On Efficient Transparent JPEG2000 Encryption[*]

Thomas Stütz and Andreas Uhl
Dept. of Computer Sciences, University of Salzburg
Salzburg, Austria
tstuetz@cosy.sbg.ac.at, uhl@cosy.sbg.ac.at

ABSTRACT

Efficient (in the sense of computationally efficient as well as efficient from a distribution technology perspective) format-compliant transparent encryption schemes for JPEG2000 are investigated. While the traditional approach of encrypting enhancement layers suffers from high computational encryption demand and drawbacks in distribution, the proposed window encryption approach can reduce computational cost and allows a controlled adaptation of the required security for many application scenarios.

Categories and Subject Descriptors

I.4.2 [**Image Processing and Computer Vision**]: Compression (Coding)
; E.3 [**Data**]: Data Encryption

General Terms

Security

Keywords

JPEG2000, transparent encryption, format-compliant encryption

1. INTRODUCTION

Encryption schemes for multimedia data need to be specifically designed to protect multimedia content and fulfill the application requirements for a particular multimedia environment [23].

For example, real-time encryption of visual data using classical ciphers requires heavy computation due to the large amounts of data involved, but many multimedia applications require security on a much lower level (e.g., TV news broadcasting [15]). In this context, several selective or partial encryption schemes have been proposed recently which do not strive for maximum security, but trade off security for computational complexity by restricting the encryption to the perceptually most relevant parts of the data.

However, encryption may have an entirely different aim as opposed to pure confidentiality in the context of multimedia applications. Macq and Quisquater [14, 15] introduce the term "transparent encryption" mainly in the context of digital TV broadcasting (also called "perceptual encryption" predominantly in the area of audio encryption): a broadcaster of pay TV does not always intend to prevent unauthorized viewers from receiving and watching their program, but rather intends to promote a contract with nonpaying watchers. Therefore, there are two major requirements that have to be met concurrently:

- Hiding a specific amount of image information (security requirement).
- Showing a specific amount of image information (quality requirement).

While the first requirement is a generalization of the confidentiality encryption approach – the condition of full encryption of all image information is extended to a "specific amount" – , the second requirement, namely to explicitly demand a certain image quality, is completely different from scenarios where confidentiality or privacy are the primary aims.

These requirements can be facilitated by providing a low quality version of the broadcast program for everyone, only legitimate (paying) users get access to the full quality visual data. This is also what is meant by the term "try and buy" scenario. Also in image databases, the availability of a thumbnail is of advantage as an incentive for buying the full-quality version. The same is of course true for online video databases. Therefore, privacy is not the primary concern in such an environment. The simplest approach to achieve this would be to distribute both versions: a low quality version to all potential viewers, and a high quality version only to paying viewers. However, this is mostly not desired due to the excessive demand of storage and bandwidth. Furthermore, the full encryption of the entire high quality version imposes a huge computational effort.

Similarly, for video surveillance it is sometimes desirable to show the video feed in order to discourage theft. However, privacy should be protected. A possible solution is to only show the poor quality as a deterrent. If an incident

[*] This work has been partially supported by the Austrian Science Fund (FWF) under Project No. P19159-N13 and by the European Commission through the IST Programme under con tract IST-2002-507932 ECRYPT.

MM&Sec'07, September 20–21, 2007, Dallas, Texas, USA.
Copyright 2007 ACM 978-1-59593-857-2/07/0009 ...$5.00.

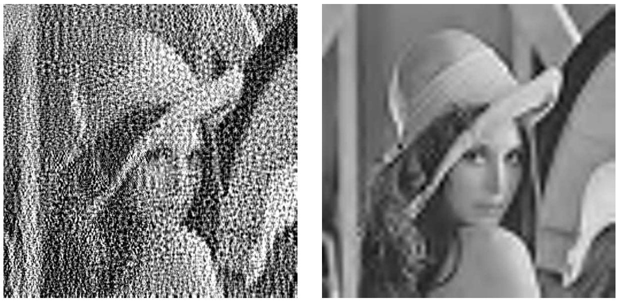
occurs, then the full quality version can be accessed by security personnel. Also, storage of both low quality (for locating the video portion of interest) and high quality versions (for identification purposes) is undesired due to excessive storage demand.

The integration of multimedia encryption into standardized multimedia formats (such as JPEG2000) in a format-compliant way has the great benefit that no additional deployment measures have to be taken. Format-compliant encryption guarantees that the encrypted multimedia file still complies with the format specifications, therefore most operations that can be conducted on the JPEG2000 stream can be transparently conducted on the encrypted stream. Consequently, these approaches can be easily integrated into media distribution techniques. Several format-compliant approaches have already been proposed in literature. Also the JPSEC standard offers and defines a JPEG2000-compliant encryption approach. Concerning the application scenario of a TV broadcaster, the property of format-compliance preservation is beneficial as the entire distribution chain can remain unchanged and the potential customers obtain the promotional low quality versions in exactly the same way as their usual TV streams. By analogy, this also applies to the database and video surveillance scenario.

Scalability of the media format is necessary for the computationally efficient integration of transparent encryption. Commonly, transparent encryption is achieved in this environment by simply encrypting the enhancement layer(s). This has been proposed by [13, 12] using a scalable video codec based on a spatial resolution pyramid, by [3, 4] using an SNR scalable MPEG-2 encoder/decoder, and by [19] for the progressive JPEG variants [7]. Yuan et al. [26] propose to use MPEG-4 FGS for transparent encryption.

For JPEG2000 the concept of transparent encryption is often introduced as an application scenario for conditional access and access control. E.g., in [9, 5, 6, 25, 10, 2, 8, 2] it is proposed to employ conditional access (access control) to protect either the higher resolutions or the higher quality layers of a JPEG2000 image. Commonly it is assumed that the unencrypted parts can be employed to reconstruct a low quality version of the encrypted content. Thereby it is assumed that customers (more specifically, their decoder) know which parts of the encrypted file contain the unencrypted parts. This assumption is, however, not consistent with the goal that no additional deployment measures should have to be taken. In fact most decoders will only be able to decode the format-compliant stream, but do not offer any advanced codestream consumption configurability. Hence the direct reconstruction of the image from the format-compliant stream has a severely reduced quality compared to the embedded public version (see Figure 1).

Anyhow, for more sophisticated ("informed") decoders the codestream consumption policy needs to be communicated (JPSEC or MPEG21 may be suitable for this purpose). For our investigations it is only of interest that a standard-compliant decoder does not have the information of how to consume the encrypted stream, while an informed decoder gets this information in a well-defined way. In the case of JPEG2000 we will propose a method of integrating this consumption information in a JPEG2000-compliant way by exploiting JPEG2000 built-in error concealment methods (**concealed encryption**), such that we further distinguish between JPEG2000 decoders capable of error concealment and those incapable of error concealment. The capabilities of the decoder have a significant influence on the quality of the reconstructed public version.



(a) Direct reconstruction (PSNR 11.88 dB, ESS 0.26) (b) Public version (PSNR 25.36 dB, ESS 0.57)

Figure 1: Direct reconstruction compared to the embedded public version

The traditional approach to implement transparent encryption on top of a scalable format is to encrypt the enhancement layers. For JPEG2000 the traditional approach is discussed in [22], but due to incorrect error concealment in the JJ2000 software, the presented empirical results are updated in this paper.

A drawback of the traditional JPEG2000 transparent encryption approach is that a high percentage of the data has to be encrypted (even more than proposed in [22], as the new experimental results show). Hence approaches that reduce the computational effort of encryption are strongly needed. Since the last quality layers do not contribute much to the image quality, it may be more reasonable not to start encrypting at the end of the data (as suggested by [22]), but at a specific point in the bitstream and encrypt a portion of the bitstream according to the required image quality. This has been demonstrated in the context of progressive JPEG in previous work [20].

