In the wavelet transform stage the KDWPSSs approaches have different computational complexity, the JPSEC approaches employ pyramidal wavelet decomposition which results in average decomposition depth of $\approx$ 1.33, while the average decomposition depth of the KDWPSSs approach depends on the WPSS selection scheme. The expected computation complexity of the different KDWPSSs schemes for a pixel (in unit costs) is given in table XIII, which is derived by multiplying the expected average decomposition depth with the number of necessary operations for a filter operation. The JPSEC approaches use AES encryption and the computational complexity in dependency of the number of pixel relies on the bitrate, i.e., the coded bits per pixel. The computational complexity of the JPSEC approaches in unit costs per pixel is given in table XIV. In figure 3(a) the computational complexity of the different JPEG 2000 encryption approaches considering the 9/7 filter is shown for a varying bitrate (in unit costs for the encryption of a single pixel). Figure 3(b) illustrates the computational complexity for the 5/3 filter. Considering the 5/3 filter, the JPSEC1 (AES-256) approach becomes more expensive than the compression-oriented anisotropic key-dependent wavelet packet scheme for bitrates in excess of 0.52 bpp and the JPSEC1 (AES-256) approach becomes even more expensive than the isotropic scheme for bitrates in excess of 0.86 bpp. Considering the 9/7 filter, the JPSEC1 (AES-128) approach becomes more expensive than the compression-oriented anisotropic key-dependent wavelet packet scheme for bitrates in excess of 1.42 bpp and the JPSEC1 (AES-128) approach becomes even more expensive than the isotropic compression-oriented scheme for bitrates in excess of 2.35 bpp. In summary key-dependent wavelet packets have advantages for higher bitrates, while the JPSEC1 approach has to be preferred for low bitrates.

The JPSEC2 approach depends on the computational complexity of the bitstream-compliant encryption algorithm, which require additional operations. For the technology example of the JPSEC standard as given in detail in [54] there are 10 additional operations necessary. Thus, for the JPSEC2 approach the bitrates at which the key-dependent wavelet packet schemes are more efficient are slightly lower.

Our theoretical performance analysis gives the computational complexity for an idealized model of computation (unit cost for all operations). However, for a practical application the theoretical analysis can only give an estimate of what can be actually achieved in terms of runtime. In the following we present "real" experimental results, however, these results are heavily biased by implementation details, most importantly the KDWPSSs implementation is based on the Java reference implementation of JPEG 2000, while the AES and parsing is conducted by software written in C, which favors the JPSEC approaches.

B. Experimental performance evaluation

The employed JPEG 2000 implementation has been JJ2000. AES encryption routines are used as described in [54], [11]. The evaluation database consisted of 100 images (512x512, 8bpp, grey scale). As the target bitrate has an insignificant influence on JJ2000's compression complexity, only a rate of 2bpp is considered. For each wavelet packet subband structure scheme 100 secret wavelet packet subband structures with a maximum decomposition level of 5 have been randomly chosen.

structure	compression		decompression	
pyramidal	0.643 s	1.55 fps	0.404 s	2.47 fps
iso	1.005 s	1 fps	0.962 s	1.04 fps
isouni	1.147 s	0.87 fps	1.011 s	0.99 fps
aniso	0.726 s	1.34 fps	0.416 s	2.40 fps
anisouni	1.891 s	0.53 fps	0.924 s	1.08 fps

TABLE XV
COMPRESSION COMPLEXITY

JPSEC encryption			
Method	throughput	codestreams with 2bpp	
JPSEC1	42.71 MB/s	0.0015 s	683.36 fps
JPSEC2	37.81 MB/s	0.0017 s	604.96 fps

TABLE XVI
RUNTIME PERFORMANCE OF ENCRYPTION ROUTINES

The average compression / decompression time for secret wavelet packet subband structures and pyramidal decomposition is given in table XV. Note the significant increase in compression complexity for the wavelet packet subband structures, which is in line with our theoretical analysis.

The parsing for JPEG 2000 packets, which is needed for the JPSEC2 approach, is very lightweight. The average time of header decoding is 0.0027 s (370.92 fps) and the average time for SOP/EPH parsing is 0.0010 s (1030.93 fps), which is only applicable if SOP/EPH markers are employed.

The encryption routines for JPSEC are very lightweight as well (see table XVI), although we encrypt the entire packet body data here (for actual transparent encryption only 90% or even less of the data needs to be encrypted).

Overall, we note that JPSEC parsing and encryption effort is orders of magnitude lower as compared to performing compression with pyramidal or other wavelet packet structures. Therefore, when considering JPSEC encryption routines plus pyramidal JPEG 2000 compression, the JPSEC approach is about 1.5 - 2 times as fast on average as compared to performing compression (and encryption) with random wavelet packet subband structures with our implementation.

