Transparent Image Encryption Using Progressive JPEG *

Thomas Stütz and Andreas Uhl

Department of Computer Sciences
Salzburg University, Austria
e-mail: {tstuetz,uhl}@cosy.sbg.ac.at

Abstract. Many application scenarios do not demand confidential encryption of visual data, but on the contrary require that certain image information is public (transparent encryption). One scenario is e.g., Pay-TV, where a low quality version should become public to attract possible customers. Transparent encryption can be implemented most efficiently in case of scalable bitstreams by encrypting enhancement layer data and baseline JPEG is therefore not well suited for designing such encryption schemes in an efficient manner. This paper investigates how transparent encryption can be realized through selective encryption of the progressive JPEG modes. The traditional approach which encrypts enhancement layers starting at the end of the bitstream suffers from high computational load. Encryption schemes with significantly reduced encryption effort are shown to deliver equivalent image quality and security.

1 Introduction

Encryption schemes for multimedia data need to be specifically designed to protect multimedia content and fulfil the application requirements for a particular multimedia environment [14].

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 [8]). 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 [8] introduce the term "transparent encryption" mainly in the context of digital TV broadcasting: a broadcaster of pay TV does not always intend to prevent

* The work described in this paper is partially supported by the Austrain Science Fund, project no. 15170 and by the Austrian Grid Project, funded by the Austrian BMBWK (Federal Ministry for Education, Science and Culture) under contract GZ 4003/2-VI/4c/2004.

unauthorised viewers from receiving and watching his program, but rather intends to promote a contract with nonpaying watchers. This can be facilitated by providing a low quality version of the broadcasted program for everyone, only legitimate (paying) users get access to the full quality visual data. This is meant also by the term "try and buy" scenario. Therefore, privacy is not the primary concern in such an environment. The simplest approach to achieve this would be to simply 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.

Transparent encryption usually transmits a high quality version of the visual data to all possible viewers but aims at protecting the details of the data which enable a pleasant viewing experience in an efficient manner. If this data are missing, the user is (hopefully) motivated to pay for the rest of the data which may be accessed upon transmission of the required key material by the broadcaster. Another application area of transparent encryption are preview images in image and video databases. Therefore, there are two major requirements that have to be met concurrently:

- To hide a specific amount of image information (security requirement).
- To show 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.

To implement transparent encryption, Macq and Quisquater [8] propose to use line permutations in the transform domain of a lossless multiresolution transform. The permutations are only applied in the region of the transform domain corresponding to fine grained details of the data. Droogenbroeck and Benedett [4] propose to encrypt bitplanes of the binary representation of raw image data, contrasting to the privacy focused approach they suggest to start with the LSB bitplane. With respect to JPEG encoded images, the authors suggest to encrypt sign and magnitude bits of medium and high frequency DCT coefficients (note that this is again exactly just the other way round as compared to encrypting low frequency coefficients only for privacy protection [2, 7]). Bodo et al. [1] propose a technique called "waterscrambling" where they embed a watermark into the motion vectors of an MPEG stream, thereby reducing the video quality significantly – only a legitimate user has access to the key and may descramble the motion vectors.

Transparent encryption may be implemented in the simplest way in the context of scalable or embedded bitstreams. Transparent encryption is achieved in this environment by simply encrypting the enhancement layer(s). This has been proposed by Kunkelmann and Horn using a scalable video codec based on a spatial resolution pyramid [7, 6] and by Dittmann and Steinmetz [3] using a SNR scalable MPEG-2 encoder/decoder. Yuan et al. [16] propose to use MPEG-4

FGS for transparent encryption, JPEG2000 transparent encryption is discussed in own earlier work [13].

The baseline JPEG format does not fit well into the transparent encryption scenario. For example, in order to selectively protect high frequency AC coefficients of a JPEG image (as discussed for example by Droogenbroeck and Benedett [4]), the file needs to be parsed for the EOB symbols 0x00 to identify the end of a 8×8 pixels block where the VLC codewords corresponding to these coefficients will be located (with two exceptions: if 0xFF is followed by 0x00, 0x00 is used as a stuffbit and has to be ignored and if AC63 (the last AC-Coefficient) does not equal 0 there will be no 0x00 and the AC coefficients have to be counted). It is clear that transparent encryption will be fairly inefficient under these circumstances where a significant parsing overhead is introduced.