In this work we propose and analyze specific JPEG2000 transparent encryption schemes denoted as **window encryption approach** since only a fraction of the file – the window –, which may be positioned at an arbitrary position in the bitstream depending on target quality, is encrypted. We compare these techniques with the traditional transparent encryption approach (which basically encrypts the entire enhancement layer information [22]) in terms of computational (encryption) demand, security, and suitability for different decoder capabilities, namely for simple JPEG2000 decoders (no error concealment), for error-concealing JPEG2000 decoders and for informed decoders.

The investigated transparent JPEG2000 schemes all employ format-compliant JPEG2000 encryption techniques and therefore JPEG2000 and JPEG2000 format-compliant encryption are shortly discussed in Section 2. In Section 3 the two different approaches for transparent encryption of JPEG2000 are presented with a focus on the deployment. Security aspects are discussed in Section 4. The computational complexity of the presented approaches is analyzed in Section 5. In Section 6 experimental results covering both transparent encryption approaches for different decoder capabilities are presented.

2. FORMAT-COMPLIANT ENCRYPTION OF JPEG2000

JPEG2000 employs a wavelet transform. The coefficients are quantized and encoded using the EBCOT scheme, which renders distortion scalability possible. Thereby the coefficients are grouped into codeblocks and these are encoded bitplane by bitplane, each with three coding passes (significance propagation pass, magnitude refinement pass and clean-up pass) except the first non-zero bitplane which only employs the clean-up pass. The passes are entropy-coded with the multiplication-free arithmetic MQ-coder. Each of these coding passes may contribute to a certain quality layer.

A packet body contains CCPs (codeblock contribution to packet) of codeblocks of a certain resolution, quality layer and precinct (a spatial inter-subband partitioning structure that contains one to several codeblocks) [21]. In the packet header the corresponding meta data is stored, e.g., the length of the CCPs and the leading zero bitplanes of a codeblock.

Format-compliant encryption schemes for JPEG2000 target the packet bodies, consisting of arithmetically coded codeblock data (CCPs). The CCPs and the packet body must not contain any two byte sequence in excess of `0xff8f` (delimiting markers) nor end with `0xff`. However, the last requirement solely avoids delimiting markers at CCP borders and is therefore of minor importance for codestream compliance. If an encryption method applied to the packet bodies complies with these requirements, the resulting encrypted JPEG2000 file is format-compliant and thus can be consumed by every standard-compliant decoder. Hence encryption schemes that avoid the generation of delimiting markers have to be employed [24, 11, 5].

An encryption method that avoids the generation of marker codes is the following: AES in Counter mode is employed to generate a random key stream. The `0xff` bytes are deleted from the key stream. The key stream bytes are then added modulo `0xff` to packet body bytes not preceded by or equal to a `0xff` byte. Obviously this method does not generate new `0xff` bytes (due to the addition modulo `0xff`), and all two byte sequences starting with a `0xff` byte are preserved (these are not in excess of `0xff8f` due to the codestream syntax). This method is similar to the one presented in [24].

It can be applied to any fraction of packet body data (the packet body byte before the first encrypted byte has to be taken into account, i.e. has to be checked to be a `0xff` byte). In general the affected CCP (one to more coding passes) will be irrecoverable for the decoder.

3. TRANSPARENT ENCRYPTION

In this work two basic transparent encryption approaches for JPEG2000 are considered. However, their suitability for transparent encryption purposes is also significantly influenced by the capabilities of the decoder, which has a major influence on the quality of the derived low quality version.

3.1 Traditional Approach

The traditional approach to implement transparent encryption on-top of a scalable bitstream is to encrypt all the enhancement layers. In the case of JPEG2000 this approach is straightforward: in the compressed JPEG2000 file the position at which the desired low quality is achieved is determined and all the successive packet body data in the file (enhancement layers) are encrypted. The appropriate position from which to start encryption can be determined by supplying a decoder adaptively with compressed data until the desired low quality is achieved. Starting from this position, the packet body data is format-compliantly encrypted, e.g., as sketched in section 2.

In [22] it is shown that most of the JPEG2000 file has to be encrypted in order to obtain a suitably low quality version. In section 6 we will show that the gap in image quality between a direct reconstruction (customer) and a possible attack is too large and thus the direct reconstruction of a sufficiently secured image is not suitable as it is simply too noisy (if no precautions are taken, e.g., concealed encryption as discussed in section 3.4).

3.2 Window Encryption Approach

The window encryption approach is an umbrella term for all schemes that format-compliantly encrypt only a fraction of the packet body data (encryption window) at a certain position in the file. The main advantage is the reduced encryption effort, the disadvantage is a possibly decreased security (cf. to section 4). In this paper we focus on the influence on the quality of the reconstruction of the low quality version.

In section 6 optimal settings for the position and the encryption amount are evaluated for the different decoder capabilities.

3.3 Decoder Capabilities

For the application scenarios of transparent encryption it is essential that the public low quality version is not only accessible, but that it is accessible in a convenient way, such that neither additional software nor special hardware are necessary. Therefore we investigate the influence of different decoders on the application of transparent encryption. The most conservative assumption is a decoder which can barely decode format-compliant data (this can be considered the real-world case, not just considering JPEG2000). A more optimistic assumption is a decoder which already implements format-specific error concealment. In the case of JPEG2000 this is currently an over-optimistic assumption (cf. section 3.4); however, in the case of more wide-spread adoption of the standard, the situation is likely to change. The error concealment capability can be exploited to minimize the gap between the reconstruction of the decoder (of the customer) and a possible attack (cf. section 3.4). The most optimistic assumption is a decoder that can use side-channel information to perfectly extract the low quality version. Apart from proprietary solutions, standardized tools like JPSEC and MPEG21 may be employed.

Simple Decoder
A simple decoder can only decode the JPEG2000 file without taking advantage of error concealment information.

Concealing Decoder
A concealing decoder is capable of applying error concealment if it detects errors in the JPEG2000 codestream.

Informed Decoder
An informed decoder is capable of extracting the public low quality version with the best possible quality.

The quality of the embedded low quality version is in general not the quality a possible customer can retrieve from a transparently encrypted JPEG2000 file, since the encrypted parts introduce severe noise into the decoded image.

3.4 Concealed Encryption and Fully Concealed Encryption

In [17] and [22] JPEG2000 error concealment (segmentation symbol) is employed to mimic attacks on partial JPEG2000 encryption. It is a clever way to mimic a sophisticated attack, but it can also be employed to minimize the gap between an attack and the reconstruction available to the customer. Thereby the segmentation symbol `0xa` (more precisely, the four bits 1001) is encoded in uniform context at the end of a codeblock's bitplane's last coding pass. The encrypted parts in general will not produce a `0xa` at the end of bitplane.

We propose to employ the JPEG2000 error concealment such that a concealing decoder is capable of identifying the encrypted parts. A drawback of this solution is that the detection of an encrypted fraction only works with a probability of 15/16, because encryption randomly generates a 1001 sequence at the end of a coding pass in 1 out of 16 cases. Therefore some noisy encrypted parts cannot be detected. In order to improve the detection rate, predictive termination of each coding pass can be additionally employed. Thereby error concealment information is deducible for every coding pass. According to the JJ2000 documentation, 3.5 bits of error concealment information are left on the spare least significant bits of each coding pass.

Both methods can be used together to increase the detection rate, but some encrypted parts may still not be detected and introduce noise. The results in Section 6 provide empirical background to assess the different error concealment strategies (for transparent encryption).

