International Journal of Advanced Science and Technology
Vol. 33, August, 2011

Destructive and Constructive Aspects of Wavelet Tree Shuffling Cryptotosystems for Image Transmission

Samuel Assegie, Paul Salama
Brian King
Purdue School of Engineering & Tech.
Indiana University - Purdue University Indianapolis
briking@iupui.edu

Abstract

The demand for online multimedia services is ever increasing. There is a need for bandwidth reduction (compression), while at the same time for increased security. Traditional cryptographic algorithms/systems for data security are often not fast enough to process the vast amounts of data generated by the multimedia applications to meet the real-time constraints. Selective encryption is a scheme that is often used for multimedia content protection. It involves encrypting only a portion of the data to reduce computational complexity(the amount of data to encrypt) while preserving a sufficient level of security. A tool that is sometime used in a selective encryption scheme is Wavelet Tree Shuffling. We will show that Wavelet Tree shuffling when used with our mechanisms can construct a secure selective encryption scheme (constructive aspects), whereas when it is used as a single mechanism within a selective encryption scheme, the result can be insecure (destructive aspects).

1 Introduction

Internet multimedia applications has become extremely popular. Valuable multimedia content such as digital images and video, are vulnerable to unauthorized access while in storage as well as during a transmission over a network. Streaming of secure images/real-time video in the presence of constraints, such as bandwidth, delay, computational complexity and channel reliability is one of the most challenging problems. For example, a 512×512 color image at 24 bits/pixel would require 6.3Mbits. While the bandwidth issue can be resolved using compression, securing multimedia data still remains a big challenge, especially in light of the diversity of devices (in terms of resource availability) that will transmit and receive the content.

Traditional image and video content protection schemes are fully layered, the whole content is first compressed, and then the compressed stream is encrypted using a standard cryptographic technique (such as TDES, AES, ...)[16]. However, the requirement for high transmission rate with limited bandwidth makes this traditional technique inadequate. In the fully layered scheme compression and encryption are two different(disjoint) processes. The multimedia content is processed as a classical text assuming that all the bits in the plaintext are equally important. But with constrained resources (in real-time networking,

high definition delivery, low power, low memory and computational capability) this scheme is inefficient. Thus techniques for securing multimedia data requiring less complexity and less adverse effect on the compression without compromising the security of the data is required. One such technique is to use selective encryption [3, 18]. Selective encryption is an encryption technique based on combining the encryption and compression process, that will reduce computational complexity as well as bandwidth utilization, by encrypting only the "essential parts of the image".

This work focuses on the use of *Wavelet Tree Shuffling*, which we will abbreviate as *shuffling*, as a selective encryption tool. First we discuss its use as the sole encryption mechanism, as suggested by Kwon, et. al. [4], in Section 4.2, we discuss the weakness of this approach. Then we discuss its use as a secondary encryption mechanism within a much broader mechanism as suggested by [13], and how it enhances the primary security encryption technique.

2 Wavelet based compression techniques

Over the past several years, the wavelet transform has gained widespread acceptance in image compression research. Since there is no need to divide the image into macro blocks (no need to block the input image), wavelet based coding at higher compression avoids blocking artifacts. Wavelet transform can decompose a signal in to different subbands. There are many compression techniques that use the wavelet transform, including JPEG-2000, EZW [14] and SPIHT [12]. We now briefly discuss EZW and SPIHT.

In 1993, Shapiro [14] presented an algorithm for entropy encoding called *Embedded Zerotree Wavelet (EZW) algorithm.* After applying the wavelet transform, the coefficients can be represented using trees because of the subsampling that is performed in the transform. A coefficient in a low subband can be thought of as having four descendants in the next higher subband. The four descendants each have four descendants in the next higher subband and we see a quad-tree structure emerges and every root has four leafs. A zerotree is a quad-tree of which all nodes are equal to or smaller than the root. The zero tree structure is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T. The idea is to define a tree of zero symbols that start at a root that is also zero and label as end-of-block. Many insignificant coefficients at higher subbands (finer resolution) can be discarded since the tree grows as powers of four.