In this work we systematically investigate the different JPEG progressive modes as defined in the JPEG extended system [10] with respect to their usefulness for providing efficient and yet secure transparent encryption schemes. Section 2 reviews the three modes which are also compared to the JPEG baseline system in terms of compression performance and data organisation. Section 3 finally discusses the respective suitability in a transparent encryption context, the paper is concluded in Section 4.

2 Progressive JPEG Modes

The basic idea of JPEG-based progressive coding [5, 12] is to organize the data into a base layer which contains a low quality approximation to the original data and several enhancement layers which, if combined with the base layer, successively improve the quality.

The three JPEG progressive modes are defined as follows (JPEG uses the term "scan" instead of layers, the first two modes are often denoted as sequential progressive modes):

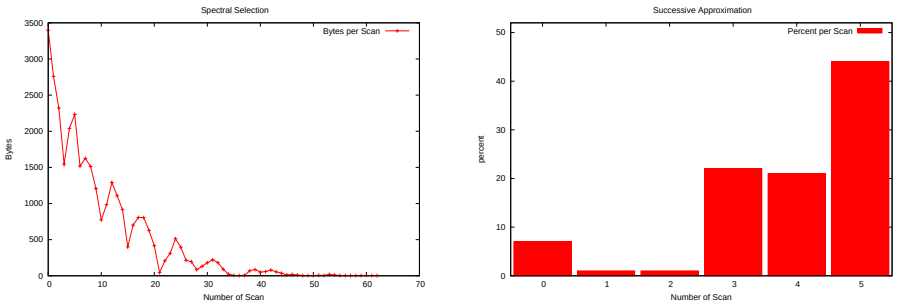
- Spectral selection: the first scan contains the DC coefficients from each block of the image, subsequent scans may consist of a varying number of AC coefficients, always taking an equal number from each block. A typical choice is to encode all DC coefficients into the first scan, subsequently groups of 6 and 7 AC coefficients are organized into one scan.
- Successive approximation: scans are organized according to the binary representation of the coefficients. The first 6 bit of a coefficient is the smallest fraction which the JPEG standard allows to specify. This fraction is coded as in baseline JPEG, while the following bits are emitted without coding. According to the standard DC and AC coefficients have to be treated separately. A typical setting is to use 6 scans (first 6 bit of all DC coefficients Huffman coded, 1 bit more of DC coefficient data, 1 bit more of DC coefficient data, first 6 bit of all AC coefficients Huffman coded, 1 bit more of AC coefficient data, 1 bit more of AC coefficient data).
- Hierarchical progressive mode: an image pyramid is constructed by repeated weighted averaging and downsampling. The lowest resolution approximation

is stored as JPEG (i.e. the first scan), reconstructed, bi-linearly upsampled, and the difference to the next resolution level is computed and stored as JPEG file with possibly different quantization strategy (similar to P and B frames in MPEG). This is repeated until the top level of the pyramid is reached.

The JPEG standard also allows to mix different modes. Note that the three modes allow a different amount of scans. Whereas spectral selection offers a maximum of 64 scans, the hierarchical progressive mode is restricted to 5 – 6 sensible scans (given a $2^8 \times 2^8$ pixels image). Successive approximation is restricted to 6 scans (assuming 8 bpp grayscale data). Similar to the scalability profiles of MPEG-2, the JPEG progressive modes are not used very much and are poorly supported and documented in commercial software.

Although providing much better functionality for transmission based applications, the compression performance could be expected to decrease using JPEG progressive modes. This would of course not favour the use of these techniques in transparent encryption scenarios. We have shown in earlier work [12] that provided coding options are chosen carefully, compression performance equivalent to and even exceeding the baseline system may be achieved. All tests concerning the sequential progressive modes were conducted using the IJG's (Independent JPEG Group) reference library, the hierarchical mode is a custom implementation based on the IJG software [12]. All results in this work employ the Lena image with 512^2 pixels and 8bpp.

Figs. 1(a) and 1(b) show how the different scans contribute to the overall file size. This is important knowledge for subsequent transparent encryption since we want to design computationally efficient schemes. In the spectral selection case (Fig. 1(a)) each scan contains one coefficient from each block and as it is expected, the size of the scan decreases for increasing coefficient frequency.



(a) Spectral Selection (b) Successive Approximation

Fig. 1. Data distribution across different scans for sequential progressive modes.

In our example, successive approximation uses the scan configuration used as an example above. We realize that the two scans containing the single DC

coefficient bits do not contribute much to the overall file size, whereas the three scans corresponding to single AC coefficient bits contribute 21% and 44% to the overall data.