But a coder (the TV broadcaster/content provider/...) can do better: he can encrypt the bitplane data such that correct error concealment information (namely the segmentation symbol and/or the appropriate decoder state for predictive termination) is no longer generated (**fully concealed encryption**), thereby enabling the decoder to detect every encrypted fraction. The algorithm for fully concealed encryption is:

```
key = getRandomKey();

encryptedJ2KFile = encrypt( J2KFile, key);

while !areAllErrorsDetectable(encryptedJ2KFile)

    key = getRandomKey();
    encryptedJ2KFile = encrypt( J2KFile, key);
```

Note that only the first coding pass to be encrypted or the corresponding bitplane of a codeblock have to be correctly identified in the function areAllErrorsDetectable. Whether the first coding pass to be encrypted or the corresponding bitplane has to be detected depends on the actual error concealment strategy of the decoder, more precisely, if the coefficient values are reset to the state prior to error detection on a coding pass basis or on a bitplane basis. In section 6 we apply a decoder which conceals on a coding pass basis in order to show the highest image quality contained in the encrypted JPEG2000 codestream.

The average computational complexity of this algorithm for fully concealed encryption is analyzed in Section 5. However, this algorithm's computational complexity has no absolute limit (for individual cases) and therefore we additionally discuss an algorithm that offers a clearly defined computational complexity for all cases but may deliver a suboptimal solution.

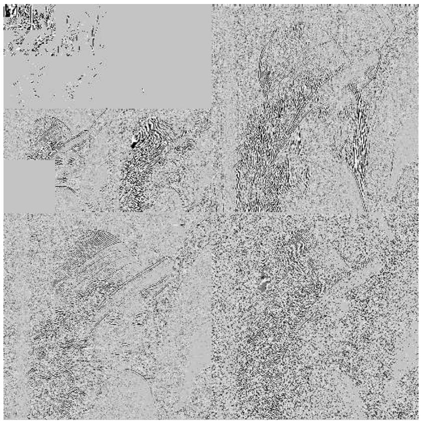


Figure 2: Affected wavelet coefficients for the window encryption approach with layer progression (1% encrypted starting at 2%)

In this context it is also notable that the reference JPEG2000 software JJ2000, which is the only one to offer this feature (concerning jasper and JJ2000), has a minor bug in the actual error concealment function (In the jj2000.j2k.entropy.decoder.StdEntropyDecoder in line 2475 (4.1 unix release) it should be "resetmask = (-1)<<(bp+1);" instead of "resetmask = (-1)<<(bp);"), which renders the error concealment mostly useless as the erroneous bitplane is written and then the decoding of the codeblock is stopped.

If the segmentation symbol and predictive termination are employed together another little bug has to be fixed: In the cleanup pass code of the JJ2000 decoder (in the jj2000.j2k.entropy.decoder.StdEntropyDecoder in line 2439 (4.1 unix release)) it should be "error = error || mq.checkPredTerm();", because otherwise a correct termination overrides a previously detected error in the segmentation symbol decoding.

The actual process of concealing detected errors is not standardized, only the detection mechanism is. A straightforward approach is to set the coefficient value to the prior state (before the error) and set the next bit (of the next bitplane to decode) to one, which minimizes the error in average for a uniform distribution of the remaining (encrypted) coefficient bits. If a decoder resets the coefficient values on a coding pass basis, the computational complexity of decoding (more copies) is higher, but the reconstructed image is of higher quality. In our experiments in Section 6 we reset the coefficient values on a coding pass basis.

4. SECURITY

The evaluation of the security of partial / selective encryption schemes for visual data – both the traditional and the window encryption transparent encryption approach fall into that category – differs from the classical analysis of ciphers as pointed out in [18]. Instead of the full recovery of the plaintext, the reduction of distortion (it is assumed in [18] that a low quality version is publicly available through a side-channel) is the main objective.

The assumption of a publicly available low quality ver-

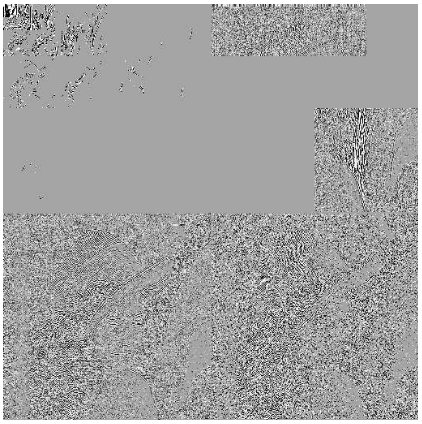


Figure 3: Affected wavelet coefficients for the window encryption approach with layer progression (5% encrypted starting at 3%)

sion – which is a central part of the model for cryptanalysis for selective encryption of [18], but is very strong and may not be suitable for all application scenarios of selective encryption – holds in the case of transparent encryption. A low quality version is always accessible and the main security issue is to prevent an attacker from deriving a better image quality than the intended publicly available version. Thereby all information contained in the unencrypted parts may be used to increase the image quality.

For transparent encryption of JPEG2000 this means that it is not the actual plaintext coefficient value a possible attacker tries to find, but any appropriate value that reduces the distortion of the encrypted value. Therefore she has to determine the coefficients which have been encrypted. Furthermore she might even succeed in identifying the fraction of the coefficient bitplane data that has been encrypted (coding passes of a certain bitplane). In our security analysis the attacker is assumed to be capable of identifying all encrypted parts even on a coding pass basis. In fact statistical analysis of the coefficient bitplane data may enable an attacker to detect the encrypted parts at least on a bitplane basis. Thus the usage of error concealment to facilitate the identification of the encrypted parts does not impose a weakening of the security of our scheme against a serious attacker. See Figures 2 and 3 for visual examples of the actually affected coefficient data if the window encryption approach is applied with different settings for the Lena image. In the case of the traditional approach all higher frequency subbands are encrypted as well. Having identified the encrypted parts of the coefficients, an attacker may replace the missing coefficient information, e.g., by exploiting inter subband dependencies or general statistical properties of the missing wavelet coefficients. However, these kind of attacks can be prevented by applying window encryption with a higher encryption percentage. Figure 3 reveals that the encryption of only 5% affects even one of the highest frequency subbands. Simple prediction schemes (e.g., linear interpolation) for the missing data did not improve the image quality.

The strengthening of the security of the window encryption scheme by increasing the encryption percentage is opposed to the demand of a computationally simple scheme. In Section 5 the issue is discussed in greater detail.

The reduction of key space for fully concealed encryption has to be taken care of by slightly increasing the length of the encryption key (a few bits will be sufficient for most cases).

4.1 Traditional Approach

All the packet body data (most of the file, over 90% in general) after the low quality version is encrypted. Thus the information available to an attacker is the low quality part and the packet header data of all packets. If the encrypted parts are not explicitly specified, an attacker has to find unencrypted parts. Therefore she can truncate the codestream in the decoder (JJ2000 options: -parsing off -nbytes) and apply some criteria to detect a decrease in image quality (e.g., smoothness). This attack will be denoted **truncation attack**. However, after a successful truncation attack the enhancement of image quality with the remaining plaintext information is not promising. Apart from the leading zero bitplanes of a codeblock the packet headers do not contain directly image content related information. However, it is common to use the biggest codeblock size available (64x64) in order to boost compression performance and thus this information is in general negligible.

4.2 Window Encryption Approach

The situation is quite different for the window encryption approach, where only a small fraction of the file is encrypted. This fraction of the file represents compressed bitplane information of wavelet coefficients. The affected coefficients can be identified and estimated via inter subband redundancy. This estimation will be different for each subband (e.g., it differs for the LL and HH subbands). However, experiments have shown that simple estimates are likely to introduce artefacts in the reverse wavelet transform and it is generally a good choice to set the affected coefficients to their prior value (the value before the encrypted fraction, i.e., an encrypted coding pass in a CCP).

