Fuzzy Authentication Algorithm with Applications to Error Localization and Correction of Images

OBAID UR-REHMAN and NATASA ZIVIC
Chair for Data Communications Systems
University of Siegen
Hoelderlinstrasse 3, 57076 Siegen,
GERMANY
{obaid.ur-rehman, natasa.zivic}@uni-siegen.de

Abstract: - Images are normally protected using standard messages authentication codes to protect them against tampering and forgeries. One problem with this approach is that when such images are transmitted over a noisy medium, even a single bit error might render the image as un-authentic to the receiver. In this paper, a noise tolerant data authentication algorithm is proposed. The proposed algorithm can perform authentication in the presence of minor errors but at the same time identify forgeries in the data. This algorithm is then extended by demonstrating its applications in image authentication. The extended algorithm is called as fuzzy authentication algorithm. It has the ability to localize errors in an image as well as correct the localized errors using error correcting codes. The proposed algorithm rejects only the potentially erroneous / unauthentic parts of the image and correct or authenticate the remaining parts if the number of errors is below a certain threshold. This reduces the need for retransmission of the complete image and only a few parts might be retransmitted if the application demands. This property is especially useful in real-time communications. It is better to obtain a part of the authentic image, rather than having no image at all. A security analysis of the proposed algorithm is given, and simulation results are presented to demonstrate its error localization and correction capabilities.

Keywords: - Fuzzy Authentication; Image Authentication; Reliability Values; Soft Authentication; Content Based Authentication; Noise Tolerant Authentication

1 Introduction

Modern communication systems are typically composed of compression, channel coding and security modules. While compression reduces the size of transmitted data, channel coding aims to deliver the data reliably to the receiver by using additional parity data. The security module provides authenticity of the origin, protection against eavesdropping and forgery attempts. The goal of compression is to eliminate redundancy, whereas the channel coding and security modules add redundancy to the compressed data in order to achieve their respective goals. Thus the solutions provided by these modules can often be conflicting. An improved coordination and information sharing between these building blocks can achieve the desired individual goals of the modules in a better manner.

In order to provide the above mentioned security features, such as message authenticity, proof of origin, integrity and protection against forgeries, cryptographic methods are used. Standard Message Authentication Code (MAC) is used to ensure the integrity and authenticity of a message. MAC is computed on a message using a shared secret key between the sender and the receiver and is appended to the message. This also helps in proving the authenticity of origin of a message. In standard applications of MAC, hard authentication is used, which ensures that even a single bit modification to the message is detected. Some applications including multimedia transmission like image, voice and streaming audio, are generally noise tolerant [5] and have real-time requirements. In such applications, it might be meaningful to retain the received object (e.g., an image), without the need for a retransmission, even if the hard authentication fails and a modest number of errors exist. For this category of applications, soft authentication algorithms have been proposed in the literature, such as Approximate Message Authentication Code (AMAC), Image Message Authentication Code (IMAC) and Noise Tolerant Message Authentication Code (NTMAC) [5-7]. Some soft, as well as hard, authentication algorithms tailored for multimedia transmission have also been proposed [8-12]. Some parallel work on image retrieval based on image content instead of the image pixel data includes [13-14]. Such algorithms are called content based image retrieval algorithms. Image encryption based on image content and region of interest selection has been shown recently in [15]. However, most of these algorithms work with little or

in no co-ordination with the other modules of a communication system. Thus the received image could be rejected due to a single (additional) error beyond the allowed limit. In this paper an algorithm for data authentication in the presence of noise is proposed. This algorithm is named as Soft Input Decryption (SID) and its proposed variant as Threshold based Soft Input Decryption (TSID).

In a standard image transmission system, the image is divided into non-overlapping (square) blocks, and for each block a discrete cosine transform (DCT) is calculated, which is followed by a quantization and entropy decoder. Among all of the DCT components, the first coefficient plays the most important role and is called the DC coefficient, while the rest are called AC coefficients. It is well known that despite of discarding a certain number of AC coefficients, the image can be reconstructed at the receiver, using the Inverse DCT, without much loss in the quality. The redundancies which are assumed to be discarded are inter-pixel or psychological redundancies and can be discarded without any significant detectable visual effects. A soft authentication algorithm based on the coordination between the channel coding and message authentication modules is proposed in this paper. The proposed algorithm works on compressed images using the Discrete Cosine Transform (DCT) and uses the TSID algorithm as well as the NTMAC [7] algorithm for the soft authentication and error localization as well as error correction. A certain number of errors below a predefined threshold are corrected, while another permissible number of errors are tolerated by the proposed authentication algorithm. DCT is used for the basic feature extraction and image compression. The proposed algorithm is based on the NTMAC, with additional error localization, correction and soft authentication properties.