The EZW algorithm encodes the obtained tree structure. The resulting output is such that the bits that are generated in order of importance, yielding a fully embedded code. The main advantage of this encoding technique is the encoder can terminate the encoding at any point, thereby allowing one to achieve a target bit rate (i.e. rate scalable). Similarly, the decoder can also stop decoding at any point resulting an image that would have been produced at the rate of the truncated bit stream. Since EZW generate bits in order of importance, those bits that affect the perceived quality of the decompressed image/video most can be placed at the beginning of the data stream; since the entire stream depends on those bits they can be a good candidate for selective encryption.

In 1996, Said and Pearlman [12] introduced a computationally simple compression technique (algorithm) called *SPIHT (Set Partitioning In Hierarchical Trees)* that is based on

the wavelet transform. SPIHT uses set partitioning and significance testing on hierarchical structures of transformed images to extend/improve on the work of Shapiro [14]. SPIHT is also a good candidate to be used in a selective encryption scheme.

3 Some Prior Work on Selective Encryption

Selective encryption has been suggested and adopted as a basic idea for encryption of digital images and videos, aiming to achieve a better trade off between the encryption load and the security level. Selective encryption is a method of selectively concealing portions of a compressed multimedia bitstream while leaving the remaining portions of the stream unchanged.

There are a number of selective encryption techniques. Here we briefly discuss only a few schemes. For more details and a more thorough discussion we suggest the reader look at [18, 6].

In 2002, Podesser, Schmidt and Uhl [10] applied the following technique. They proposed a selective bitplane encryption using AES. They conducted a series of experiments on 8-bit grayscale images, and observed the following: (1) encrypting only the MSB is not secure; a replacement attack is possible, (2) encrypting the first two MSBs gives hard visual degradation, and (3) encrypting three bitplanes gives very hard visual degradation.

Zeng and Lei [20] proposed a selective encryption scheme in the frequency domain (wavelet domain). The general scheme consists of selective scrambling of coefficients by using different primitives (selective bit scrambling, block shuffling, and/or rotation). The input video frames are transformed using wavelet transform and each subband represents selected spatial frequency information of the input video frame. The authors propose two ways to scramble the coefficients. In their first suggestion, they observed that some bits of the transform coefficients have high entropy and can thus be encrypted without greatly affecting compressibility. In their second suggestion, the authors observed that shuffling the arrangement of coefficients in a transform coefficient map can provide effective security without destroying compressibility, as long as the shuffling does not destroy the low-entropy aspects of the map relied upon by the bitstream coder. To increase security, the authors suggested block shuffling. Each subband is divided into a number of blocks of equal size (the size of the block can vary for different subbands) and within each subband, blocks of coefficients will be shuffled according to a shuffling table generated using a key.

Kwon, Lee, Kim, Jin, and Ko [4] described a scheme which involves shuffling of spatial orientation trees(SOT) to secure multimedia data. The authors mentioned the deficiency of traditional block shuffling technique and proposed *wavelet tree shuffling* as an alternate security mechanism as part of the security architecture for multimedia digital rights management. The authors proposed a 4-level wavelet transform. According to Shapiro's[14] algorithm, this will result in 13 sub-bands, and the wavelet coefficients are grouped according to wavelet trees.

In 2005, Salama and King [13] proposed a joint encryption-compression technique (Selective Encryption) for securing multimedia data based on EZW. Their approach is selectively encrypting those bits for which the entire bit stream depends. Through a serious of experiments/simulations, the authors found that encrypting the leading 256 bits of a 512×512 image will provide sufficient security. In their scheme, first the image will be transformed using discrete wavelet transform, apply EZW, and then entropy encoded before it is encrypted

using the proposed Selective Encryption technique. The authors developed a security analysis of the proposed joint compression-encryption technique, and demonstrated an attack called *Database Attack*. In this attack, an adversary(unintended receiver) can intercept the encrypted signal and attempt to replace the encrypted portion of the data stream by another portion that he/she would generate. For the attack to be successful, the interceptor would need a selective database (small enough for computations to be feasible and large enough to include all possible target images), that contains at least one of the possible images that can be transmitted. The attacker then performs a brute force attack by encoding all images in the database and comparing the unprotected part of the stream with the corresponding part of the compressed images from the database. If there is a match, the attacker can replace one stream with another. We discuss this attack and scheme in Section 5.