Another possible security issue is that when a coding pass is encrypted it is commonly assumed that the successive coding passes are destroyed. In [8] a thorough discussion of arithmetic coding and security is conducted, in which the poor resynchronization properties of arithmetic coding are pointed out, which backs up this assumption.

Nevertheless the window encryption approach is potentially more susceptible to further security issues compared to the traditional approach. The actual security does, however, greatly depend on the encryption amount and position.

Since an attacker is likely to identify the encrypted parts anyway, the explicit signaling either through error concealment information (concealed encryption) or other means (informed decoder) is no security threat. In our investigations we consider the fully concealed encryption the best quality that can be derived from the encrypted JPEG2000 file (with the window encryption approach).

In [16] a security measure for visual data is introduced which separates evaluation of luminance and edge information into a *luminance similarity score* (LSS) and an *edge similarity score* (ESS). ESS ranges, with increasing similarity, between 0 and 1. We use the weights and blocksizes proposed by [16] in combination with Sobel edge detection. The distortion introduced by transparent JPEG2000 encryption is measured rather appropriately with the PSNR and hence ESS is only additionally given for the visual examples.

5. COMPUTATIONAL COMPLEXITY

If fully concealed encryption is not applied, the computational complexity of the encryption schemes compared to the compression pipeline is very low. In [23] an in-depth analysis can be found for JPEG2000 and AES and it is concluded that in an "online-scenario" (compression is part of the application scenario) the runtime benefits of selective encryption are marginal (below 1%). For "offline-scenarios", e.g., video on demand, where compression is done offline, performance benefits can be achieved. In general, format-compliant encryption of JPEG2000 is computationally more demanding than AES encryption, as the structure of the JPEG2000 codestream needs to be preserved. The employment of SOP and EPH markers reduces the cost of the identification of the packet body data in the JPEG2000 codestream and therefore the overhead of format-compliant encryption with this approach is small. Thus if compression takes place, the window encryption approach will not significantly improve the performance of the overall compression/encryption system compared to the traditional approach, as encryption makes up only a small fraction of the overall complexity.

However, if fully concealed encryption is applied, window encryption can lower the computational complexity for both offline- and online-scenarios. In order to assess whether the encrypted coding passes or bitplanes are identified, major parts of the decompression pipeline have to be executed. In a simple implementation the entire decompression pipeline has to be conducted, but at least the arithmetic decoding of the encrypted parts always has to be executed in order to test whether the error concealment information has been accidentally generated. Hence every evaluation of a certain key is computationally expensive. Thus it necessary to determine the average number of iterations (evaluations of a certain key) that is needed until a correct key is found. The average number of iterations depends on the probability of all relevant errors being detected in an encrypted JPEG2000 file. If concealment is conducted on a coding pass basis, it has to be taken into account which coding pass (significance propagation, magnitude refinement or cleanup) of a codeblock is encrypted first. According to JJ2000 documentation, predictive termination embeds in average 3.5 bit of error concealment information for every coding pass. Additionally, the segmentation symbol independently offers 4 bit of error concealment information. The application of both strategies protects the data of the first bitplane (only cleanup pass) with approximately 7.5 bit. In our experiments, for which we employed a test set of 50 images each coded with 40 codeblocks, a slightly smaller value of 7.09 bit has been evaluated (or in other words, every 137 encrypted first bitplane's cleanup pass has not been detected). Every bitplane apart from the first is protected with approximately 14.5 bit (3 x 3.5 bit + 4 bit). Every cleanup pass contains on average 7.5 bit of error concealment information, while magnitude refinement and significance propagation pass contain on average 3.5 bit of error concealment information. For the further analysis we assume that the first encrypted fraction of a codeblock is detected in one of 2^7 cases, which is certainly met for concealment on a bitplane basis. Furthermore the first bitplane's cleanup pass is the most important one, as most of the codeblock data is encrypted from the start and thus the successive coding pass data is useless. Not detecting the first bitplane's cleanup pass introduces the most noise, hence the assumption that the first encrypted fraction of a codeblock is not detected in one out of 2^7 cases, though not totally precise, is appropriate for the further analysis. Depending on n, the number of encrypted codeblocks of a JPEG2000 image, the probability that all fractions which have been encrypted first are detected, can be estimated with p^n, where an appropriate value for $p = 1 - 2^{-7}$. E.g., 69 codeblocks are encrypted (as for a 512x512 image with wavelet decomposition level 5 and the traditional approach, i.e., all codeblocks but the codeblock of the lowest frequency subband are affected) the probability that all first fractions to be encrypted are detected is about 0.582. If high resolution and multi component images are encrypted, this probability decreases significantly. Window encryption reduces the computational complexity as n, the number of encrypted codeblocks, is kept minimal. E.g., only 13 codeblocks have to be taken into account with the window encryption approach with 1% encrypted at 2% in the codestream (cf. to Figure 2), and the corresponding probability that all encrypted data is detected is 0.903. For high resolution and multi component images the resulting reduction of complexity due to a reduced number of necessary iterations is even more significant. For a 2048x1024 image, the probability for 2^9 codeblocks is 0.018, while window encryption (approximately $2^9/6$ codeblocks) increases the probability to 0.512. A drawback of the presented algorithm for fully concealed encryption is that the number of iterations is random and only its probability (and thus the expected average number of iterations) can be given. This is, however, unsatisfactory and insufficient for many application scenarios that require a constant processing time. For these application scenarios an algorithm with a constant time demand is needed. The following algorithm trades off the optimal solution (full identification of the encrypted parts) for a constant runtime.

For k different encryption keys the encryption and the decompression is conducted and the key with the highest PSNR rating of the corresponding encrypted image is selected. For a sufficiently large number k this is equal to the fully concealed encryption. The probability of finding the correct solution for k different keys can be given explicitly by $1 - (1 - p^n)^k$, where n is the number of encrypted codeblocks. Again we see that the window encryption approach reduces computational complexity as it reduces the number of different encryption keys to test for a given desired probability of detecting all errors. If the correct solution is not found, the algorithm is likely to find a solution that is very close to the optimal solution in terms of image quality (PSNR). For 69 codeblocks the test of 5 different keys will give the correct solution with a probability of 0.987, while the corresponding window encryption (13 codeblocks) has a probability of 0.999991 to find the correct solution. For 2^9 codeblocks the correct solution is found with a probability of 0.087, while the corresponding window encryption ($2^9/6$ codeblocks) has a probability of 0.972.

To sum up, window encryption can significantly improve the runtime performance of fully concealed encryption. For the application scenario of transparent encryption the second algorithm is better suited as a reasonably high image quality of the encrypted image is very likely and the computational complexity is exactly defined and can even be adjusted to the desired level. Trying a few different keys (depending on the number of codeblocks to encrypt) will in general avoid most of the low quality exceptions that concealed encryption is likely to produce.

6. RESULTS

In this section we will present results covering both the traditional and the window encryption approach. The actual application of both approaches requires selecting appropriate parameters; the traditional approach requires selecting a position in the file from that on the packet body data is encrypted, while the window encryption approach needs the specification of two parameters the encryption window size and the position from which to start encryption. The experimental results shall help to identify the appropriate parameter settings for the specific application scenario.

For the specific application scenario, the decoder capabilities and the applied error concealment strategy have an essential influence. An in-depth analysis for the different decoder capabilities as outlined in section 3.3 is conducted and it is evaluated to which degree and with which specific parameters the two basic approaches are applicable. Visual examples for suggested parameter values are given and the security aspect for those parameter settings are discussed.

Since both approaches require the specification of the actual start of encryption (which is given in percent with respect to the absolute JPEG2000 compressed file size), the compression parameter with predominant influence on the actual choice of this parameter is the progression order. Thus we present results of the two major progression types, namely resolution and layer progression.