This paper is organized as follows; In Section 2, the Soft Input Decryption Algorithm is described. In Section 2, the Soft Input Decryption Algorithm with Threshold is discussed. In Section 4, the DCT is briefly reviewed followed by the description of the proposed algorithm. The analysis of the proposed algorithm is given in Section 5. Simulation results are presented in Section 6, followed by the conclusion in Section 7.

2 Soft Input Decryption (SID) Algorithm

SID algorithm is the basis for Joint Channel Coding and Cryptography concept [1], [2]. This concept develops further the idea from [3], [4] on using the soft output (Log Likelihood Ratios or L-values) of the SISO (Soft Input Soft Output) channel decoding in order to try and correct the input of cryptographic mechanisms and therefore improving the results of decryption.

The algorithm of Soft Input Decryption (presented in Fig. 1) deals with blocks of data where each block contains a message (as payload) to which its Cryptographic Check Value (CCV) is concatenated as a security redundancy. At the transmitter, the CCV is generated by passing the message through a cryptographic check function and using a secret key K shared with the receiver. Message M and its CCV are thereafter encoded by the channel encoder and transmitted over a noisy channel. After performing channel decoding at the receiver, the (SISO) channel decoder outputs the estimated message M′, the estimated CCV′ and the *L*-values of all the bits of M′ and CCV′. These three elements are the input to Soft Input Decryption process which can be briefly described as follows: At first, CCV″ = *CCV*(M″ = M′) is calculated by applying the shared secret key K. If CCV″ = CCV′, the verification result is successful / true and M' is accepted as correct.

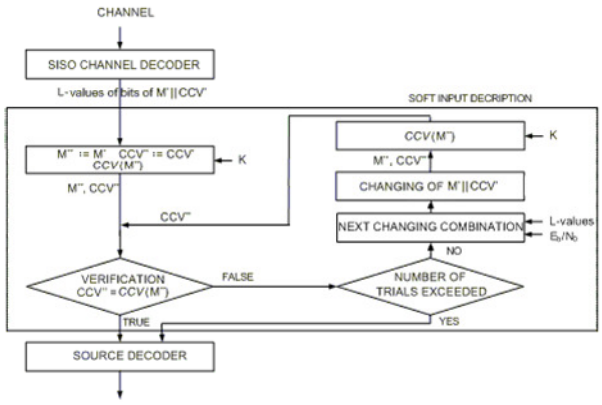

Fig. 1 Soft Input Decryption Algorithm

If the verification result is un-successful / false, the *L*-values of the channel decoder are analyzed. Some bits from M′ and CCV′ with the lowest |*L*|-values are flipped / inverted. There are different strategies for choosing the bits to be inverted, but the simpliest (and also efficient) way is to sort the |*L*|-values in increasing order and then to pick those bits from M′ and CCV′ which correspond to sorted |*L*|-values.

After the bit inversion, which results in M″ and CCV″, the verification process compares CCV″ and *CCV*(M″) for equality. If the verification result is

„false" again, another bit or combination of bits is inverted (corresponding to the binary interpretation of *N*-bit counter incremented by 1). This iterative process continues till either the verification result is „true" or a threshold number of iterations have been performed.

The idea of inversion of the least reliable bits originated from Chase decoding algorithms [3] in 1972, which were the generalization of the GMD (Generalized Minimum Distance) algorithms from 1966 [4] and improved channel decoding. These algorithms have been applied to a binary (n, k) linear block code and are referenced as LRP (Least Reliability Positions) algorithms. The novelty of SID is the inversion of bits iteratively for data protected by cryptographic redundancy with feedback between the decryptor and the channel decoder, since it gives the possibility to check whether the inversion iteration was successful or not (through the process of verification).