In 2006, Wu and Mao [8] proposed a shuffling technique as part of their selective encryption architecture. The authors use the MPEG-4 fine granularity scalability (FGS) functionality provided by the MPEG-4 streaming video profile [5] to illustrate their concept and approach. A video is first encoded into two layers, a base layer that provides a basic quality level at a low bit rate and an enhancement layer that provides successive refinement. The enhancement layer is encoded bitplane by bitplane from the most significant bitplane to the least significant one to achieve fine granularity scalability. The authors propose an intra bit plan shuffling on each bit plane of n-bits according to a set of cryptographically secure shuffle tables and using a run-EOP approach. In addition to bit-plane shuffling, the authors also proposed randomly flipping the sign bit s_i of each coefficient according to a pseudo-random bit b_i from a one-time pad, i.e., the sign remains the same when $b_i = 0$ and changes when $b_i = 1$.

4 Attacking a wavelet tree shuffling encryption scheme

4.1 A Generic Framework of a Wavelet Tree Shuffling Encryption Scheme

Here we outline the construction of an encryption scheme which is based on the use of permuting the trees which are produced by the wavelet transform. This scheme was suggested by Kwon et. al. in [4].

Suppose the image I is of size $M \times N$. Then I can be represented as

$$I = \begin{bmatrix} m_{0,0} & \cdots & m_{0,N-1} \\ \vdots & \ddots & \vdots \\ m_{M-1,0} & \cdots & m_{M-1,N-1} \end{bmatrix}_{M \times N} \tag{1}$$

where $m_{i,j}$ is the i, j pixel of I.
For a level L of wavelet decomposition the number of SOT's (spatial orientation tree) is

$$T = \frac{M \cdot N}{2^{2L}} \tag{2}$$

If $M = 2^d$ then the maximum level of decomposition will be d. Thus, if an image I of size 512×512 is decomposed using 4 levels of decomposition then there will be 1024 trees. Since there are 1024! permutations of the trees, this would require a key of at least 1024 bits. A symmetric cryptosystem which uses a key of size 1024 should provide security for well over 50 years [1, 2]. However such a scheme does not possess such security.

Encryption:

In this procedure, the coefficient matrix I of size $M \times N$ is shuffled using a permutation (determined by symmetric key K) to form a corresponding image C. This is achieved by applying a permutation (shuffling) to the SOT's that were created during the wavelet transform. More formally let $\mathcal{WT}$ denote the 2D discrete wavelet transform and PERM_K denote the permutation that shuffles the wavelet trees. Then the ciphertext C is generated as follows:

First the wavelet transform is applied to I,

$$\mathcal{WT}(I) = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T)$$

(here $\mathcal{T}_i$ denotes the i^{th} tree produced by the wavelet transform). Then given key K, the trees $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T)$ are then permuted as

$$C = \text{PERM}_K(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T).$$

This process is illustrated in Fig. 1.

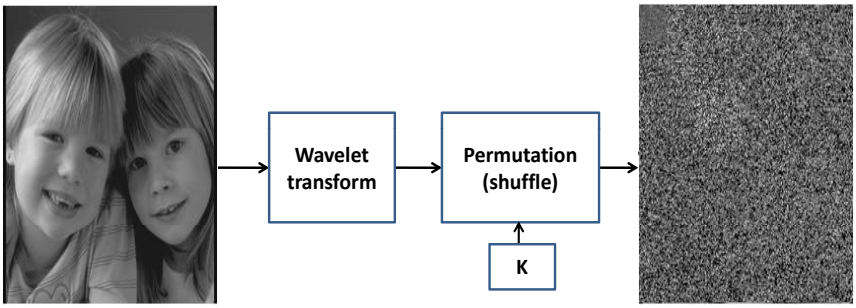

Figure 1. Encrypting by shuffling wavelet trees

Decryption:

Given the ciphertext C, the image I is then reconstructed as follows. First the inverse of the permutation (that was induced by key K) is applied. Thus

$$(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T) = \text{PERM}_K^{-1}(C).$$

Then the $\mathcal{IWT}$ (inverse wavelet transform) is applied. The result is

$$I = \mathcal{IWT}(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T).$$

4.2 The Attack

In this work, we are analyzing shuffling of the wavlet trees (SOT's) as if it were the only mechanism used for encryption [4].

Norcern and Uhl [9] also discussed the insecurity of a system (in terms of compression performance) that was based on randomly permuting wavelet-subbands incorporated in the JPEG2000 or the SPIHT coder proposed by Uehara et. al. in [17]. Their work differs from ours in the sense that they were attacking a scheme that was randomly permuting the coefficients of wavelet subbands, while our attack is on encryption schemes that are shuffling the tree structure created after the wavelet transform.

The basis of our attack will be a chosen plaintext attack, in particular a lunch time attack We will assume that the attacker has temporary possession of the encryption machine and can feed the machine selected plaintext and will receive the corresponding ciphertext. From this lunch time attack the adversary will be able to determine the permutation (encryption) key.

First consider an image of size $\mathbf{M} \times \mathbf{N}$ and suppose it is transformed in to wavelet coefficients using the L-level discrete wavelet transform(DWT). With an L-level decomposition, we have $3L + 1$ frequency bands. In Fig 4, when $L = 4$, the lowest frequency subband is located in the top left(i.e., the LL4 subband, and the highest frequency subband is at the bottom right.(i.e., the HH1 subband). The relation between this frequency bands can be seen as a parent-child relationship [14]. Thus, for an image of size $M \times N$ with L-level of decomposition, in total we will have $\frac{M}{2^L} \cdot \frac{N}{2^L}$ trees. After constructing wavelet trees, a secret key K is used to randomly shuffle the trees.

Clearly if an attacker can guess the size of the image and the number of levels of decomposition, then shuffling of the wavelet trees (SOT's) is vulnerable to the lunch time attack. A simple scenario of a lunch time attack: a legitimate user s away from the desk (computer) without locking their computer/machine, the attackers can use a chosen plaintext attack. A chosen plaintext attack can be launched by any party (adversary) who can access the machine while the user is absent. For the attack to be successful, the adversary needs to have access for the encryption machine. Once the adversary has access to the encryption machine, he/she can choose a series of chosen plaintexts and feed them to the encryption machine as shown in Fig. 5. Thus if I_j denotes a chosen plaintext, and if we denote the wavelet tree shuffling encryption scheme (as illustrated in Fig 1) by $E(I_j, K)$ where K is the key, then the adversary will generate chosen plaintexts

$$E(I_1, K), E(I_2, K), \ldots, E(I_r, K).$$

The adversary can use these to determine the image size and the level of decomposition. Thus we will assume that the adversary knows these parameters.

In [4], Kwon et. al. proposed the shuffling of wavelet trees of a wavelet coefficient which undergoes 4-levels of wavelet decomposition as part of their security architecture for digital rights management. According to Equation 2, for an image of size 512×512 which undergoes 4-levels of wavelet decomposition, there will be 1024 trees. Any shuffling of these trees using a randomized shuffling key(shuffling matrix) should provide the security of 1024 bit key size.

Though Kwon et. al. [4] were using a level $L = 4$ of decomposition for an image of size 512×512, we will provide an analysis/experiments based on image size of 256×256. Only a slight modification of our experiments would have to be constructed to attack an image of size 512×512. The following table demonstrates the relationship between the level of decomposition and the number of trees.

