Bit Domain Encryption

Anil Yekkala and C.E. Veni Madhavan

[1] NXP Semiconductors India Ltd., Bangalore
anil.yekkala@philips.com
[2] Indian Institute of Science, Bangalore
cevm@csa.iisc.ernet.in

Abstract. In recent years we have seen significant growth in multimedia based Internet applications and multimedia commerce. The advancements in multimedia commerce resulted has in a rapid growth, in the amount of multimedia data transferred over network. The multimedia data stored on the web servers as well as the data transferred over the network needs to be protected from piracy and eavesdropping. Hence, there is a strong need for encrypting the multimedia content. But owing to the size of the multimedia content and real time requirements for encoding, transmitting and decoding the multimedia content, usage of standard encryption/decryption schemes prove to be an overhead. Hence lightweight encryption schemes are gaining popularity. The schemes are designed using structure of the multimedia content, and partially encrypting the content such that it results in insertion of sufficient noise to make the content unintelligible. In this paper we present a scalable and secure lightweight encryption scheme for image and video data in bit domain. The scheme in addition to being secure and scalable has negligible impact on compression.

1 Introduction

The rapid growth in multimedia based Internet systems and applications like video telephony, video on demand, network based DVD recorders and IP television has created a substantial need for multimedia security. One of the important requirements for multimedia security is transmission of the digital multimedia content in a secure manner using encryption for protecting it from eavesdropping. The traditional method of encrypting multimedia content is to consider the two-dimensional/three-dimensional image/video stream as a one-dimensional stream and to encrypt the entire content using a standard block cipher like AES or DES [MOV01] or using a stream cipher.

Even though the approach of encrypting the entire content provides the desired security requirements, it imposes a large overhead on the multimedia codex. This is due to the size of the multimedia content, and also due to real time requirements of transmission and rendering. Hence, lightweight encryption schemes are gaining popularity for multimedia encryption. Lightweight encryption schemes are based on the principle "Encrypt minimal and induce maximum noise". Lightweight encryption schemes are designed to take the structure of

J. Blanc-Talon et al. (Eds.): ACIVS 2008, LNCS 5259, pp. 45–56, 2008.

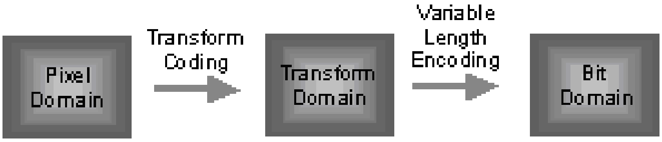

Fig. 1. Typical lossy image encoder

the multimedia content into consideration. The various stages involved in image compression are shown in Figure1.

1.1 Current Literature

Several lightweight encryption schemes [FSE04] for image and video exist in the literature. Most of the existing lightweight encryption schemes are based on encrypting the image and video content in its DCT domain [BSW02], [T96]. It has been found [SSS], [QNT97], that the schemes proposed in DCT domain either have an impact on the compression or fail to provide the desired level of security and scalability. Whereas, pixel domain scheme proposed by Podessar et al. [PSU02] and a modified scheme proposed by Yekkala et al. [YU07] provide a scalable encryption scheme. The pixel domain schemes are based on dividing the image into eight bit planes, but the schemes can not be used if the data has to be compressed. This is due to fact that encryption destroys the statistical redundancy within the image to large extent. Hence encryption in pixel domain will have severe impact on compression achieved. An encryption scheme proposed by Qiao et al. [QN97] exists in bit domain, which uses the statistical properties of MPEG data stream. Even though the scheme is considered to be very secure and has no impact on compression, the scheme fails to provide scalability. It also requires 50% of entire data to be encrypted. Hence, in cases where security is less significant compared to real time performance, the scheme may not be applicable.

2 Proposed Scheme

2.1 Design of Encryption Scheme

One of desirable features for a lightweight encryption scheme is to have minimal impact on the compression achieved. Therefore, the encryption scheme for image and video data must be designed in such a way that the statistical properties of the data, which is exploited by the compression scheme, do not get altered too much due to the application of encryption. In case of lossy compression the compression is obtained in two stages. The first stage of compression involves transforming the data signal into a transformed domain followed by quantization. The second stage involves entropy encoding on the quantized coefficients. Hence, the application of encryption in spatial domain or in transform domain will have an impact on the amount of compression that can be achieved.