The above results are based on our implementation, in the following we give results for other implementations. In order to highlight the dependence on the actual implementations we present a comparison with two other JPEG 2000 implementations: the JasPer implementation compresses at 12.89 fps and decodes at 21.45 fps [11], while the Kakadu implementation [3] even achieves 39.88 fps for compression and 60.22 fps for decompression (both with 2bpp, 5 level wavelet decomposition and no quality layers).

Most interestingly the Kakadu implementation also offers JPEG 2000 Part 2 features, i.e., arbitrary WPSSs are supported to some extent. The deepest possible decomposition has an average decomposition depth of $\approx$ 3.33, i.e., 6.66 filter operations per coefficient, which is close to the compression oriented distributions on WPSSs. The Kakadu implementation achieves 24.05 fps for compression and 40.06 fps for decompression, which is also approximately 1.5-2 times slower than the pyramidal decomposition case.

C. Interpretation of results

Our analytical results indicate that secret wavelet packet structures offer performance advantages for high bitrates compared to the JPSEC approaches. The experimental results, however, reveal that our actual implementation does not follow our theoretical analysis. There are two main reasons for this difference between theory and practice:

1) Different implementations are compared, e.g., a Java implementation (wavelet packets in JJ2000) and a C++ implementation (Kakadu) are compared to a C implementation.
2) The global memory access patterns (the entire image) of wavelet packet transforms have to compete against the extremely local memory access patterns of the JPSEC / AES implementation.

The first reason could only be resolved by rewriting the wavelet packet code in an highly optimized JPEG 2000 implementation, though the achievable performance gain is limited (wavelet packets in the faster and optimized Kakadu implementation are still not competitive to an implementation in JPSEC). The major factors are memory access patterns, which play a decisive role in the performance of the wavelet transform [60]. The effect of memory access patterns is highly dependent on the cache sizes of the CPU, i.e., if the cache is large enough to contain the entire image, the impact of memory access patterns becomes less significant.

VI. DISCUSSION

Media encryption schemes relying on secret transform domains have been proposed mainly motivated by the significant reduction of the computational demand for encryption as compared to traditional transparent encryption methods and by potential capabilities in encrypted domain signal processing. A thorough analysis of the properties of the KDWPSSs approach has revealed that in fact many of these advantages are correct only under very specific preconditions and that significant disadvantages as compared to traditional encryption schemes do exist.

- **Compression impact**: The compression-oriented approach for lightweight encryption with KDWPSSs obviously impacts on compression performance. If all possible subband structures were used, the approach would not be suitable for application due to the high variance of obtained compression results. The approach of pruning the set of all subband structures (compression oriented distribution) is successful in controlling the loss in compression performance to an acceptable level, however, still with this technique the loss exists. On the other hand, techniques based on JPSEC (like JPSEC1 and JPSEC2 as discussed in this work) of course do not at all influence compression performance.

[3] Linux binaries in version 6.3.1 from http://www.kakadusoftware.com

- **Security**: From a security point of view, the JPSEC1 approach is superior as it relies on state-of-the-art cryptography. In case of the JPSEC1 scheme, only the file length and a part of the JPEG 2000 main header is preserved, the entire image data is securely encrypted. Considering MQ security, the computation of a high quality reconstruction of the original data is impossible on the basis of the file length. The security of the JPSEC2 approach is discussed in detail in [11], the packet headers leak almost no visual information, certainly not in a good quality. The KDWPSSs schemes cannot prevent access to lower resolutions and are thus less secure in terms MQ-security.
- **Computational demand**: As opposed to the reduction of computational amount in terms of encryption, additional computations are introduced in the *compression* algorithm. A theoretical analysis shows potential benefits of the KDWPSS approach in case of high quality (high bitrates).

  Experimental results show that current implementations cannot achieve the computational benefits shown in the theoretical analysis. Therefore, from a runtime performance point of view transparent encryption with wavelet packet subband structures currently does not offer advantages over the JPSEC technique.

  However, there are specific application scenarios and eventual future developments where this approach still has to be considered competitive:

  1) If a random wavelet packet decomposition has to be conducted anyway for other reasons, the transparent encryption functionality comes for free. Random wavelet packet subband structures have been proposed to implement key-dependency and security in watermarking schemes [31], [32], [33] or to enable multiple re-watermarking [34]. For this application scenario, using the wavelet-packet subband structure encryption approach comes at virtually no cost, while JPSEC based encryption has to be performed in addition to the watermarking stage. For example, a content provider transmits video transparently encrypted with the KDWPSS technique and additionally embeds an annotation watermark, providing metainformation about the video transmitted. Only the customer willing to pay for the decryption key (i.e. the decomposition structure) will be able to extract the watermark information and the high quality video data. Other watermarks (eventually containing copyright or fingerprinting information) can be additionally embedded with different decomposition structures or the classical DWT avoiding inference with the embedded annotation mark.
  2) If the cost of arbitrary wavelet decompositions and the corresponding compression pipeline can be substantially reduced, the relation between compression and encryption routines in terms of computational effort changes and therefore in such a case the entire comparison has to be reconsidered. The simple algorithmic structure of the wavelet packet transforms enables highly parallel implementations, which may shift the performance advantages towards encryption with wavelet packet subband structures. Also, the increase of cache sizes also favors the wavelet packet transform part.
  3) If the cost of the encryption routines increases we again have to reconsider the relation between compression and encryption and we result in the same situation as described before. For example, this will happen if we need to increase the number of rounds for AES encryption, as this has been the case with DES and tripleDES.
- **Encrypted domain processing**: It is a significant misinterpretation that the encrypted domain is a transform domain. In fact, the encrypted domain is a JPEG 2000 bitstream and only signal processing operations that can be conducted in this domain are possible. For example, it is possible to apply rate-distortion optimal rate adaptation (in case the underlying bitstream is in quality progression order), in case SOP and EPH headers are used (all JPEG2000 packet-based), even more sophisticated adaptation can be done. Signal processing operations which rely on transform coefficient data cannot be applied to KDWPSS protected data, since if transform data were available, the security of the scheme is immediately compromised.

Given these facts, it has to be stated that apart from highly specialised application scenarios involving wavelet packet-based watermarking or eventual future developments in hardware design and complexity of conventional encryption schemes, hardly any sensible realistic application scenarios can be identified for KDWPSS-based transparent encryption at the present time.

VII. Conclusion

A primary argument for proposing KDWPSS-based transparent encryption has always been its “lightweight” nature for encryption, introduced by shifting complexity from encryption into the compression pipeline.

In our comparison, we have taken the standpoint of a *joint* compression and encryption algorithm. When comparing JPEG 2000 encryption with key-dependent wavelet packet subband structures (KDWPSSs) to JPSEC based approaches under this view, the overhead through additional complexity introduced in the compression step is significant. The theoretical analysis shows runtime performance advantages of KDWPSSs only for high bitrates. Empiric performance evaluation reveals that furthermore these runtime performance advantages cannot be achieved with state-of-the-art JPEG 2000 implementations.

Signal processing in the encrypted domain, often used as a second argument favoring this approach, can be applied only in a very restricted sense (and not based on transform coefficient data).

The security of JPEG 2000 encryption with KDWPSSs has been analyzed in depth and has been found to be less secure than encryption with JPSEC, as smaller resolutions remain accessible. This makes encryption schemes based on KDWPSS suited for transparent encryption only. The high resolution image data can be considered securely protected.

All these facts taken together with a slight decrease in compression efficiency as compared to classical (pyramidal) JPEG 2000 make KDWPSS-based encryption approaches suited for actual application under very rare and specialised conditions only.

Appendix I
Details on the Entropy Computation of WPSS Distributions

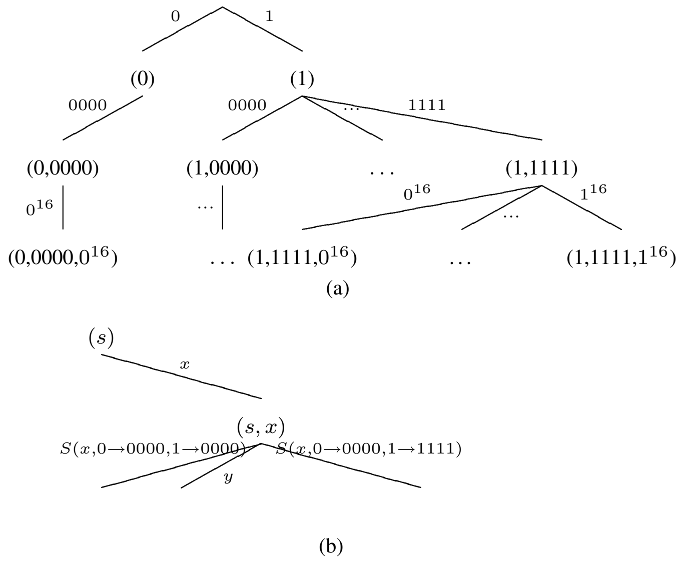
A WPSS (wavelet packet subband structure) is derived by the following randomized algorithm (see section II-B.2): every subband at depth l is further decomposed with a certain probability p_l, which may only depend on the depth l.