The results were obtained using the software JJ2000 and custom software to encrypt the JPEG2000 files and to perform fully concealed encryption. Apart from evaluations on numerous single images, such as the Lena, Cameraman, Goldhill, Barbara, frames from the Akiyo sequence, frames from the Mobile sequence, ... , an averaged evaluation was conducted on a set of 50 representative images (a compilation of sub-sampled versions (512x288) from the publicly available VQEG test sequences[1]). The progression order has been subject of investigation (set either to layer or resolution progression), as well as different error concealment options (either the segmentation symbol, labeled "seg" in the plots, predictive termination, labeled "pterm" or both strategies have been employed). In some plots the results for the segmentation symbol have not been plotted for the sake of clarity, but the results are always better than the predictive termination and worse than both strategies together. In Figures 6, 7, 13, and 17 the results for both error concealment strategies (predictive termination and concealment symbol) and the buggy concealment are plotted. The employment of the segmentation symbol is better suited for the concealment of transparent encryption. The reason is that most of the encrypted fractions start at the first coding pass of codeblock, which is protected by 4 bits in case of the employment of the segmentation symbol and in average with 3.5 bits in case of predictive termination. Evidently, the first coding pass is crucial (and its protection with 3.5 bits in average is clearly worse than its protection with constantly 4 bits). Both methods together offer the most reliable error detection.

Apart from the insertion of start and end of packet header markers and coding of the segmentation symbol, the other compression parameters were set to JJ2000 default values, which do not include a target bitrate (almost lossless compression) and 32 quality layers.

6.1 Traditional Approach

The best possible reconstruction and therefore attack of a traditionally transparently encrypted image is the truncation attack, which is identical to the fully concealed case. In order to embed the appropriate low quality version, the start of encryption has to be chosen according to the image quality obtained by the truncation attack. In Figure 4 and Figure 5 the quality of the reconstructions is plotted against the start of encryption (in percent of the file size) for the JPEG2000 compressed Lena image with layer and resolution progression. While actual numbers slightly vary from source image to source image, the characteristics of the plots and the differences between the reconstructions of the different decoders are preserved. Specifically for the Lena image and as a rule of thumb for the average case, a start of encryption at 3% for layer progression is suitable for a moderately low quality public version (see Figure 9(b)) and a slightly higher start at 4% is appropriate for resolution progression (see Figure 1(b)).

However, the image quality a possible customer can retrieve depends on the capabilities of her decoder.

6.1.1 *Simple Decoder*

There is an enormous gap in image quality between the direct reconstruction of a simple decoder and the truncation attack. Visual examples are shown for the Lena image and resolution progression (start of encryption is at 4%) in Figure 5 and for layer progression in Figure 4. For a variable start of encryption, the enormous gap between the image qualities of the truncation attack and the direct reconstruction is illustrated in Figure 4 (layer progression) and Figure 5 (resolution progression). The plots for the averaged cases, as shown in Figure 6 (layer progression) and Figure 7 (resolution progression) prove that the gap is wide in general. Hence the traditional approach cannot meet the quality requirement of transparent encryption for a simple decoder.

6.1.2 *Concealing Decoder*

A concealing decoder can take advantage of the embedded error concealment information in order to deliver a better reconstruction of the embedded low quality version. The encrypted JPEG2000 file is still accessible by simple decoders, but these do not profit from the embedded error concealment information. For a concealing decoder we have evaluated the case of fully concealed encryption (it is taken care of by the content provider that all encrypted parts can be detected) and the concealed encryption case for three different error concealment strategies (seg, pterm and both). Thereby only error concealment information is embedded and the random erroneous generation of segmentation symbols is accepted. Figures 4, 5, 6 and 7 illustrate the experimental results for the Lena image and the test set. Concealed encryption with the segmentation symbol or the predictive termination alone is rather likely to produce distorted images (the error concealment information, e.g., the segmentation symbol, is generated by accident), while the usage of both strategies improves the average image quality considerably. Nevertheless extremely low image qualities are still possible, which might severely degrade the overall quality of video (JPEG2000 is employed as intra frame codec by the Digital Cinema Initiative [1]). The fully concealed encryption is equal to the truncation attack (the coefficients are reset on a coding pass basis) and reliably delivers the desired image quality.

[1] ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/

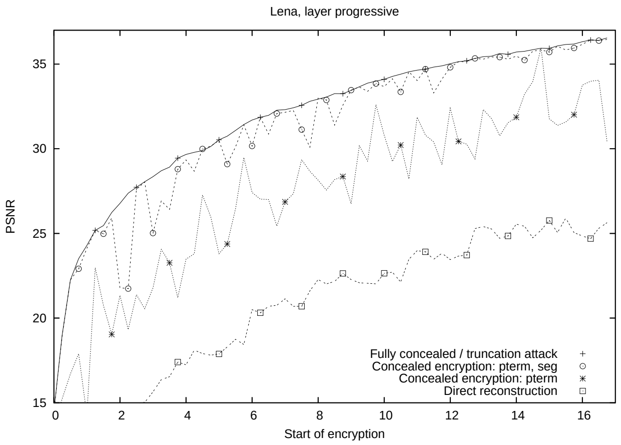


Figure 4: The traditional approach for the Lena image and layer progression

Fully concealed encryption can guarantee a certain image quality at the customer side (especially beneficial for video sequences), while the encrypted JPEG2000 file is still accessible for simple decoders.

6.1.3 Informed Decoder

Basically the best image quality contained in the encrypted JPEG2000 file can be accessed (i.e., the same as the truncation attack and the fully concealed encryption). The drawback of this solution is that in general other container formats have to be employed which may render the encrypted file useless for only JPEG2000-compliant decoders. However, with JPSEC it is possible to combine fully concealed encryption (or just concealed encryption) with a still JPEG2000-format-compliant encrypted codestream; only an additional marker segment (SEC) is present in the JPEG2000 main header.

Visual examples for the traditional approach and resolution progression are given in Figure 8 for a start of encryption at 4% of the codestream.

For layer progression and a start of encryption at 3% of the codestream Figures 9 and 10 give visual examples.

The drawbacks of the traditional approach are the higher computational complexity due to the higher encryption amount and consequently a higher complexity of the fully concealed encryption (see Section 5), as most of the file needs to be encrypted and simple decoders cannot supply a sufficiently good image quality. Therefore the window encryption approach is evaluated in Section 6.2.

6.2 Window Encryption Approach

For the window encryption approach the currently best attack is identical to fully concealed encryption. Thereby all encrypted coefficient data is concealed on a coding pass basis. The window encryption approach can be implemented with various parameters; however, the main goal is to identify parameter settings which reduce the encryption effort and improve the applicability for the widest range of decoders.