Table 1. Average number of bits occupied by DC value

Image	Average number of bits
Baboon	4.5
Bandon	6.8
Lena	6.9
Opera	5.6
Peppers	6.8
Pills	7.4

Hence, based on the discussions the best option for encryption without impacting compression is to encrypt the image in bit domain. But direct application of encryption in bit domain is not feasible since the Huffman codewords will get impacted, and these Huffman codewords are used by the decoder. Hence, decoding may fail even with the knowledge of the key. Hence, an approach can be to encrypt a fixed number of most significant bits (MSB) in each block after entropy encoding. Selection of MSB bits for encryption is important, since the MSB bits occupy information related to DC value and lower frequency AC values, which contain the maximum energy within an image.

It is to be noted that in order to encrypt a fixed number of bits in each block, only 6 MSB bits can be encrypted in each block. This is due to the fact that in JPEG the length of the block varies from block to block, and a minimum length of the block can be 6 bits i.e. 2 bits for the DC differential and 4 bits for the EOB indicator (this case arises when the quantized DC difference of the block as well as all the quantized AC values are equal to zero), whereas the maximum length of the block can be well above 100 bits.

But encrypting only 6 MSB bits of each block may not provide desired level of security. This is due to the fact that on an average in each block 6 bits is needed to represent the encoded DC differential value (refer to Table1). Hence, encrypting only 6 MSB bits in each block will be equivalent to encrypting only the DC value within a block in most of the cases, and it can be considered to be weak due to the linear and orthogonal properties of DCT transformation as discussed in [YVU07]. Therefore, encrypting only a fixed number of bits may not provide desired level of security and moreover such a scheme cannot be scalable based on security requirements.

In order to design an efficient encryption scheme in bit domain, which in addition to being secure will be scalable and will have minimal impact on compression, we propose the following

- The number of MSB bits to be encrypted within a block will depend upon the intelligibility of the block.
- Maximum intelligibility within an image is present in its edges. Hence blocks containing edges are considered to be important for purpose of encryption.
- In order to create minimum intelligibility within an encrypted image, higher number of MSB bits will be encrypted in edge blocks (i.e. blocks containing edges) after encoding compared to non-edge blocks.

- Edge blocks will be identified based on the number of bits used to represent an encoded block. If the number of bits is above a predefined selected threshold, the block is assumed to contain an edge. This is due to the fact that blocks containing edges will have more non-zero AC values or significantly higher AC values. Hence the number of bits required to encode such blocks will be higher due to usage of entropy encoding (run length encoding using predefined Huffman codewords).
- A minimum threshold (say m) for length of the encoded block will be defined for determining an edge block. Edge blocks (i.e. blocks containing edges) will be identified by checking the length of the encoded block. If the length is above m bits then the corresponding block will be considered to be an edge block.
- Encryption will be performed on encoded blocks by encrypting m MSB bits in case of edge blocks (it is to be noted that length of edge blocks will be atleast m bits), and n bits ($0 \leq n \leq 6$) in case of non-edge blocks (it is to noted that length of encoded block is atleast 6 bits).
- Two codewords will be used for indicating the end of block indicator namely $EOB0$ and $EOB1$ instead of single default EOB value (generally binary $1010b$). $EOB0$ will be used by the encoder to indicate to the decoder that in the subsequent block only n MSB bits are encrypted, whereas $EOB1$ will be used to indicate to the decoder that in the subsequent block m MSB bits are encrypted.
- If a block does not contain an end of block indicator (i.e. when the 63rd AC coefficient is non-zero), only n MSB bits in the subsequent block will be encrypted, irrespective of the length of the subsequent block.
- The length of the Huffman codewords for the end of block indicators $EOB0$ and $EOB1$ in the modified Huffman table will be 5 bits instead of default 4 bits.
- The value of n can vary from 0 to 6 bits, whereas the value of m can be decided based on the level of security needed.
- Only n MSB bits are encrypted for the first block, irrespective of the length of the block. This is due to the fact that decoder will not have the knowledge on number of bits present in the first block.

2.2 Encoding and Encryption Scheme

The procedure for incorporating the proposed lightweight encryption in bit domain, while encoding an 8-bit grey image using JPEG is explained in following subsections.