Game Tree

This randomized algorithm can be illustrated with the corresponding *game tree*. A game tree $\mathcal{G}_g$ is the tuple (V, E, l, p):

V ...set of vertices
$E \subset V \times V$ ...set of edges
$l : E \rightarrow \{0,1\}^*$
$p : E \rightarrow \mathbb{R}^+$

The vertices of a game tree correspond to a certain WPSS. The edge label (l) in a game tree indicates decompostion decisions and is associated with the probability p that this decompostion decisions are selected in the randomized algorithm. In figure 4(a) a game tree is illustrated, showing the edge labels and the vertices. The first decision is whether the entire image gets decomposed (split into four subbands), there are two outcomes, an edge labelled with a ”0” indicates no decomposition, an edge labelled with a ”1” indicates a decomposition into four distinct subbands. A ”0” in the label of a edge is replaced by ”0000” in the label of next edge, in order to obtain same length labels at a certain depth of the tree. A ”1” in the label of a edge is replaced by one of the strings ”0000”, ..., ”1111” in the label of the next edge, which indicates the further decompositions of the 4 subbands. I.e., the string ”0000” indicates that no subband is decomposed, the string ”0001” indicates that the last subband (HH) is decomposed, ..., and the string ”1111” indicates that all subbands are decomposed. A label uniquely identifies a further decomposition. Note that 0^{16} denotes the string 0000000000000000 and analogous is the meaning of 1^{16}. A unique code for a node is derived by a separated concatenation of the edge labels from a path from the root to the node. We use ”,” as a separator. Each node corresponds to a certain WPSS. Figure4(b) shows how the edges and edge labels are determined by the predecessor edge. The function $pre : E \rightarrow E$ gives the predecessor edge of an edge, e.g., in figure 4(b) the predecessor of the edge labelled y, $l(e) = y$, is the edge labelled x, i.e., $l(pre(e)) = x$. $S(x, r_1, \ldots r_n)$ denotes the string obtained by applying the substitution rules r_1 to r_n to string x. If there is a complex rule, which allow choices, i.e., the right side of one rule is a set of strings, $S(x, r_1, \ldots, r_n)$ denotes the set of all possible substitutions.



(a)

(b)

Fig. 4. Random generation of isotropic WPSS, the game tree

In figure 4(b) new edges are added to the (s, x) depending on the edge label x: the label of the first edge is obtained by substituting every ”0” in x by ”0000” and every ”1” by ”0000”. The labels of all outgoing edges of (s, x) are obtained by all possible substitutions of a ”1” in x. The last edge (see figure 4(b)) is obtained by substituting all ”1”s by ”1111”. A predecessor label determines the edges and edge labels in the following way:

$l(e) = y$
$l(pre(e)) = x$
$x \in \{0,1\}^n$
$x = (x_1, \ldots, x_n), \qquad x_i \in \{0,1\}$
$y \in S(x, 0 \rightarrow 0000, 1 \rightarrow \{0000, \ldots, 1111\}) \subset \{0,1\}^{4n}$

The function p assigns each edge a probability (the probabilities of the outgoing edges of a node sum up to 1). The probability of an edge can be determined by its label y and the label of its predecessor x, by simply considering the number of actual decompostion decisions ($\sum y_i$) and the number of maximally possible decomposition decisions ($4 \sum x_i$):

$l(e) = y$
$l(pre(e)) = x$
$p(y) = p_l^{\sum y_i}(1 - p_l)^{4\sum x_i - \sum y_i}$

Every leaf of a game tree with depth g corresponds to exactly one WPSS ψ, the probability of a WPSS ψ is derived by the product of the edge weights p of the path from the root to the leaf.

$\psi \in V : \qquad p(\psi) = \Pi_{e \in \text{Path}(\text{root},\psi)} p(e)$

We denote the entropy of corresponding distribution for a game tree $\mathcal{G}_g$ by:

$$H(\mathcal{G}_g) = \sum_{\psi \in \text{Leaves}(\mathcal{G}_g)} -p(\psi) \, \text{ld} \, p(\psi)$$

However, as the number of leaves at depth g is $Q(g)$ the computation of the entropy of the distribution on WPSSs on the basis of this formula is soon infeasible with growing g.