Assume for the moment that the level of decomposition was $L = 5$. In Fig 2, we provide the original chosen plaintext image. This image was selected, an image consisting of a series of integers from 1 to 64, where each integer is in white placed within a small back box,

Table 1. Relationship between no. of decomp. and no. of trees for an image of size 256×256

No. of decomp.	No. of trees
3	1024
4	256
5	64
6	8
7	4

in order to clearly determine the permutation key and the levels of decomposition. Using the encryption machine, and applying the shuffle to this plaintext we get the ciphertext in Fig 3, for a level of decomposition $L = 3$, we get the ciphertext in Fig 4, for a level of decomposition $L = 4$, and we get the ciphertext in Fig 5, for a level of decomposition $L = 5$. Observe the clarity of the image Fig 5, the digits are clearly visible thus revealing that we have determined the level of decomposition. Once the adversary possesses Fig 5, they know the level of decomposition as we well as the permutation, hence they know the key, and so they have broken the cipher.

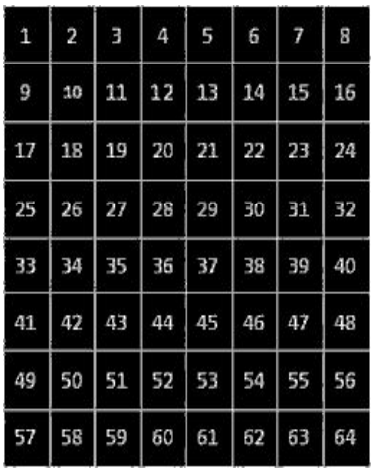

Figure 2. *Original Image*

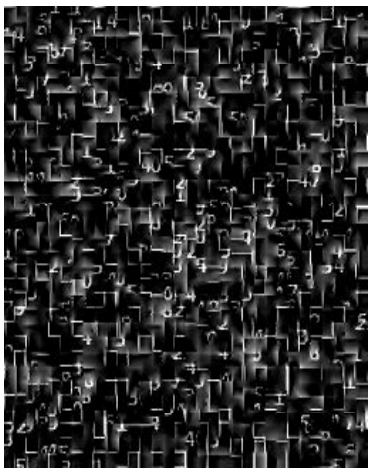

Figure 3. *Shuffled Image where* $L = 3$

So due to the nature of wavelet trees if an adversary would have guessed the correct size of the image and the number of level of decomposition, he/she can get a ciphertext that will leak information about the randomized key used to shuffle the wavelet trees (SOT's). The adversary needs to encrypt only $\log_2(V)$ times for each carefully chosen plaintext images to find the key, where V is $max(M, N)$ for an image of size $M \times N$. This is because the maximum number of possible levels of decomposition is bounded by the logarithm of the dimensions as observed in Section 4.1. An image of size 256×256 has at most 8 levels of decomposition, as illustrated in the above figures, by the choosing an appropriate plaintext, and viewing the potential ciphertexts from the vantage of different levels of wavelet decomposition (such as $L = 3$, $L = 4$, $L = 5$), the correct level and key can be determined. That is, the image is a carefully chosen image, determined by the adversary and the resulting