Inputs

- The input image in form of an 8-bit pixel array. The size of the array will depend upon the resolution of the image.
- Quantization table for quantizing DCT coefficients.
- Modified Huffman table for Luminance coefficients (i.e DC differences and AC values) with a codeword for $EOB0$ and $EOB1$.

- Threshold m to be used for detecting edge blocks and also to indicate the number of MSB bits to be encrypted in such edge blocks.
- Threshold n $(0 \leq n \leq 6)$ indicating number of MSB bits to be encrypted in non-edge blocks.
- Encryption algorithm and its corresponding parameters including the key and initialization vector. The encryption algorithm can be either a standard block cipher like AES in OFB mode or a stream cipher.

Procedure. For simplicity of explanation, it is assumed that the $63rd$ AC coefficient in each block will be equal to zero, hence each block will require an EOB indicator. In practice if the $63rd$ AC coefficient is present in a block, in the subsequent block only n bits will be encrypted.

1. The entire image is divided into blocks of size 8×8 pixels.
2. DCT transformation is applied on each block, followed by quantization and then differential encoding on DC values.
3. Entropy encoding is applied on the quantized DCT coefficients, block by block, starting from the first block in the first row namely $I_{1,1}$.
4. Quantized coefficients of the block $I_{1,1}$ are then encoded using run length encoding based on the input Huffman tables, and 5 zero bits are kept reserved for the end of block indicator, whose codeword will be determined based on the length of the next block.
5. Quantized coefficients of the block $I_{1,2}$ are then encoded similar to $I_{1,1}$ using run length encoding based on the input Huffman tables, and 5 zero bits are kept reserved for the end of block indicator, and also length of the encoded block $I_{1,2}$ is determined.
6. If the length of the encoded block $I_{1,2}$ is above m bits, then the bits reserved for end of block indicator for $I_{1,1}$ will be replaced by the five bit codeword $EOB1$ otherwise by the five bit codeword $EOB0$.
7. n MSB bits of the encoded block $I_{1,1}$ will be encrypted.
8. Quantized coefficients of the block $I_{1,3}$ will then be encoded and its length will be determined. Based on the length, the bits reserved for end of block indicator of $I_{1,2}$ will be replaced by either $EOB0$ or $EOB1$.
9. If the length of the block $I_{1,2}$ is above m bits, then m MSB bits of encoded block $I_{1,2}$ will be encrypted, otherwise n MSB bits of encoded block $I_{1,2}$ will be encrypted.
10. Proceeding in a similar manner all the blocks will be encoded and encrypted subsequently row by row, starting from the leftmost block in each row.

It is to be noted that encryption can be carried out on an already encoded image by first performing partial decoding (i.e. Huffman decoding) to determine the length of the blocks. Then we encrypt either m or n MSB bits of a block based on the length of the block, and replace the end of block indicators by either $EOB0$ or $EOB1$ based on the length of next block. Thus, the encryption scheme can be also used as pluggable application for JPEG.

2.3 Decoding and Decryption Scheme

The procedure for incorporating proposed decryption algorithm in bit domain, while decoding an 8-bit grey image using JPEG is explained in following subsections.

Input

- Encoded and encrypted image.
- The threshold values m and n.
- Decryption algorithm and its corresponding parameters.

Procedure

1. The blocks are processed sequentially row by row starting from the leftmost block in the first row.
2. The n MSB bits of the encoded block $I_{1,1}$ are decrypted (i.e. for the first block only n MSB bits are encrypted).
3. The encoded block $I_{1,1}$ is decoded using entropy decoding and the quantized DCT coefficients are obtained along with the codeword of end of block indicator.
4. If the codeword of end of block indicator of the block $I_{1,1}$ is $EOB1$, then m MSB bits of the encoded block $I_{1,2}$ are decrypted, otherwise if the code word is $EOB0$ then n MSB bits of the encoded block $I_{1,2}$ are decrypted. Then the block is subsequently decoded in standard conventional manner to obtain the pixel values.
5. Proceeding in a similar manner, decryption will be performed on all blocks followed by entropy decoding to obtain the quantized DCT coefficients and the codeword indicating the number of bits encrypted in the next subsequent block.
6. Differential DC decoding is performed to obtain the DC values, followed by de-quantization and then IDCT transformation is applied on each block to obtain the pixel values.