Cumulative Game Tree

A simpler representation of a game tree $\mathcal{G}_g$ is its corresponding *cumulative game tree* (CuGa-Tree), $\mathcal{C}_g$. A CuGa-Tree $\mathcal{C}_g$ is the tuple (V, E, l, p, n):
$l : E \to \mathbb{N}$
$p : E \to \mathbb{R}^+$
$n : E \to \mathbb{N}$
A CuGa-Tree summarizes the edges of a node with the with the same probability p, i.e., with the same number of decomposition decisions, i.e., with the same number of "1"s in the edge label of the game tree. Thus the edge label of a CuGa-Tree indicates the number of decomposition decisions, i.e., the number of subbands which are further decomposed. We have to keep track how many edges of the game tree are summarized by an edge of a CuGa-Tree, therefore we introduce a weight function $n : E \to \mathbb{N}$. A CuGa-Tree with depth 2 is shown in figure 5.
The edges and edge labels are determined by the predecessor edge (see figure 5(b)): the successors of an edge with label $l(e) = i$ (this number of subbands have been decomposed) can be in the range of 0 to $4i$, as every subband may have up to four children:
$l(pre(e)) = i$
$l(e) \in \{0, \ldots, 4i\}$
The probability for an edge is similar to game trees:
$p(e) = p_l^{l(e)}(1-p_l)^{4l(pre(e))-l(e)}$
The number of edges in the game tree with the same probability is derived by counting the number of edges with the same number of "1"s, i.e., decomposition decisions in the edge label: There are $4l(pre(e))$ possible positions for $l(e)$ "1"s and thus there are $\binom{4l(pre(e))}{l(e)}$ edges with th same probability in the game tree:
$n(e) = \binom{4l(pre(e))}{l(e)}$
The probability p and weight n are defined for vertices in the following way:
$\psi \in V : p(\psi) = \Pi_{e \in \mathrm{Path}(\mathrm{root},\psi)} p(e)$
$\psi \in V : n(\psi) = \Pi_{e \in \mathrm{Path}(\mathrm{root},\psi)} n(e)$

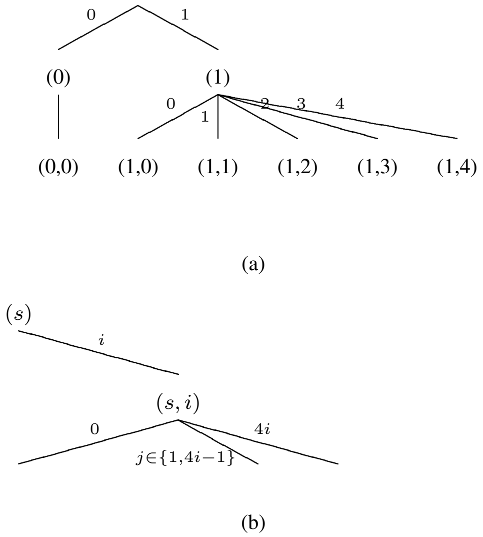
The entropy of the corresponding distribution of a CuGa-Tree $\mathcal{C}_g$ can be computed by:

$$H(\mathcal{C}_g) = \sum_{\psi \in \mathrm{Leaves}(\mathcal{C}_g)} -n(\psi)p(\psi)\mathtt{ld}p(\psi)$$

The nodes can be uniquely identified by the path from the root, i.e., by tuple of edge labels. A node at depth g is a g-tuple of edge labels $(x_1, \ldots, x_g)$. The set of all nodes at depth g is given by $\{(x_1, \ldots, x_g) | x_1 \in \{0,1\}, x_{i+1} \leq 4x_i\}$.

Tree Recursion

However, the recursive structure of the CuGa-Tree and the properties of the entropy [48] allow a "tree" recursive computation of the entropy of a CuGa-Tree, respectively its corresponding distribution on WPSSs. For the "tree" recursive definition we introduce partial CuGa-Trees, denoted $\mathcal{C}^v_{g,m}$. In figure 6 a partial CuGa-Tree $\mathcal{C}^v_{g,m}$ is illustrated and defined, it starts at vertex v and has edges labelled "0" to "m". At each edge there is a partial CuGa-Tree with a reduced depth of $g-1$. A partial CuGa-Tree $\mathcal{C}^v_{0,m}$ is equivalent to v, this is the base case of the recursion. A CuGa-Tree $\mathcal{C}_g$ is equivalent to the partial CuGa-Tree $C^{\epsilon}_{g,1}$, i.e., the "tree" recursion leads to the same tree as the "edge" recursion.



(a)

(b)

Fig. 5. Recursive "edge" definition of CuGa-Trees

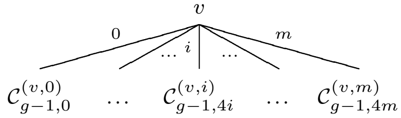

v
0
$\ldots\ i$
$\ldots$
m
$\mathcal{C}^{(v,0)}_{g-1,0}$
$\ldots$
$\mathcal{C}^{(v,i)}_{g-1,4i}$
$\ldots$
$\mathcal{C}^{(v,m)}_{g-1,4m}$

Fig. 6. Recursive "tree" definition of CuGa-Trees: $\mathcal{C}^v_{g,m}$

The entropy of a depth one CuGa Tree with m children and decomposition probability p is calculated by:

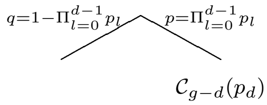
$$H(\mathcal{C}^v_{1,m}) = \sum_{i=0}^{m} \binom{m}{i} p^i(1-p)^{m-i}(-1)\mathtt{ld}p^i(1-p)^{m-i}$$