Table 1 shows two examples for the hierarchical JPEG case using 6 scans (6 pyramid levels), the first optimized for good compression performance (note that in this case the quality of the base layer needs to be low, in our example it is set to $q_f = 10$ [12] resulting in a total of 48589 bytes), the second optimized for a high quality base layer ($q_f = 95$, resulting in a total of 53113 bytes).

Layer	0	1	2	3	4	5
$q_f = 10$	0.6%	0.7%	0.8%	1.3%	2.6%	94.1%
$q_f = 95$	0.7%	1.3%	2.7%	6.9%	19.0%	69.4%

Table 1. Percentage of the overall file size contributed by the single layers.

It is clearly visible that the distribution of the amount of data among the scans differs significantly depending on the compression settings. This also implies that encrypting e.g. layer 4 only imples a variation in encryption amount between 2.55% and 18.89% of the entire data (which is rather significant since the overall data volume differs only by 10% in our example).

3 Transparent Encryption

3.1 The Classical Approach

The classical approach for transparent encryption of visual data in layered representation is to simply encrypt the enhancement layers, successively encrypting more and more data starting at the end of the file. The remaining scans (i.e. the base layer or scans left in plaintext) may be expected to contain data corresponding to the visual information in lower quality. As we shall see, this approach implies that large amounts of data need to be encrypted to provide a sufficient quality decrease. As an alternative we will investigate strategies where visually more important data, which is not located in the last portions of enhancement information, is encrypted first. The goal is to have similar results as compared to the classical approach but less encryption effort.

Note that in most transparent encryption scenarios the encryption of DC coefficient data has to be avoided since otherwise luminance information is entirely or partially lost and the result is a severely alienated image which might refrain a potential costumer from getting interested in the data (quality requirement is not met). This immediately results in a lower bound in achievable image quality that may be achieved with the sequential progressive modes: this bound is attained by reconstructing the image based on DC coefficient data only as shown in Figs. 4 and 6. The situation is more complicated for the hierarchical mode due to the flexibility in its coding parameters. Depending on the quality of the base layer (and the depth of the pyramid) employed, reconstructions using the

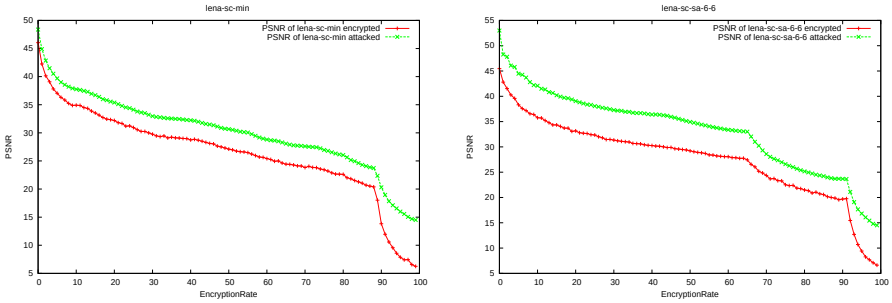
image pyramids' base only may vary to a large extent in quality (see Fig. 2 for corresponding examples of a two layer pyramid with high quality $q_f = 95$ and low quality $q_f = 10$ base layers).

Fig. 2. Image reconstruction based on the base layer only.

Decoding a partially encrypted image by treating the encrypted data as being unencrypted leads to images severely degraded by noise type patterns (which originate from the encrypted parts). Using these images to judge the security of the system leads to misinterpretations since a hostile attacker can do much better. In particular, an attacker could simply ignore the encrypted parts (which can be easily identified by statistical means) or replace them by typical non-noisy data. This kind of attack is called "error-concealment" [15] or "replacement attack" [11] in the literature. The IJG software ignores empty scans during decoding – therefore a simple error concealment attack sets the scans affected from encryption simply to zero. In the hierarchical JPEG case we set residual pyramid levels to zero if affectd by encryption, the base layer is replaced by uniform gray value 128. See also [5, 12] for these attacks against DCT-based coding/encryption schemes. In order to assess the quality of the visual material after reconstruction in addition to visual inspection we use PSNR and ESS (Edge Similarity Score [9]), the latter measuring the similarity of dominating edges on a block basis in the range $[0, 1]$.

Figs. 3(a) and 3(b) show PSNR values when starting encryption at the end of the file and successively increasing the amount of data encrypted, for spectral selection and successive approximation, respectively, using direct reconstruction and under an error concealment attack. We clearly note the effect of the attack which improves the reconstructions by 4 – 5 dB.