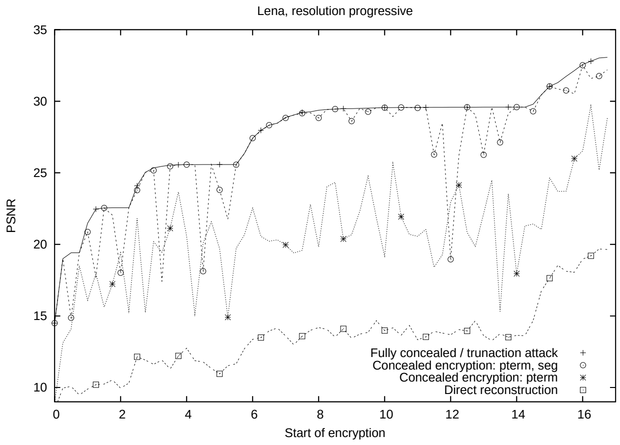


Figure 5: The traditional approach for the Lena image and resolution progression

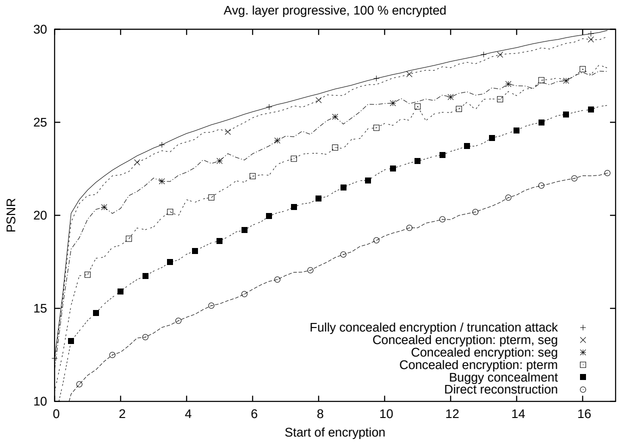


Figure 6: The traditional approach for layer progression

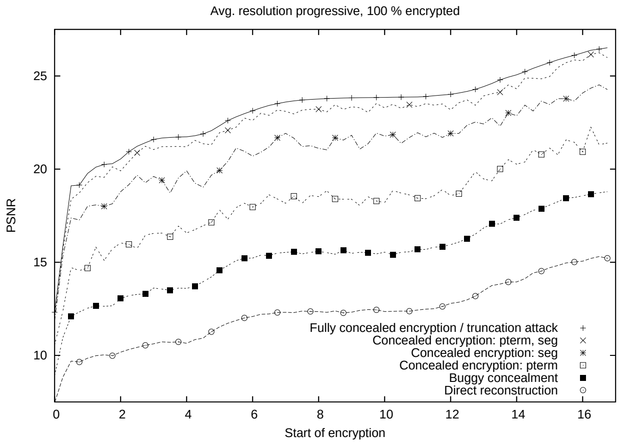
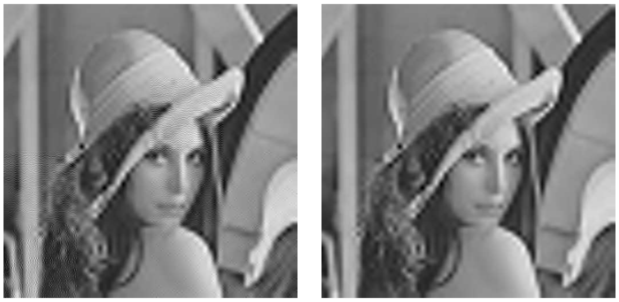


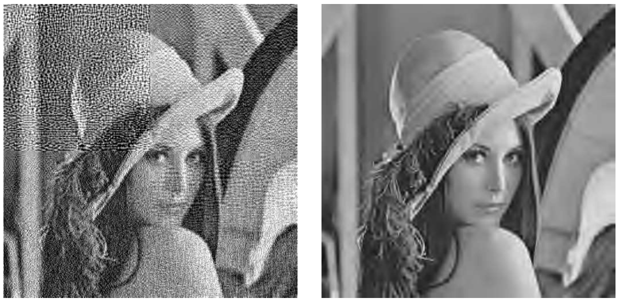
Figure 7: The traditional approach for resolution progression



(a) Concealed encryption (PSNR 23.83 db, ESS 0.59)

(b) Fully concealed encryption (PSNR 25.57 db, ESS 0.60)

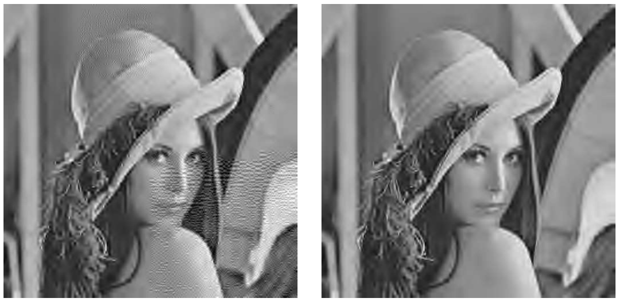
Figure 8: The traditional approach starting at 4% and resolution progression: concealed compared to fully concealed encryption



(a) Direct reconstruction (PSNR 15.75 dB, ESS 0.33)

(b) Fully concealed encryption (PSNR 28.42 dB, ESS 0.69)

Figure 9: The traditional approach starting at 3% and layer progression: direct reconstruction compared to embedded public version



(a) Concealed encryption (PSNR 25.33 dB, ESS 0.62

(b) Fully concealed encryption (PSNR 28.42 dB, ESS 0.69

Figure 10: The traditional approach starting at 3% and layer progression: concealed compared to fully concealed encryption

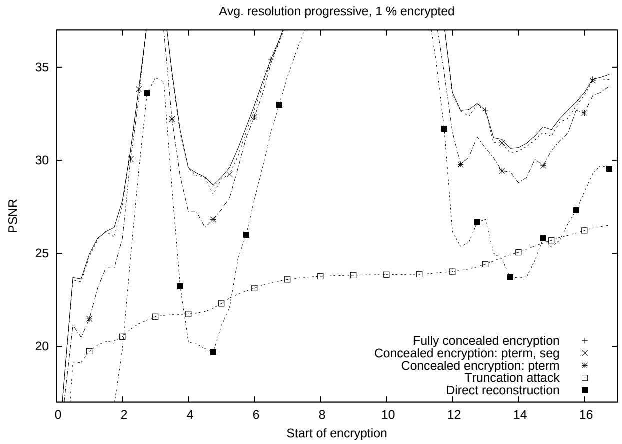


Figure 11: The window encryption approach with 1% encrypted and resolution progression

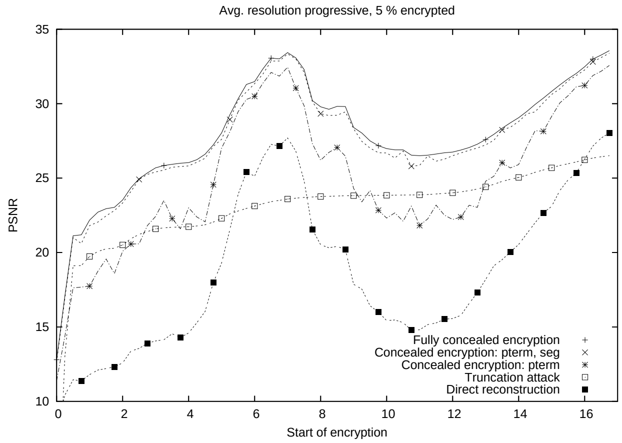


Figure 12: The window encryption approach with 5% encrypted and resolution progression

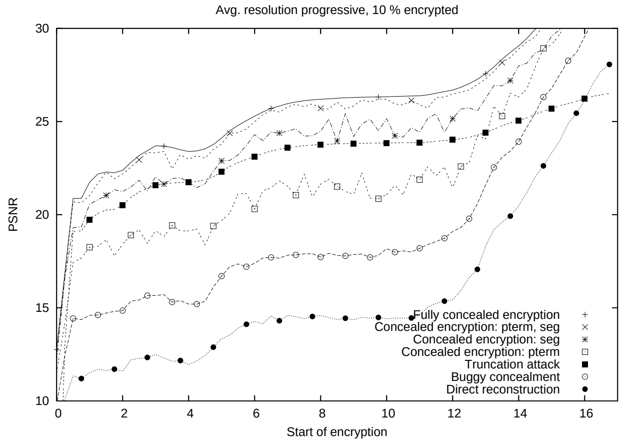


Figure 13: The window encryption approach with 10% encrypted and resolution progression

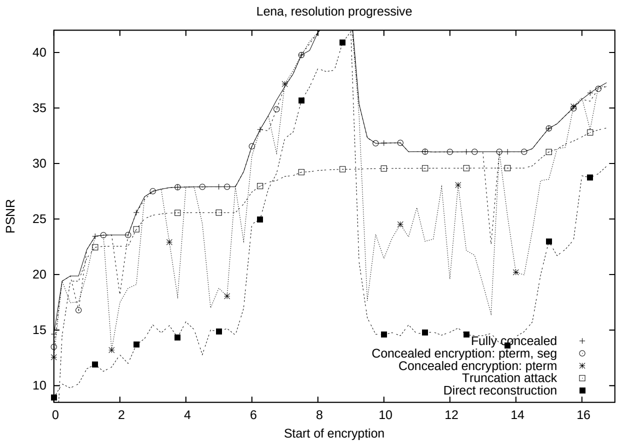


Figure 14: The window encryption approach with 5% encrypted and resolution progression for the Lena image

Thus we present evaluations of the window encryption approach for 1%, 5% and 10% encrypted at varying starts of encryption (again given in percent of the absolute file size) for both layer and resolution progression.