The entropy of CuGa-Trees can be computed recursively by:

$$H(\mathcal{C}^v_{g,m}) = H(\mathcal{C}^v_{(1,m)}) + \sum_{i=0}^{m} \binom{m}{i} p^i(1-p)^{m-i} H(\mathcal{C}^{(v,i)}_{g-1,4i})$$

Appendix II
Details on the Entropy Computation of Distributions of Decomposition Structures on Resolutions

In order to assess the MQ-security of KDWPSS we need to compute the entropy of the resulting distribution of the decomposition structures on a resolution, i.e., on the subband is the result of always decomposing the low pass band further (no high pass filtering, i.e., either the LL, LX or XL subband). Thus only the case of a low pass band decomposition is of interest up to the depth d of the targeted resolution (see figure 7). The entropy of the decomposition structures of a resolution corresponds to the entropy of the tree of figure 7, the depth of the resolution has to be considered for the split-probability in sub-tree $\mathcal{C}_{g-d}$, which is indicated by the notation $\mathcal{C}_{g-d}(p_d)$. Thus entropy computation is straight-forward, namely the



(a) $\mathcal{R}_{g,d}$

Fig. 7. The entropy of distributions of decomposition structures on resolutions

entropy of the decomposition structures for a resolution d can be computed by:

$$q = 1 - \Pi_{l=0}^{d-1} p_l$$

$$p = \Pi_{l=0}^{d-1} p_l$$

$$H(\mathcal{R}_{g,d}) = q\,\mathtt{ld}1/q + p\,\mathtt{ld}1/p + pH(\mathcal{C}_{g-d}(p_d))$$

Acknowledgments

The authors thank Stefan Huber, Rade Kutil, and Roland Kwitt for the valuable discussions and contributions.

References

[1] B. M. Macq and J.-J. Quisquater, “Cryptology for digital TV broadcasting,” *Proceedings of the IEEE*, vol. 83, pp. 944–957, June 1995.

[2] ISO/IEC 15444-1, “Information technology – JPEG2000 image coding system, Part 1: Core coding system,” Dec. 2000.

[3] D. Taubman and M. Marcellin, *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[4] Digital Cinema Initiatives, LLC (DCI), “Digital cinema system specification v1.1.” online presentation, Apr. 2007.

[5] R. Grosbois, P. Gerbelot, and T. Ebrahimi, “Authentication and access control in the JPEG2000 compressed domain,” in *Applications of Digital Image Processing XXIV* (A. Tescher, ed.), vol. 4472 of *Proceedings of SPIE*, (San Diego, CA, USA), pp. 95–104, July 2001.

[6] 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)*, vol. III, (Barcelona, Spain), pp. 205–208, Sept. 2003.

[7] T. Stütz and A. Uhl, “On format-compliant iterative encryption of JPEG2000,” in *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM’06)*, (San Diego, CA, USA), pp. 985–990, IEEE Computer Society, Dec. 2006.

[8] M. Grangetto, E. Magli, and G. Olmo, “Multimedia selective encryption by means of randomized arithmetic coding,” *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905–917, 2006.

[9] 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.

[10] T. Stütz and A. Uhl, “On efficient transparent JPEG2000 encryption,” in *Proceedings of ACM Multimedia and Security Workshop, MM-SEC ’07*, (New York, NY, USA), pp. 97–108, ACM, Sept. 2007.

[11] D. Engel, T. Stütz, and A. Uhl, “A survey on JPEG2000 encryption,” *Multimedia Systems*, vol. 15, no. 4, pp. 243–270, 2009.

[12] ISO/IEC 15444-8, “Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000,” Apr. 2007.

[13] ITU-T T.807, “Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000,” Mar. 2006.

[14] B. Macq and J. Quisquater, “Digital images multiresolution encryption,” *The Journal of the Interactive Multimedia Association Intellectual Property Project*, vol. 1, pp. 179–206, Jan. 1994.

[15] T. Stütz, V. Pankajakshan, F. Autrusseau, A. Uhl, and H. Hofbauer, “Subjective and objective quality assessment of transparently encrypted JPEG2000 images,” in *Proceedings of the ACM Multimedia and Security Workshop (MMSEC ’10)*, (Rome, Italy), ACM, Sept. 2010.

[16] 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*, vol. V, (Toulouse, France), pp. 465–468, IEEE, May 2006.

[17] J. Fridrich, A. C. Baldoza, and R. J. Simard, “Robust digital watermarking based on key-dependent basis functions,” in *Information hiding: second international workshop* (D. Aucsmith, ed.), vol. 1525 of *Lecture notes in computer science*, (Portland, OR, USA), pp. 143–157, Springer Verlag, Berlin, Germany, Apr. 1998.