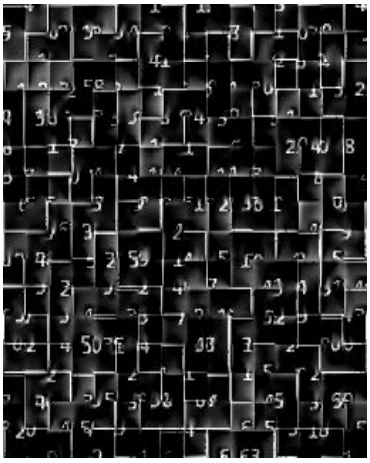

Figure 4. Shuffled Image where $L = 4$

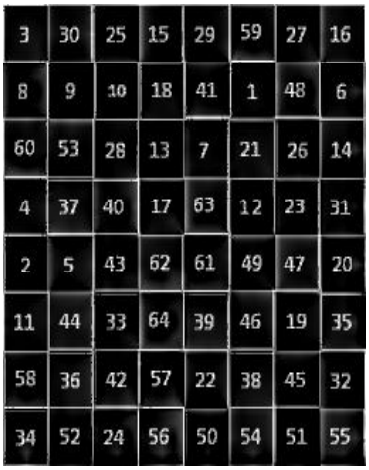

Figure 5. *Shuffled Image where* $L = 5$

ciphertext (Fig. 3, 4, 5) will be analyzed, and if the adversary can easily deduce how the wavelet trees are shuffled then they have found the key. For example, for an image of size 256×256 which has gone through 4 levels of wavelet decomposition we need to run the encryption machine for the chosen plaintext at most 8 times and as it can be easily be seen how the trees are shuffled by looking at Fig 5. The series of chosen plaintext (image) used by the adversary will be limited since the possible number of decomposition are limited (small numbers) to distort the image visually. For example, for an image of size 512×512 a wavelet transform of the image with 8 level of decomposition will have only 4 wavelet trees and the key size to shuffle these trees also be 4, and will not distort the image. Thus, the adversary can easily guess the key used to shuffle the wavelet trees.

5 Applying Shuffling as a secondary encryption technique-constructive aspects

5.1 Why use shuffling

The concept of perfect secrecy implies that $Prob(P|C) = Prob(P)$ for all possible P, where P represents the plaintext message (an image in our case) and C represents the ciphertext (the encryption of P). All selective encryption techniques clearly do not satisfy perfect secrecy. In the case of selective encryption, we can reduce the security requirement to a requirement that the encryption method is such that the ciphertext C provides "very little" information concerning the plaintext message P. A question is how to quantify an encryption method's security given that some information is revealed and what constitutes "very little". The assurance of the security of a selective encryption application is usually reduced to the argument that it is difficult to reconstruct the image using those bits of the ciphertext that have been left unmodified . A traditional tool that a cryptographer uses to prove the security of an encryption method is to assume that an adversary has access to some "oracle". The adversarial model is such that the adversary can utilize this oracle to determine secret information by making queries to the oracle. The adversary is limited to certain types of queries to the oracle, for example the adversary cannot ask the oracle to

decrypt the targeted ciphertext. Given an appropriate query, the oracle will respond with the correct answer. The goal of the adversary is to use the queries to obtain some knowledge of the encryption system and/or secret key.

In practice there usually is not an oracle for an encryption system, at least not a known oracle to all. However, some encryption systems are weak in the sense they could provide a "hidden oracle", perhaps one discovered by the adversary. In many implemented cryptosystems, an oracale has been discovered. This is why the importance of analyzing an encryption system under the existence of an oracle is important. The goal of the cryptographer is to prove that the encryption method is so secure that it can withstand this adversary/oracle game. That is, if the encryption method is used within the adversary/oracle game, then despite all valid queries made to the oracle, the adversary is unable to obtain information about the key and the plaintext. While usually this oracle is a hypothetical entity, in the case of selective encryption every adversary possesses a natural oracle. That is, consider an image P that is selectively encrypted producing ciphertext C. Assume that the adversary has been able to capture C then they are able to construct their own oracle. Suppose that the adversary possesses several images $L_1, L_2, \ldots, L_b$ and they wish to ask the oracle whether L_i is the image encrypted. The adversary compresses L_i in the appropriate manner (according to the compression method used in the selective encryption) and examines the corresponding bit stream of the compressed L_i. Since P has been selectively encrypted there are bits of P that are contained in ciphertext C that have not been "scrambled in any fashion", i.e. have not been modified. Thus by examining the bit stream of C, restricting the adversary's examination to those bits that were sent unmodified, an oracle exists. If the stream of a L_i and C do not match then the adversary knows that L_i was not P, if the streams do match then, with some probability (not always necessarily high), L_i is the same as P. This obversation was made in [13] that a natural oracle exists. The Section 5.2 will discuss a method that will effectively diminish this oracle.