It may happen that a pair of M″ and CCV″ generated by the bit inversion passes the verification although the message M″ is not equal to the original message M. The probability of such an event is very small, which will also be taken in consideration in the next Section.

In simulations, both message and CCV have the length of 160 bits. CCV has been calculated using RIPEMD160 with a key *K*. Simulations have been performed using a Convolutional encoder of code rate r = 1/2 and constraint length m = 2 as the simplest one, but used very often in theory and practice. BPSK modulation and an Additive White Gaussian Noise (AWGN) channel are used together with a SISO decoder based on the Maximum A-Posteriori (MAP) algorithm. The MAP decoder was programmed in such a way, that it supports the output of L-values.

The presentation of the results of the simulations (Fig. 2) clearly shows obtained benefits and potentials of SID algorithm. In order to measure the improvement, a parameter named the Cryptographic Check Error Rate (CCER) is defined as follows:

$$CCER = \frac{number\ \ of\ \ incorrect\ \ CCVs}{number\ \ of\ \ received\ \ CCVs}, \quad (1)$$

where an incorrect cryptographic check value is each CCV which didn't pass the verification.

Fig. 2 shows the improvement of CCER with Soft Input Decryption for the cases that up to 8 of the lowest |*L*|-values (graph b) or up to the 16 lowest |*L*|-values are used (graph c). Example: CCER of 10^{-3} at E_b/N_0 ~ 6.2 dB without Soft Input Decryption is the same as for E_b/N_0 ~ 4.2 dB with Soft Input Decryption, or decreased from 10^{-1} to 10^{-3} at E_b/N_0 ~ 4.2 dB. Using up to 16 of the lowest |*L*|-values a coding gain of 2.33 dB can be reached, and CCER > 10^{-1} without Soft Input Decryption can be reduced down to CCER < 10^{-4} at E_b/N_0 ~ 4 dB.

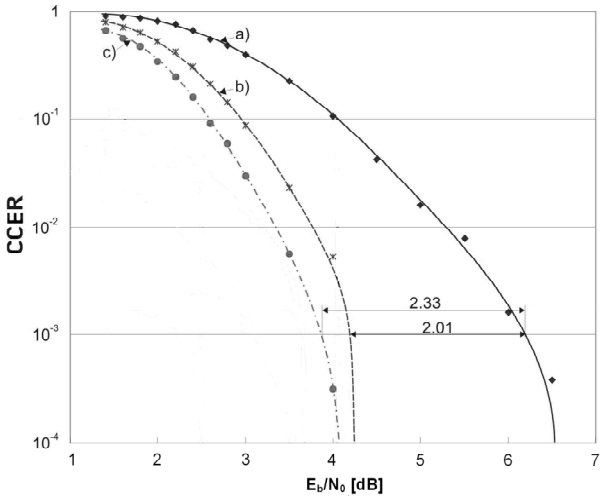

Fig. 2 Results of Soft Input Decryption using up to 8 (b) and 16 (c) lowest |L|-values compared to results without Soft Input Decryption (a)

At this point, a few words need to be said regarding the *L*-values and the type of channel decoder which can be used together with the SID. The channel decoder is assumed to be SISO (Soft Input Soft Output). SISO is a concept of channel decoding, which was originally used in iterative and Turbo coding, because the (soft) output is fed-back internally. Soft output of the channel decoder is used here as the soft input for the cryptographic verification process (called Soft Input Verification). Soft output of the channel decoder is usually expressed as L-value of each output bit u′,

$$L(u') = \ln \frac{P(u=1)}{P(u=0)}, \quad (2)$$

L(u′) represents the reliability of the decision made by the channel decoder, i.e., if the sent bit u was a 1 or 0. The sign of the *L*-value shows the hard output of bit u′ (1 or 0) and the magnitude, i.e., |*L*|, is used as reliability value of the hard decision. Example: if *L* is positive, the hard output is 1, otherwise it is 0. The higher the |*L*|, the more reliable is the hard decision and vice versa: a lower |*L*| means a less reliable decision. When the *L*-Value is equal to 0, the probability of the correctness of the decision is 0.5.