3 Results

The results of the proposed scheme are discussed in terms of four important characteristics of lightweight encryption schemes namely Security, Scalability, Simplicity and Impact on Compression.

3.1 Security

Since the bits are encrypted in bit domain the Huffman codewords will also get encrypted, hence the amount of noise introduced in an encrypted image will be quite enormous for the purpose of security. Moreover the encryption is performed in bit domain. Hence, strength of the scheme will depend upon the underlying encryption algorithm. The impact of encrypting a Lena image using the proposed approach for $m = 24$ and $n = 6$ can be seen in Figure2.

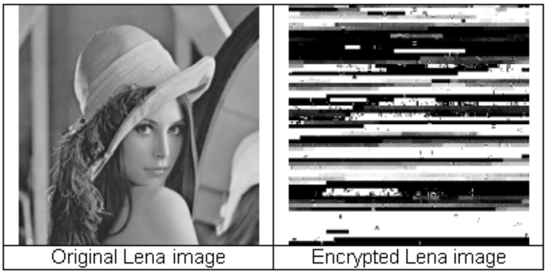

Fig. 2. Bit domain encryption on Lena image

3.2 Scalability and Choice of Paramaters

The scalability of the encryption scheme can be adjusted by adjusting the values of m and values of n.

Table2 shows for some standard images the percentage of bits encrypted for different values of m, when n is kept fixed at 6. It can be observed from the Table2 that percentage of bits encrypted increases initially with increase in size of m, but a point of inflexion appears approximately for most of the images when m is between 50 to 60 bits, and the maximum percentage of bits encrypted for an image is around 50%. The behavior of percentage of bits encrypted with choice of m is shown in Figure3 for Lena image. Hence for fixed value of n=6, the value of m can vary from 8 bits to 52 bits and can offer 12 levels of scalability (i.e. the point of inflexion for m appears, when the value of m is between 50 and 60). It is to be noted that number of scalability levels can differ from image to image,

Table 2. Percentage of bits encrypted for various choices of m

Choice of m	% of bits encrypted for various images					
	Baboon	Bandon	Lena	Opera	Peppers	Pills
12	8	21	19	16	18	17
16	11	27	25	21	24	23
20	14	33	30	26	30	28
24	17	39	35	30	35	33
28	20	43	39	35	40	36
32	23	47	42	38	43	39
36	26	50	44	41	46	41
40	29	52	45	43	48	44
44	32	53	45	46	49	45
48	35	53	46	48	49	47
52	38	53	47	49	48	48
56	40	52	47	51	47	48
60	42	52	46	52	46	49
64	44	49	45	52	45	49
68	46	47	45	52	44	49

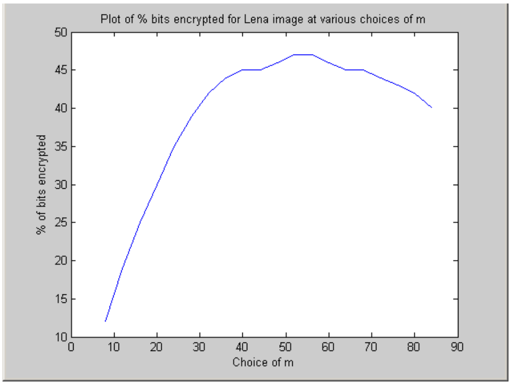

Fig. 3. Plot for percentage of bits encrypted versus choice of m for Lena image

and also based on Quantization table. For example in case of Baboon image 20 levels of scalability can be achieved.

3.3 Impact on Compression

The application of bit domain encryption has minimal impact on compression, and it can be observed from Table3. The impact on compression is caused due to modification of Huffman table for supporting two codewords for end of block indicators.