A rate-scalable compression based selective encryption can provide confidentiality. In the preliminary selective encryption scheme proposed by Salama and King [13] (see Fig. 6), In their preliminary selection encryption scheme, which used rate scalable compression scheme EZW, Salama and King [13]. In this preliminary version the first step would be to apply the wavelet transform (WT) to the image. Then successive approximation (SA) and arithmetic coding (AC) would be applied. At this time, the image is now represented as a data dependent sequence. the selective encryption (SE) is then applied to this sequence, since the sequence is data dependent only the leading bits need to be encrypted. In this work [13] we found that only 128 bits needed to be encrypted. This process is illustrated in Figure 6.

In the database attack, an adversary intercepts the ciphertext and attempts to determine the plaintext. portion of the data stream by a "guessed stream" that the adversary generates. This strategy will work if the adversary has somehow managed to obtain additional information concerning the content of the transmitted image, that is they are able to"guess the correct image" with a probability better than guessing the most likely choice, without seeing anything. In fact, if the adversary cannot "guess the correct image" but can guess a low-resolution version of the image then the attack will be successful. Let D represent a database of the possible plaintext images or coarse versions of the images. For this attack to be successful, the adversary would need a database D that contains at least one of the possible images or a coarse version of it, with a suitably high probability. The adversary then performs a brute force attack by encoding all the images in the database and comparing the

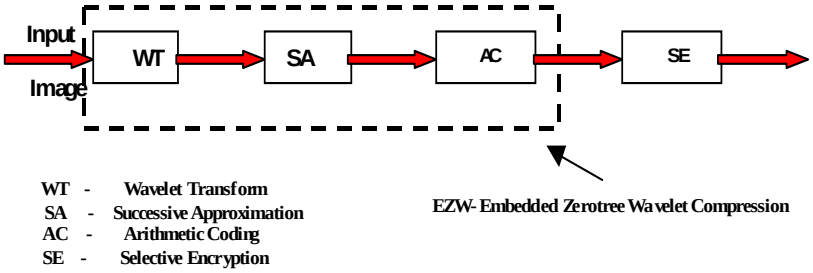

Figure 6. Preliminary Selective Encryption Scheme[13]

unprotected part of the transmitted EZW compressed stream with the corresponding part of the compressed images from the database. If there is a match then the attacker can replace the encrypted part of the transmitted stream with the initial part of the database-stream. In fact, the adversary could have pre-computed and stored each image in the database in compressed form. This attack is successful provided that the database size is small enough for it to be "feasible" to perform the needed computations . Also, the database needs to be large enough to include all possible target images. The amount of computation is infeasible but that in-itself is again naïve. One could utilize an additional encryption mechanism that would frustrate a database attack by making the number of needed computations infeasible.

5.2 A solution to the Database attack

To frustrate the database attack and to remove this natural oracle is to randomly shuffle or permute the spatial orientation trees (SOTs). that are created by the wavelet transform, prior to encoding them (denoted by the S-Box in Figure 7). This has the effect of obscuring some of the features in an image at the cost of a small loss in compression performance. Furthermore, the permutation satisfies:

- The cleartext of the transmitted stream will have been modified, thus for an adversary to overcome the use of this permutation they will have to compress and attempt to use all possible permutation transformations within their database (consequently we have most likely have succeeded in making the number of computations needed to perform the database attack infeasible); and
- By inserting the permutation prior to the successive approximation (SA) stage of EZW we have successfully integrated some aspects of encryption with compression.

In our final selection encryption scheme, which uses similar steps as before, we use wavelet transform, successive approximation, arithmetic codling and the selective encryption scheme, with one addition a permutation/substitution box (S-Box), which is applied to the SOT trees prior to the successive approximation phase. See Figure 7.

Note that if $\mathcal{T}$ represents the number of SOT trees and $|D|$ the size of the database then the amount of work that an adversary will need to perform to conduct the Database attack to the our final encryption scheme will be equivalent to permuting each of compressed members of D, so the work would require $2^T \cdot |D|$ compressions.