It is obvious that SID method greatly depends on the "quality" of *L*-values, which means that "better information" on bit reliability will give better results

after SID. The decoder which produces *L*-values need not to be exclusively of SISO type, but each *L*-value provided for each particular bit must, in some way, represent its reliability. Such a requirement disqualifies, for example, Viterbi decoding since then, the produced array of *L*-values gives the probabilities of the paths through trellis and not bit reliabilities. From the other side, there is MAP decoder (used in simulations) which internally calculates probabilities of each bit value taking into account the values of all other bits before and after a particular bit. Having all these probabilities already obtained by MAP, using (2) it is easy to get *L*-values which are suitable for the use within SID.

MAP is also a SISO decoder, which, besides the bit array from the output of line decoder (i.e. demodulator), as input can take the soft information on the probabilities of demodulated bits as well. It is important because then MAP has the opportunity to be used as a part of different more efficient (and more complex) decoding schemes with feedback. One of them is Turbo-decoding scheme where two MAP blocks are coupled over "crossed feedbacks", working together in an iterative process. In the above mentioned simulations, the "pure MAP" without feedback is used, which assumes that all the input probabilities of demodulated bits (soft input) have been preset on the value of 0.5.

Having in mind the way on which MAP internally calculates bit probabilities, one can conclude that for better results, the length of input bit array (in our case message M concatenated with its CCV) shouldn't be too small, since each bit probability (and the decision based on it) will be calculated using more neighboring bits. On the other side, the more distant bits have smaller influence on the calculations and also can add numerical noise, so it is expected that there is an optimal length of MAP input array (which is equal to the length of output array). The obtained results are also affected by the quality of *L*-values from the MAP decoder, which as explained, depends on the lengths of message and CCV. After all, the resume is that message and CCV together should have an optimal length (or close to optimal). In such a case where the messages are too long (or a continuous data stream is to be transmitted), they should be fragmented and for each fragment the CCV should be calculated, so that all together have more or less optimal length.

3 Soft Input Decryption Using Threshold

Although it seems that the SID method, especially in the scenario with a feedback, has exploited the whole soft information available from the channel decoder but further improvements of coding gain are still possible. Since the decryption of CCV is very sensitive to errors, the verification process need not to be so "strict" as it used to be. This gives more space for the SID method, which is now able to choose L-values more precisely by setting the verification threshold. Setting the threshold increases the probability of the false verification (collision), but even then it is extremely small.

The efficiency of SID method depends on the obvious factors such as the lengths of the message and the CCV, the number of bits chosen for inversion and the E_b/N_0 ratio. Some results of many simulations for different values of these parameters are presented in the previous Section as well as in [1]. Besides the mentioned parameters, the "quality" of *L*-values produced by channel decoder plays an important role, which indirectly affects the overall efficiency of the process involving SID. Naturally, SID works better with a smaller as compared to a bigger portion of data which is to be corrected.

One way to decrease the data length used by SID is to exclude the bits of cryptographic check value from correction within SID. Namely, if the length of the message is *m* and the length of cryptographic check value is *n*, SID now considers only *m* instead of *m+n* elements. SID picks *N* lowest |*L*|-values and inverts corresponding bits (only from decoded message M') in the iterative process of verification.

The above proposed enhancement of SID method is possible since the cryptographic check value satisfies the so called avalanche criterion which assumes that wrong decoded (i.e., reconstructed) CCV has in average 50% of the bits wrong. This means that if only one bit from the decoded message M′ is erroneous, around *n/2* bits in CCV″ will be erroneous as well. In other words, when a decoded message M′ is incorrect, CCV(M′) must have many wrong bits, much more (i.e. significantly more) than the decoded CCV'. So in the case that CCV″ contains "only a few" incorrect bits, i.e., when the Hamming distance (HD) between CCV′ and CCV″, i.e., HD(CCV', CCV") is "small enough", it is obvious that M′ is correct (M′ equals original M) and that the difference between CCV′ and CCV″ exists only because of the errors in CCV′. Hence, during the verification process within "Thresholded SID" (TSID), there is no need to check if all the bits from CCV″ are equal to the corresponding bits in CCV′, but the criterion for successful verification would be that *d*=HD(CCV',

CCV") is less than a threshold d_{max} which is to be determined.