The result curves bend sharply towards lower quality when DC coefficient data is reached for both cases at about 90% of the data encrypted (which again documents that encrypting DC coefficient data violates the quality requirements

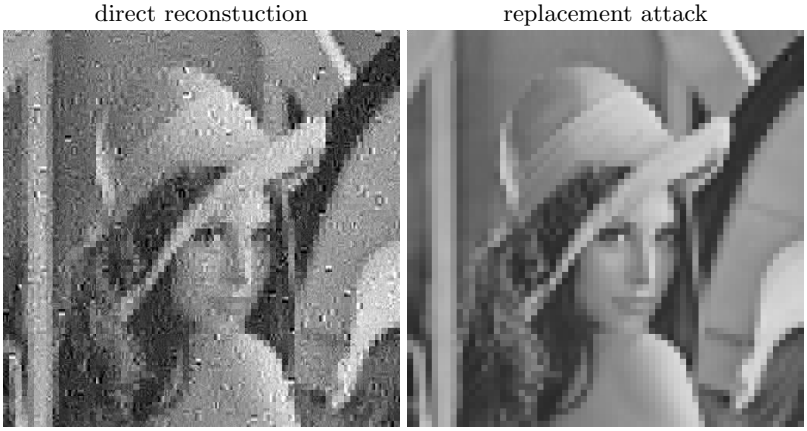


(a) Spectral selection (b) Successive Approximation

Fig. 3. Increasing the amount of data encrypted.

of transparent encryption), in the successive approximation case we additionally observe different behaviour also when the 6 significant AC coefficient bits are met at about 65% of the data encrypted.

Fig. 4 gives a visual example of the effectiveness of the conducted attack against transparent encryption of 89% spectral selection data. The attack improves the visual quality and PSNR values considerably.



PSNR=20.39,ESS=0.18 PSNR=23.74,ESS=0.16

Fig. 4. Transparent encryption of spectral selection (89% encrypted).

This example shows a dilemma which makes the parameters for transparent encryption difficult to adjust. If the amount of encryption is selected to deliver optimal quality without the assumption of a conducted attack (not too good to motivate viewers to pay for better quality and not too low to raise the viewers' interest in the material), the quality is too high after a successful attack has been

mounted. In this scenario, customers able to perform a corresponding attack will probably do so instead of paying. In case the amount of encryption is adjusted to deliver optimal quality assuming an attack has been mounted, the non-attacked material is of rather low quality and might not be suited to raise the average viewers' interest. Therefore, a compromise between those two strategies has to be found. Also, the decision which strategy is applied of course also depends on the business model and the target customer group of the overall application scenario.

3.2 Reducing Encryption Effort

We have seen that the necessity of meeting the security requirement leads to the encryption of a large amount of data in case encryption preceeds from the end of the file to the beginning as done traditionally. However, since the last layers do not contribute much to the image quality, it may be more reasonable not to start encrypting at the end of the data, but at a specific point after the DC data according to the required image quality. For most applications starting right after the DC data will be appropriate, in order to minimize the image quality and the encryption rate. Fig. 5 shows an example where encrypting the first AC coefficient only (7% of the file size) results in almost the same image quality as when encrypting 70% of the data starting from the end of the file in the case of spectral selection. We result in a much more efficient transparent encryption scheme employing this idea.

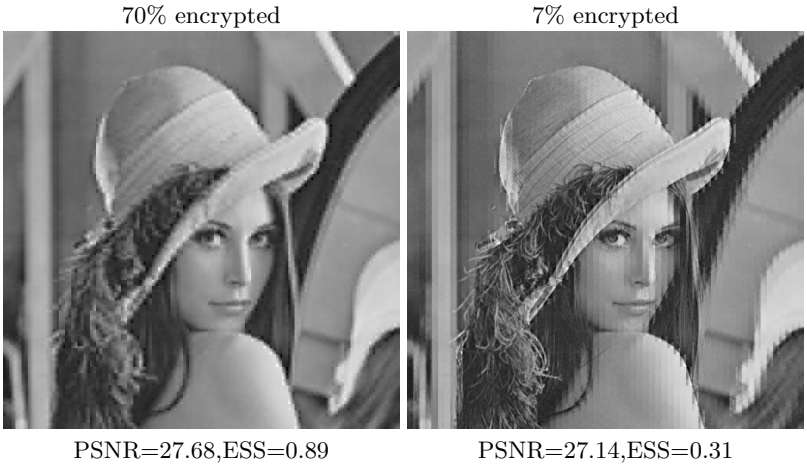


Fig. 5. Efficient transparent encryption of spectral selection (after attack).