From Table3 it can be seen that the impact is less then 3.1% for all the images. An alternative approach to have zero impact on compression is to avoid using two Huffman codewords for indicating the number of bits encrypted in the next block. This will not have any impact on the Huffman tables used. But in such a case a protocol must be designed between the encoder and decoder to indicate the blocks where m MSB bits are encrypted and to indicate the block where only n MSB bits are encrypted. This can be done by transmitting a 1-0 matrix, where 1 at index (i, j) of the matrix will indicate that only m MSB bits

Table 3. Impact on compression using bit domain encryption

Image	Resolution	Size after compression using JPEG without encryption (bytes)	Size after compression using JPEG with encryption (bytes)	Impact %
Baboon	512512	68940	70111	1.7
Bandon	610403	24635	25397	3.1
Brandyrose	518744	41647	42941	3.1
Lena	512512	32572	33419	2.6
Opera	695586	60170	61843	2.8
Peppers	512512	33734	34635	2.7
Pills	800519	54762	55791	1.9

of the block at *ith* row and *jth* column have been encrypted, whereas 0 will indicate that only n MSB bits in the corresponding block at *ith* row and *jth* column have been encrypted. Hence, for an image of resolution 512 × 512 the 1-0 matrix size will be of size 64 × 64, requiring an additional data of 512 bytes, which is comparatively less then the impact on compression by modifying the Huffman table. But it may have impact on the security since knowledge of the 1-0 matrix will reveal some information about the image, by giving information on the blocks containing edge and smooth blocks. Hence the 1-0 matrix must be also communicated in a secured manner. Moreover the usage of 1-0 matrix in case of video will increase the complexity of the communication model, since for each frame the sender must send the corresponding 1-0 matrix along with the frame to the receiver.

3.4 Protocol for Encryption and Decryption

Using the proposed scheme a simple protocol is required between the encoder and the decoder. The encoder needs to convey just the parameters m and n along with the encryption algorithms and its corresponding secret keys.

4 Extensions

4.1 Extension of the Scheme for Color Images

The encryption and encoding scheme for the luminance component in color images is similar to gray image, and also uses the parameters m and n specifying the number of MSB bits to be encrypted in edge and non-edge blocks. For the chrominance components two additional parameters are defined namely mc and nc. mc defines the threshold for identifying edge blocks in chrominance components and also the number of MSB bits to be encrypted in such blocks. nc denotes the number of MSB bits to be encrypted in non-edge blocks of chrominance component. The maximum value of nc can be 4, since the minimum length of an encoded chrominance block is 4 bits (2 bits for differential DC and 2 bits for the end of block indicator). Similar, to the luminance components two codewords namely $EOB0c$ and $EOB1c$ of length three bits are used for end of block indicator instead of one codeword of length 2. The first codeword $EOB0c$ indicating that in the subsequent block only nc MSB bits are encrypted, whereas a codeword of $EOB1c$ will indicate that in the subsequent block only mc MSB bits are encrypted.

4.2 Extension of the Scheme for Video

The scheme can be extended for MPEG by encrypting all blocks in I frames and all I blocks (i.e. the intra coded blocks) in P and B frames. The number of bits to be encrypted will again depend upon the length of the respective block, and similar modifications will be required in the Huffman tables to incorporate two codewords for end of block indicator.

Table 4. Proposed compressed domain edge detector versus Sobel-edge detector

Image	No of edge pixels detected using Sobel-edge detector	No of edge pixels detected using proposed edge detector	% of blocks decoded completely using proposed method
Baboon	12620	12618	79%
Bandon	7913	7684	30%
Lena	8254	189	33%
Opera	15463	15162	44%
Peppers	6245	6166	33%
Pills	12820	12708	53%

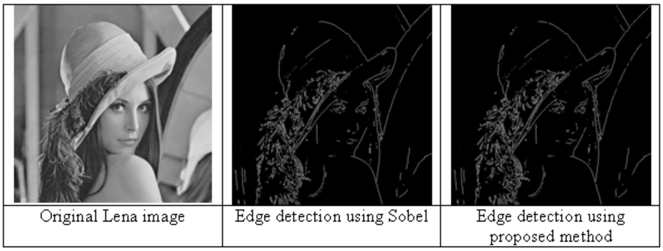
Original Lena image
Edge detection using Sobel
Edge detection using proposed method

Fig. 4. Proposed compressed domain edge detector versus Sobel-edge detector

4.3 Extension of the Scheme for Edge Detection

The scheme can be extended for detecting edges of an image in compressed domain. It can be used for first finding out edge blocks based on the length of the encoded block in bit domain i.e. the number of bits used to represent the encoded block. If the length of the encoded block is above a predefined threshold, then the block is considered to be an edge block and full decoding is performed followed by a standard edge detection technique like Sobel edge detector [P91]. The rest of the non-edge blocks are ignored.