[18] 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*, (Las Vegas, NV, USA), pp. 237–243, June 1999.

[19] M. Cancellaro, M. Carli, K. Egiazarian, and J. Astola, “Perceptual data hiding in tree structured haar transform domain,” in *Security, Steganography, and Watermarking of Multimedia Contents IX* (E. J. Delp and P. W. Wong, eds.), Proceedings of SPIE, (San Jose, CA, USA), pp. 65051Q1–65051S11, SPIE, Jan. 2007.

[20] 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)*, vol. 83, pp. 2095–2116, Oct. 2003.

[21] W. Dietl, P. Meerwald, and A. Uhl, “Key-dependent pyramidal wavelet domains for secure watermark embedding,” in *Proceedings of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V* (E. J. Delp and P. W. Wong, eds.), vol. 5020, (Santa Clara, CA, USA), pp. 728–739, SPIE, Jan. 2003.

[22] P. Meerwald and A. Uhl, “Watermark security via wavelet filter parametrization,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP’01)*, vol. 3, (Thessaloniki, Greece), pp. 1027–1030, IEEE Signal Processing Society, Oct. 2001.

[23] I. Djurovic, S. Stankovic, and I. Pitas, “Digital watermarking in the fractional fourier transformation domain,” *Journal of Network and Computer Applications*, vol. 24, pp. 167–173, 2001.

[24] G. Unnikrishnan and K. Singh, “Double random fractional fourier-domain encoding for optical security,” *Optical Engineering*, vol. 39, pp. 2853–2859, Nov. 2000.

[25] L. Vorwerk, T. Engel, and C. Meinel, “A proposal for a combination of compression and encryption,” in *Visual Communications and Image Processing 2000*, vol. 4067 of *Proceedings of SPIE*, (Perth, Australia), pp. 694–702, June 2000.

[26] 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*, (Victoria, Canada), pp. 1–4, IEEE Signal Processing Society, Aug. 2001.

[27] A. Pommer and A. Uhl, “Lightweight protection of visual data using high-dimensional wavelet parametrization,” in *Image Analysis and Processing - ICIAP 2005* (F. Roli and S. Vitulano, eds.), vol. 3617 of *Lecture Notes on Computer Science*, (Cagliari, Italy), pp. 645–652, Springer-Verlag, Sept. 2005.

[28] A. Uhl and A. Pommer, “Are parameterised biorthogonal wavelet filters suited (better) for selective encryption?,” in *Multimedia and Security Workshop 2004* (J. Dittmann and J. Fridrich, eds.), (Magdeburg, Germany), pp. 100–106, Sept. 2004.

[29] A. Pommer and A. Uhl, “Selective encryption of wavelet packet subband structures for secure transmission of visual data,” in *Multimedia and Security Workshop, ACM Multimedia* (J. Dittmann, J. Fridrich, and P. Wohlmacher, eds.), (Juan-les-Pins, France), pp. 67–70, Dec. 2002.

[30] A. Pommer and A. Uhl, “Selective encryption of wavelet-packet encoded image data — efficiency and security,” *ACM Multimedia Systems (Special issue on Multimedia Security)*, vol. 9, no. 3, pp. 279–287, 2003.

[31] W. Dietl and A. Uhl, “Watermark security via secret wavelet packet subband structures,” in *Communications and Multimedia Security. Proceedings of the Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security* (A. Lioy and D. Mazzocchi, eds.), vol. 2828 of *Lecture Notes on Computer Science*, (Turin, Italy), pp. 214–225, Springer-Verlag, Oct. 2003.

[32] 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.

[33] M. Brachtl, W. M. Dietl, and A. Uhl, “Key-dependency for a wavelet-based blind watermarking algorithm,” in *ACM Multimedia and Security Workshop* (J. Dittmann and J. Fridrich, eds.), (Magdeburg, Germany), pp. 175–179, Sept. 2004.

[34] J. Hämmerle-Uhl, M. Liedlgruber, A. Uhl, and H. Wernisch, “Multiple re-watermarking using varying wavelet packets,” in *Proceedings of the 2008 IEEE Conference on Multimedia & Expo, ICME ’08*, (Hannover, Germany), pp. 213–216, June 2008.

[35] ISO/IEC 15444-2, “Information technology – JPEG2000 image coding system, Part 2: Extensions,” May 2004.

[36] 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.

[37] 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*, (New York, NY, USA), pp. 63–70, Aug. 2005.

[38] 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*, (Geneva, Switzerland), pp. 202–207, Sept. 2006.

[39] G. Laimer and A. Uhl, “Key dependent JPEG2000-based robust hashing for secure image authentication,” *EURASIP Journal on Information Security*, vol. Article ID 895174, pp. doi:10.1155/2008/895174, 19 pages, 2008.