The same observations may be made and similar solution strategies can be applied in case of successive approximation. Using the traditional approach, large quantities of data need to be protected to meet both security and quality requirements, respectively (see for example Fig. 6 where 92% of the data are

encrypted: all AC data plus the two single DC coefficient bit scans). Considering the results shown in Fig. 3(b), it is evident that the scan containing the 6 bit AC coefficient data mainly influences the image quality. However, when encrypting this scan we have to process 22% of the overall data accordingly, which is still too much for most applications. One possibility is to split up this scan using spectral selection: Fig. 6 shows results for the encryption of the leading 5 AC coefficients (only the 6 most significant bits, which represent 14% of coefficient data, are encrypted), which leads to similar image quality as the encryption of about 92% of the data with the traditional approach. Again, we were able to significantly reduce the encryption effort as compared to the traditional technique.

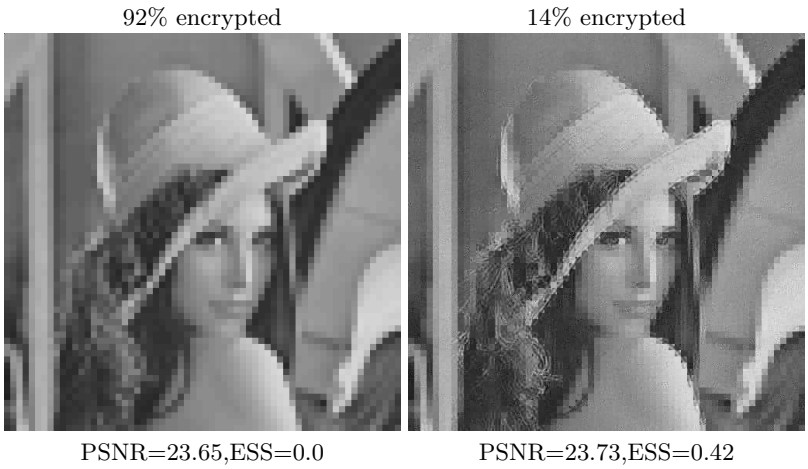


Fig. 6. Efficient transparent encryption of successive approximation (after attack).

Contrasting to the sequential progressive JPEG modes, hierarchical JPEG can offer a great flexibility in its coding parameters. However, as we have seen in the example of Fig. 2, using the traditional approach of encrypting enhancement layers starting at the end of the file at the highest layer) requires the encryption of 70% or 94% of the overall data in the examples of the two layer scenario. When the number of layers is increased, a higher and higher percentage of data has to be encrypted using this technique. Therefore we apply the same principle as discussed before and encrypt scans between the base layer (layer 0) and the highest enhancement layers. Table 2 shows corresponding results when this idea is applied to the two variants of 6-level pyramids given as an example at the end of section 2 (and we also provide the amount of data in percentages contained in the different layers).

Results in the table indicate that we may reduce the neccessary encryption amount significantly using this approach. However, we notice an enormous gap in the quality results between the direct reconstruction and the result obtained by the error concealment attack. Note the extreme example when encryption layer 3 where the difference in PSNR between the directly reconstructed image and the attacked version is more than 10 dB ! This fact makes it extremely

Layers encrypted	0	1	2	3	4	5
$q_f = 10$, % enc.	0.6	0.7	0.8	1.3	2.6	94.1
PSNR direct	12.7	18.4	17.9	11.8	13.8	8.8
PSNR attacked	18.4	19.5	21.0	21.9	22.9	24.6
ESS direct	0.69	0.56	0.43	0.39	0.42	0.35
ESS attacked	0.81	0.61	0.53	0.49	0.57	0.47
$q_f = 95$, % enc.	0.7	1.3	2.7	6.9	19.0	69.4
PSNR direct	16.7	20.5	20.0	14.5	8.5	8.8
PSNR attacked	18.4	22.9	25.1	26.9	26.6	23.9
ESS direct	0.64	0.51	0.40	0.43	0.47	0.53
ESS attacked	0.77	0.66	0.62	0.67	0.68	0.79

Table 2. Results of protecting various layers when applied to the compression optimized pyramid ($q_f = 10$) and to the quality optimized pyramid ($q_f = 95$).

difficult to adjust the encryption parameters properly to meet both, security and quality requirements (which is a difficult task in any case as discussed earlier): the quality of a directly reconstructed image must not be too low, otherwise the average customer will lose interest; but an attacker in this case will succeed in generating a high quality version.