In order to determine length of the encoded block in bits, a partially decoding is required by performing entropy decoding, and bit length of the encoded blocks is computed while decoding. Using a predefined threshold the blocks are categorized into edge and non-edge blocks. If the length of the block is above a predefined threshold, then the block is considered to be an edge block, otherwise it is considered to be a non-edge block. Only the edge blocks will then be further decoded into its pixel domain and standard edge detection techniques like Sobel-edge detector or Laplace edge detector is applied to check for edge pixels. The results of the proposed edge detector are shown in Table4 with respect to direct application of Sobel edge detector. The table shows number of edge pixels detected using standard Sobel edge detection in column (ii), number of edge pixels detected using proposed edge detector in column (iii), and finally the percentage of blocks decoded completely for performing edge detection in column (iv). A fixed threshold of 70 pixels has been used for detecting edge blocks. Table4 clearly shows that most of edge pixels (more then 96%) have been detected using the proposed method, and it also shows that in most of the cases less then 50% of the blocks have been decoded completely, hence providing a computational time advantage by a factor of 2. Even though only 20% computational time has been saved in the case of Baboon image, a larger threshold of encoded block length would have reduced the computational time considerably, without impacting the edge detection algorithm much. The result of edge detection on Lena image is shown in Figure4.

5 Conclusions

Based on the results it can be concluded that proposed lightweight encryption scheme could be used for encrypting image and video content, while not compromising on the security. The proposed scheme allows scalability of the algorithm based on the security needs, and it has been also observed the scheme has a minimal impact on compression. It is also seen that the scheme can be extended for identifying edges in an Image in compressed domain, hence saving significant computational time for edge detection.

References

[BSW02] Bhargava, B., Shi, C., Wang, Y.: MPEG Video Encryption Algorithms (August 2002), `http://raidlab.cs.purdue.edu/papers/mm.ps`

[FSE04] Furht, B., Socek, D., Eskicioglu, A.M.: Fundamentals of Multimedia Encryption Techniques. Multimedia Security Handbook. CRC press, Boca Raton (2004)

[MOV01] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC press, Boca Raton (August 2001) (fifth printing)

[P91] Pratt, W.K.: Digital Image Processing, 2 edn. John Wiley & Sons, Chichester (April 1991)

[PSU02] Podesser, Schmidt, H.P., Uhl, A.: Selective Bitplane Encryption for Secure Transmission of Image Data in Mobile Environments. In: 5th Nordic Signal Processing Symposium on board, Hurtigruten, Norway, October 4-7 (2002)

[QN97] Qiao, L., Nahrstedt, K.: A New Algorithm for MPEG Video Encryption. In: Proceedings of the 1st International Conference on Imaging Science, Systems and Technology (CISST 1997), Las Vegas, Nevada, pp. 21–29 (July 1997)

[QNT97] Qiao, L., Nahrstedt, K., Tam, I.: Is MPEG Encryption by Using Random List Instead of Zigzag. In: IEEE International Symposium on Consumer Electronics (December 1997)

[SSS] Seidel, T., Socek, D., Sramka, M.: Cryptanalysis of Video Encryption Algorithms. In: 3rd Central European Conference on Cryptology, TATRACRYPT 2003, Bratislava, Slovak Republic (2003)

[T96] Tang, L.: Methods for Encrypting and Decrypting MPEG Video Data Efficiently. In: Proceedings of the 4th ACM International Multimedia Conference, Boston, MA, November 18-22, pp. 219–230 (1996)

[W91] Wallace, G.K.: The JPEG still picture compression standard. IEEE Transactions Consumer Electronics (1991)

[YVU07] Yekkala, A.K., Veni Madhavan, C.E., Udupa, N.: DCT properties as handle for image compression and cryptanalysis. In: Advances in Pattern Recognition, Proceedings of the Sixth International Conference, Indian Statistical Institute, Kolkata, India, January 2 - 4 (2007)

[YU07] Yekkala, A., Veni Madhavan, C.E.: Bit Plane Encoding and Encryption. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 103–110. Springer, Heidelberg (2007)