[40] G. Laimer and A. Uhl, “Improving security of JPEG2000-based robust hashing using key-dependent wavelet packet subband structures,” in *Proceedings of the 7th WSEAS International Conference on Wavelet Analysis & Multirate Systems (WAMUS’07)* (P. Dondon, V. Mladenov, S. Impedovo, and S. Cepisca, eds.), (Arcachon, France), pp. 127–132, Oct. 2007.

[41] D. Engel and A. Uhl, “Lightweight JPEG2000 encryption with anisotropic wavelet packets,” in *Proceedings of International Conference on Multimedia & Expo, ICME ’06*, (Toronto, Canada), pp. 2177–2180, IEEE, July 2006.

[42] D. Engel and A. Uhl, “An evaluation of lightweight JPEG2000 encryption with anisotropic wavelet packets,” in *Security, Steganography, and Watermarking of Multimedia Contents IX* (E. J. Delp and P. W. Wong, eds.), Proceedings of SPIE, (San Jose, CA, USA), pp. 65051S1–65051S10, SPIE, Jan. 2007.

[43] M. Wickerhauser, *Adapted wavelet analysis from theory to software*. Wellesley, Mass.: A.K. Peters, 1994.

[44] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1993.

[45] R. Kutil and D. Engel, “Methods for the anisotropic wavelet packet transform,” *Applied and Computational Harmonic Analysis*, vol. 25, no. 3, pp. 295–314, 2008.

[46] D. Xu and M. N. Do, “Anisotropic 2-D wavelet packets and rectangular tiling: theory and algorithms,” in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing X* (M. A. Unser, A. Aldroubi, and A. F. Laine, eds.), vol. 5207 of *SPIE Proceedings*, (San Diego, CA, USA), pp. 619–630, SPIE, Aug. 2003.

[47] T. Stütz, B. Mühlbacher, and A. Uhl, “Best wavelet packet bases in a JPEG2000 rate-distortion sense: The impact of header data,” in *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME ’10*, (Singapore), pp. 19–24, IEEE, July 2010.

[48] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, pp. 656–715, Oct. 1949.

[49] O. Goldreich, *The Foundations of Cryptography*. Cambridge University Press, 2001.

[50] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, “Format-preserving encryption,” in *Proceedings of Selected Areas in Cryptography, SAC ’09*, vol. 5867, (Calgary, Canada), pp. 295–312, Springer-Verlag, Aug. 2009.

[51] T. Stütz and A. Uhl, “Efficient format-compliant encryption of regular languages: Block-based cycle-walking,” in *Proceedings of the 11th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security, CMS ’10* (B. D. Decker and I. Schaumller-Bichl, eds.), vol. 6109 of *IFIP Advances in Information and Communication Technology*, (Linz, Austria), pp. 81 – 92, Springer, May 2010.

[52] A. Said, “Measuring the strength of partial encryption schemes,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP’05)*, vol. 2, Sept. 2005.

[53] Y. Mao and M. Wu, “A joint signal processing and cryptographic approach to multimedia encryption,” *IEEE Transactions on Image Processing*, vol. 15, pp. 2061–2075, July 2006.

[54] 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*, vol. 2007, no. Article ID 94565, p. 20 pages, 2007.

[55] J. Apostolopoulos, S. Wee, F. Dufaux, T. Ebrahimi, Q. Sun, and Z. Zhang, “The emerging JPEG2000 security (JPSEC) standard,” in *Proceedings of International Symposium on Circuits and Systems, ISCAS’06*, IEEE, May 2006.

[56] 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*, (Dubai, UAE), pp. 1067–1070, IEEE, Nov. 2007.

[57] D. Engel, T. Stütz, and A. Uhl, “Efficient transparent JPEG2000 encryption,” in *Multimedia Forensics and Security* (C.-T. Li, ed.), pp. 336–359, Hershey, PA, USA: IGI Global, 2008.

[58] A. Uhl and C. Obermair, “Transparent encryption of JPEG2000 bitstreams,” in *Proceedings EC-SIP-M 2005 (5th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services)* (P. Podhradsky *et al.*, eds.), (Smolenice, Slovak Republic), pp. 322–327, 2005.

[59] A. Pommer and A. Uhl, “Selective encryption of wavelet packet subband structures for obscured transmission of visual data,” in *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium (SPS 2002)*, (Leuven, Belgium), pp. 25–28, IEEE Benelux Signal Processing Chapter, Mar. 2002.

[60] P. Meerwald, R. Norcen, and A. Uhl, “Cache issues with JPEG2000 wavelet lifting,” in *Visual Communications and Image Processing 2002 (VCIP’02)* (C.-C. J. Kuo, ed.), vol. 4671 of *SPIE Proceedings*, (San Jose, CA, USA), pp. 626–634, SPIE, Jan. 2002.