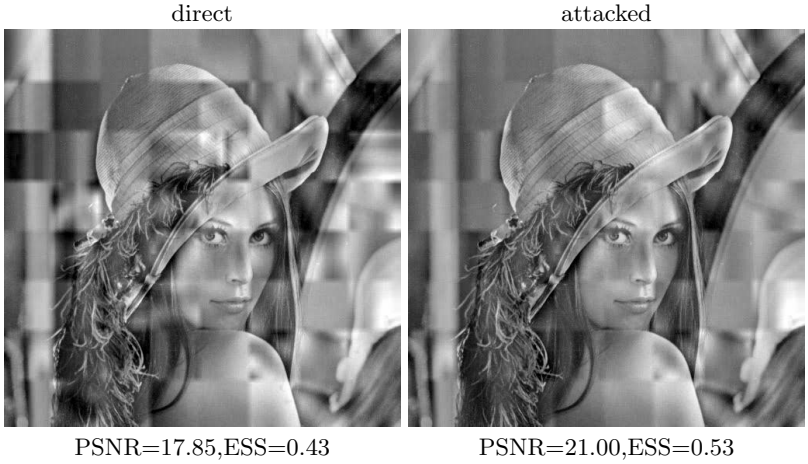


Fig. 7. Encryption of layer 2 (0.80% encrypted), compression optimized pyramid.

As a consequence, for real life applications we have to rely on settings minimizing this quality gap. Table 2 shows that this gap is by far less pronounced when layer 1 or 2 are encrypted only. Fig. 7 displays the case of encrypting layer 2 for the compression optimized pyramid ($q_f = 10$), Fig. 8 shows the case of encrypting layer 1 for the quality optimized pyramid ($q_f = 95$). If the quality requirements are met for the target application, these settings are a very good choice since the encryption effort is very small (0.80% and 1.25% of the overall

file size) and the security is rather satisfactory since the discussed gap is rather small in these cases.

Fig. 8. Encryption of layer 1 (1.25% encrypted), quality optimized pyramid.

Table 2 reveals an additional property when avoiding to encrypt the highest enhancement layer(s): as can be seen, the gap between quality using direct reconstruction and attacked visual data is maximal when the high layers are encrypted. Therefore, besides reducing the encryption amount as suggested in this work we also improve the applicability in real-word scenarios of the scheme by encrypting layers more closely to the base layer.

4 Conclusions

Progressive and hierarchical JPEG may be used for transparent encryption in an efficient manner due to the scalable data format. Parsing the file and searching for the data to be protected can be avoided in this fashion. We have shown that the traditional approach applied to scalable data which starts encryption from the end of the bitstream (enhancement layer encryption) suffers from high encryption demands. The same functionality can be achieved by protecting data situated between base and enhancement layers while reducing the computational encryption effort significantly.

References

[1] Y. Bodo, N. Laurent, and J.-L. Degelay. A scrambling method based on disturbance of motion vector. In *Proceedings of ACM Multimedia 2002*, pages 89–90, Juan Le Pins, France, December 2003.

[2] H. Cheng and X. Li. On the application of image decomposition to image compression and encryption. In P. Horster, editor, *Communications and Multimedia Security II, IFIP TC6/TC11 Second Joint Working Conference on Communications*

and Multimedia Security, CMS '96, pages 116–127, Essen, Germany, September 1996. Chapman & Hall.

[3] 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.

[4] Marc Van Droogenbroeck and Raphaël Benedett. Techniques for a selective encryption of uncompressed and compressed images. In *Proceedings of ACIVS (Advanced Concepts for Intelligent Vision Systems)*, pages 90–97, Ghent University, Belgium, September 2002.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] W.B. Pennebaker and J.L. Mitchell. *JPEG – Still image compression standard*. Van Nostrand Reinhold, New York, 1993.

[11] M. Podesser, H.-P. Schmidt, and A. Uhl. Selective bitplane encryption for secure transmission of image data in mobile environments. In *CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002)*, Tromso-Trondheim, Norway, October 2002. IEEE Norway Section. file cr1037.pdf.

[12] 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.

[13] 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.

[14] 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.

[15] Jiangtao Wen, Mike Severa, Wenjun Zeng, Max Luttrell, and Weiyin Jin. A format-compliant configurable encryption framework for access control of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):545–557, June 2002.

[16] 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.