At first we will shortly discuss resolution progression for the window encryption approach, which cannot provide improved functionality (with respect to the distribution and the decoder capabilities) compared to the traditional approach. If only 1% is encrypted, extreme peaks in image quality for all graphs can be found (in the plot based on the image set shown in Figure 11), while there is a significant gap between the direct reconstruction and the actually best quality version, which is obtained by the fully concealed encryption. In a slightly reduced way the same characteristics can be found for 5% encryption and resolution progression (cf. Figure 12). The individual plot for the Lena image (see Figure 14 explains this behavior: the peaks in image quality are achieved whenever the encryption window is located at the last quality layer contributions of a resolution. If even more data is encrypted, the window encryption approach applied to resolution progressive JPEG2000 files behaves more and more like the traditional approach, e.g., Figure 13 which illustrates the results for the window encryption approach with 10% encrypted strongly resembles Figure 7, which illustrates the tradtional approach for resolution progression. Summing up, the window encryption approach for resolution progression can reduce the encryption effort, but does not offer additional functionality.

For the window encryption approach and layer progression, the results are presented for 1% encrypted (see Figure 15), 5% encrypted (see Figure 16) and 10% encrypted (see Figure 17). The best results are obtained for the encryption of only 1%: Firstly the encryption effort is severly reduced and secondly the gap between a direct reconstruction and the best reconstructible image quality (the fully concealed encryption) is reduced. Visual examples of the direct reconstruction and the fully concealed encryption are given in Figure 18. However, reducing the encryption effort this dramatically may impose a security threat (in the sense of [18], i.e., an attacker is capable of deriving a considerably

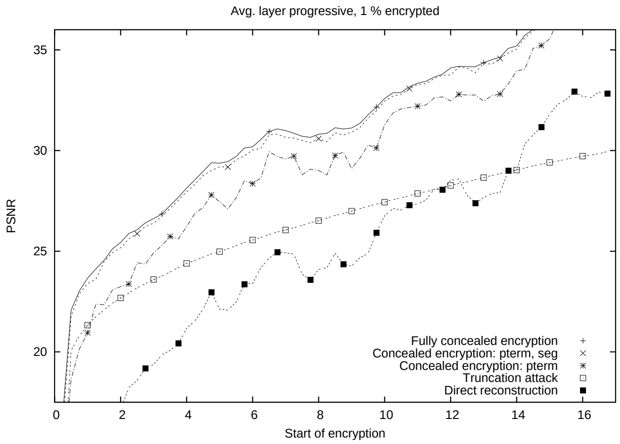


Figure 15: The window encryption approach with 1% encrypted and layer progression

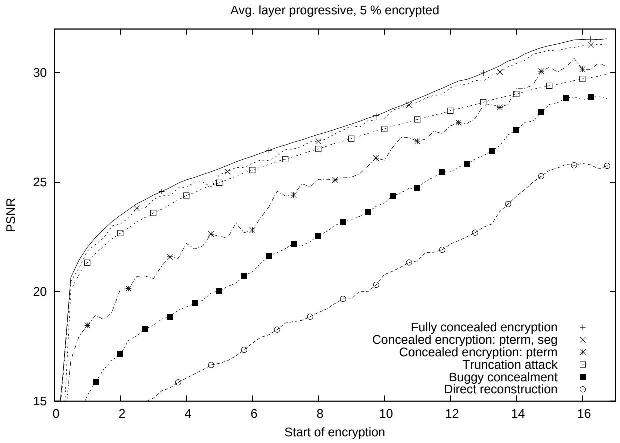


Figure 16: The window encryption approach with 5% encrypted and layer progression

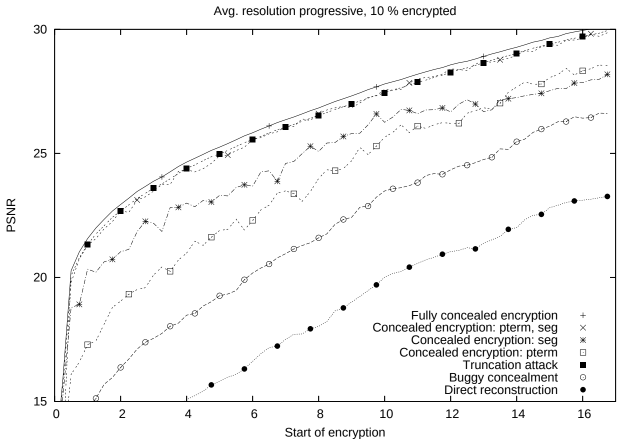
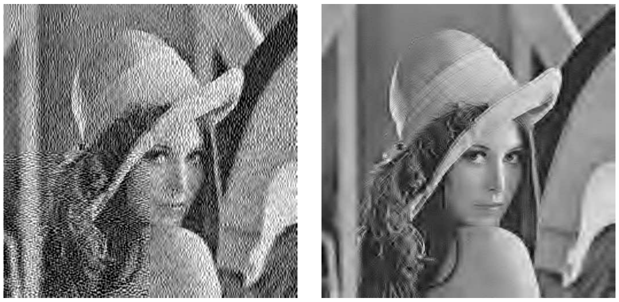


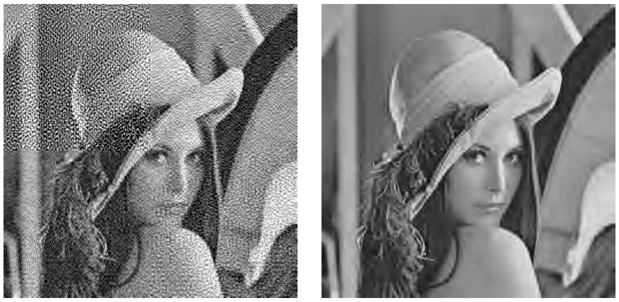
Figure 17: The window encryption approach with 10% encrypted and layer progression



(a) Direct reconstruction (PSNR 17.32 dB, ESS 0.30)

(b) Fully concealed encryption (PSNR 28.24, ESS 0.64)

Figure 18: The window encryption approach with layer progression (1% encrypted starting at 2%)



(a) Direct reconstruction (PSNR 16.70 dB, ESS 0.33)

(b) Fully concealed encryption (PSNR 28.92 dB, ESS 0.69)

Figure 19: The window encryption approach with layer progression (5% encrypted starting at 3%)

improved image quality from the partially encrypted image). Figure 2 shows the reconstructed wavelet coefficients after fully concealing the encrypted parts, i.e. setting the coefficient values to the last untainted state. A considerable number of intermediate subband coefficients is affected (regions with uniform gray color) by encrypting only 1% of the layer progressive JPEG2000 Lena image. Possible attacks either have to estimate the missing coefficient data on the basis of preserved information (basically the shown coefficients) or find an attack to retrieve the image information of the successive coding passes, which depend on the encrypted passes.