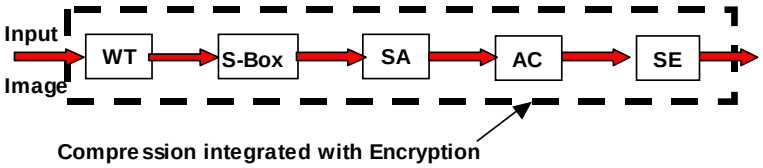

Figure 7. Selective Encryption Scheme Secure against the Database Attack[13]

6 Conclusion

We have discussed the wavelet tree shuffling encryption scheme [4]. We have showed that the shuffling of the wavelet trees, when used as a sole security mechanism, is insecure against a chosen plaintext attack. However despite its weaknesses, shuffling of wavelet trees can add security when other cryptographic primitives are used. As we discussed the scheme in [13], which shows that the use of shuffling of the wavelet transform trees can enhance the security of a selective encryption scheme to the Database attack.

References

[1] "Recommendation for Key Management", *Special Publication 800-57 Part 1, NIST*, 03/2007.

[2] Keylength - Cryptographic Key Length Recommendation. http://www.keylength.com.

[3] B. Furht and D. Kirovski, Eds. *Multimedia Security Handbook*, CRC Press, 2004.

[4] G. Kwon, T. Lee, K. Kim, J.. Jin, and S. Ko, "Multimedia digital right management using selective scrambling for mobile handset", *Computational Intelligence and Security*, LNAI, vol. 3802, pp. 1098-1103, 2005.

[5] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard", *IEEE Trans. on Circuits & Systems for Video Technology*, vol.11, no.3, pp. 301-317, March 2001.

[6] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, B. and J. Quisquater, "Overview on selective encryption of image and video: challenges and perspectives", *EURASIP J. Inf. Security*, pp. 1-18, 2008.

[7] M. Naor and M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks. Proceedings of the 22nd Annual Symposium on Theory of Computing, ACM, 1990

[8] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption", *IEEE Trans. Image Processing*, 15 (7) (2006), pp. 2061-2075.

[9] R. Norcen and A. Uhl, "Encryption of wavelet-coded imagery using random permutations", in *Proceedings of the IEEE International Conference on Image Processing (ICIPŠ04)*, (Singapore), IEEE Signal Processing Society, Oct. 2004.

[10] M. Podesser, H. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments", In *Proceedings of the 5th Nordic Signal Processing Symposium (NORSIG 02)*, 2002.

[11] A. Pommer and A. Uhl, "Selective encryption of wavelet-packet encoded image data: efficiency and security", *Multimedia Systems*, vol. 9, no. 3, pp. 279-287, 2003.

[12] A. Said and W.A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees", In *IEEE Trans. Circuits and Systems for Video Technology*, volume 6(3), pp. 243-250, June 1996.

[13] P. Salama and B. King, "Efficient secure image transmission: compression integrated with encryption", *Proc. SPIE.*, Vol. SPIE-5681, pp. 47-58, 2005

[14] M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients", *IEEE Trans. Signal Processing*, Vol. 41, pp. 3445-3462, 1993.

[15] A. Shamir, "How to share a secret", *Comm of ACM*, vol. 22, no. 11, pp. 612-613, November 1979.

[16] D. Stinson, *Cryptography: theory and practice*, 2^{nd} ed., CRC Press, 2002.

[17] T. Uehara, R. Safavi-Naini, and P. Ogunbona, "Securing wavelet compression with random permutations", in *Proceedings of the 2000 IEEE Pacific Rim Conference on Multimedia*, Sydney, Dec. 2000, pp. 332-335, IEEE Signal Processing Society.

[18] A. Uhl and A. Pommer, *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*. Boston: Springer, 2005.

[19] C. Valens, As appeared in http://pagesperso-orange.fr/polyvalens/clemens/ezw/ezw.html, 1999

[20] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video", *IEEE Transactions onMultimedia*, vol. 5, no. 1, pp. 118-129, 2003.