Increasing the encryption percentage leads to results that resemble the traditional approach (see Figure 16 and Figure 17). Visual examples are given for the 5% encryption starting at 3% in the file (cf. Figure 19), which are close to the recommendation for the traditional approach. The effect on the wavelet coefficients is shown in Figure 19, compared to 1% encrypted starting at 2% even more coefficient data is affected.

6.2.1 Simple Decoder

Simple decoders can profit if only 1% is encrypted, since the gap between a direct reconstruction and the embedded public version is reduced. However, a weakening of the security has to be accepted. Encrypting a higher percentage, the gap increases and the results get closer to that of the traditional approach.

6.2.2 Concealing Decoder

For concealing decoders the reduction of encrypted data reduces the possibility that noise is introduced by wrongly considering encrypted data as valid. Additionally, the reduction of encrypted data is beneficial for the fully concealed encryption case as less packet body data has to be considered for the computationally complex avoidance of correct error concealment information for the encrypted parts (at least the arithmetic decoding of the encrypted data has to be performed in order to determine whether error concealment information, e.g., a segmentation symbol, is accidentally generated). This issue is discussed in more detail in Section 5.

Again the usage of both error concealment strategies is beneficial and delivers a significantly higher image quality.

6.2.3 Informed Decoder

For the informed decoder the only direct benefit is the reduced encryption effort.

However, with JPSEC the efficient integration of the window encryption approach with fully concealed encryption (or concealed encryption) and only a small portion encrypted is possible and recommended in order to efficiently meet the requirements of transparent encryption for all decoder capabilities. Security is thereby traded-off for applicability, in the sense of reduced encryption complexity and improved distribution capability.

7. CONCLUSION

In this work we have proposed the window encryption approach for efficient transparent encryption with JPEG2000. The application of JPEG2000 error concealment strategies to facilitate the effective deployment of transparent JPEG2000 encryption is proposed and experimentally approved. Our experiments and theoretical analysis approve that the usage of both error concealment strategies is beneficial for performance as well as applicability. Extensive experiments that cover both the traditional and the window encryption approach have been presented and discussed. On the basis of our evaluations, we have found that for applications where security is not the main objective, the window encryption approach with only a small portion encrypted, e.g., with 1% encrypted starting at 2%, is recommendable. The actual encryption percentage can be adjusted to the desired level of security, but there is a trade-off of applicability for security.

8. REFERENCES

[1] Digital Cinema Initiatives and LLC (DCI). Digital cinema system specification v1.0. online presentation, July 2005.

[2] J. Apostolopoulos, F. Dufaux S. Wee, T. Ebrahimi, Q. Sun, and Z. Zhang. The emerging JPEG2000 security (JPSEC) standard. In *IEEE Proc. Int. Symp. on Circuits and Systems (ISCAS)*. IEEE, May 2006.

[3] Jana Dittmann and Ralf Steinmetz. Enabling technology for the trading of MPEG-encoded video. In *Information Security and Privacy: Second Australasian Conference, ACISP '97*, volume 1270, pages 314–324, July 1997.

[4] Jana Dittmann and Ralf Steinmetz. A technical approach to the transparent encryption of MPEG-2

video. In S. K. Katsikas, editor, *Communications and Multimedia Security, IFIP TC6/TC11 Third Joint Working Conference, CMS '97*, pages 215–226, Athens, Greece, September 1997. Chapman and Hall.

[5] F. Dufaux, S. Wee, J. Apostolopoulos, and T. Ebrahimi. JPSEC for secure imaging in JPEG2000. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXVII*, volume 5558, pages 319–330. SPIE, August 2004.

[6] Frederic Dufaux and Touradj Ebrahimi. Securing JPEG2000 compressed images. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXVI*, volume 5203, pages 397–406. SPIE, 2003.

[7] Mark M. Fisch, Herbert Stögner, and Andreas Uhl. Layered encryption techniques for DCT-coded visual data. In *Proceedings (CD-ROM) of the European Signal Processing Conference, EUSIPCO '04*, Vienna, Austria, September 2004. paper cr1361.

[8] M. Grangetto, E. Magli, and G. Olmo. Multimedia selective encryption by means of randomized arithmetic coding. *IEEE Transactions on Multimedia*, 8(5):905–917, 2006.

[9] Raphaël Grosbois, Pierre Gerbelot, and Touradj Ebrahimi. Authentication and access control in the JPEG2000 compressed domain. In A.G. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, pages 95–104, San Diego, CA, USA, July 2001.

[10] S. Imaizumi, O. Watanabe, M. Fujiyoshi, and H. Kiya. Generalized hierarchical encryption of JPEG2000 codestreams for access control. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'05)*, volume 2. IEEE, September 2005.

[11] H. Kiya, D. Imaizumi, and O. Watanabe. Partial-scrambling of image encoded using JPEG2000 without generating marker codes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, volume III, pages 205–208, Barcelona, Spain, September 2003.

[12] T. Kunkelmann and U. Horn. Partial video encryption based on scalable coding. In 5^{th} *International Workshop on Systems, Signals and Image Processing (IWSSIP'98)*, pages 215–218, Zagreb, Croatia, 1998.

[13] Thomas Kunkelmann. Applying encryption to video communication. In *Proceedings of the Multimedia and Security Workshop at ACM Multimedia '98*, pages 41–47, Bristol, England, September 1998.

[14] B. Macq and J.J. Quisquater. Digital images multiresolution encryption. *The Journal of the Interactive Multimedia Association Intellectual Property Project*, 1(1):179–206, January 1994.

[15] Benoit M. Macq and Jean-Jacques Quisquater. Cryptology for digital TV broadcasting. *Proceedings of the IEEE*, 83(6):944–957, June 1995.

[16] Y. Mao and M. Wu. Security evaluation for communication-friendly encryption of multimedia. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapore, October 2004. IEEE Signal Processing Society.

[17] Roland Norcen and Andreas Uhl. Selective encryption of the JPEG2000 bitstream. In A. Lioy and D. Mazzocchi, editors, *Communications and Multimedia Security. Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, CMS '03*, volume 2828 of *Lecture Notes on Computer Science*, pages 194 – 204, Turin, Italy, October 2003. Springer-Verlag.

[18] A. Said. Measuring the strength of partial encryption schemes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'05)*, volume 2, September 2005.

[19] Thomas Stütz and Andreas Uhl. Image confidentiality using progressive JPEG. In *Proceedings of Fifth International Conference on Information, Communication and Signal Processing, ICICS '05*, pages 1107–1111, Bangkok, Thailand, December 2005.

[20] Thomas Stütz and Andreas Uhl. Transparent image encryption using progressive JPEG. In S.K. Katsikas et al., editors, *Information Security. Proceedings of the 9th Infomation Security Conference (ISC'06)*, volume 4176 of *Lecture Notes on Computer Science*, pages 286–298. Springer Verlag, September 2006.

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

[22] A. Uhl and Ch. Obermair. Transparent encryption of JPEG2000 bitstreams. In P. Podhradsky et al., editors, *Proceedings EC-SIP-M 2005 (5th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services)*, pages 322–327, Smolenice, Slovak Republic, 2005.

[23] A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.

[24] H. Wu and D. Ma. Efficient and secure encryption schemes for JPEG2000. In *Proceedings of the 2004 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, pages 869–872, May 2004.

[25] Yongdong Wu and Robert H. Deng. Progressive protection of JPEG2000 codestreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapure, October 2004. IEEE Signal Processing Society.

[26] C. Yuan, B. B. Zhu, M. Su, Y. Wang, S. Li, and Y. Zhong. Layered access control for MPEG-4 FGS. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, Barcelona, Spain, September 2003.