In order to determine the appropriate value for decision threshold d_{max}, the statistical distribution of HD between CCV′ and CCV″ has to be found. For given BER after decoder (P_e) and the length of the message m, the probability distribution function (*pdf*) over different values of d can be expressed in the form of total probability sum:

$$pdf(d) = P_{M'correct} \cdot pdf_1(d) + P_{M'incorrect} \cdot pdf_2(d), \quad (3)$$

where $P_{M'correct}$ and $P_{M' incorrect}$ are the probabilities that decoded message M′ does not contain or contains errors respectively:

$$P_{M'correct} = (1 - P_e)^m, \quad (4)$$

$$P_{M'incorrect} = 1 - (1 - P_e)^m, \quad (5)$$

and $pdf_1(d)$ and $pdf_2(d)$ are conditional probability distribution functions of the HD after M′ is correct or incorrect.

In the case of successful verification, Hamming distance d = HD(CCV′, CCV″) is expected to have a small value, smaller than the decision threshold d_{max} (which is to be found). Also, CCV″ will be equal to the original CCV (because M′ is equal to original M), so d will be equal to the number of errors in CCV′ only, and d_{max} should be greater than the possibly largest number of errors in CCV′. Since after channel decoder the remaining errors (if exist) are uniformly distributed only over the CCV′ (with the length of n bits), the number of errors in CCV′ has a binomial distribution B(n,P_e), i.e.,

$$pdf_1(d) = \binom{n}{d} P_e^d \cdot (1 - P_e)^{n-d}, \quad 0 \le d \le n, \quad (6)$$

with mean value $n \cdot P_e$ and standard deviation $\sigma^2 = nP_e(1\text{-}P_e)$.

When the verification is unsuccessful, HD(CCV′, CCV″) is "large" (above the decision threshold d_{max}) as a consequence of the characteristics of cryptographic check value. Namely, when the message is wrongly decoded (M′ is incorrect, i.e., M′ contains one or more errors) the number of errors in CCV″ is expected to be n/2 due to the avalanche criterion. In this case, CCV″ can take any of 2^n values (with equal probability).

Definition 1: Bit arrays A and B have length of n elements. Each element a_i and b_i is independent from other bits and has equal probability of taking the values 0 and 1, i.e.,

$$a_i = \begin{cases} 0, & p_0 = 1/2 \\ 1, & p_1 = 1/2 \end{cases} \qquad b_i = \begin{cases} 0, & p_0 = 1/2 \\ 1, & p_1 = 1/2 \end{cases} \quad (7)$$

Definition 2: Y_k and N_k are subsets of set S={1,2,...,n} where: Y_k has D elements, N_k has n-D elements, $Y_k \cup N_k = S$ and $Y_k \cap N_k = \varnothing$.

Lema 1: The Hamming distance d between arrays A and B:

$$d = HD(A,B) = \sum_{i=1}^{n} |a_i - b_i| \quad (8)$$

has the Binomial distribution B(n, p=1/2) i.e.,

$$pdf(d) = \binom{n}{d} \cdot \frac{1}{2^n}, \quad 0 \le d \le n, \quad (9)$$

Proof: The probability that d takes a concrete value D is:

$$P\{d = D\} = P\left\{\sum_{i=1}^{n} |a_i - b_i| = D\right\} = \\ = \sum_{Y_k, N_k} P\{a_i = b_i, \forall i \in Y_k\} \cdot P\{a_i \ne b_i, \forall i \in N_k\} \quad (10)$$

where the summation is applied on each subset pair (Y_k,N_k). The number of those pairs is: $k_{\max} = \binom{n}{D}$, so we have,

$$P\{d = D\} = \\ = \binom{n}{D} \cdot \left[\prod_{i=1}^{D} \left[P\{a_i = 1 \wedge b_i = 1\} + P\{a_i = 0 \wedge b_i = 0\} \right] + \right. \\ \left. + \prod_{i=1}^{n-D} \left[P\{a_i = 1 \wedge b_i = 0\} + P\{a_i = 0 \wedge b_i = 1\} \right] \right] = \\ = \binom{n}{D} \cdot \left[\left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \right)^D + \left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} \right)^{n-D} \right] = \\ = \binom{n}{D} \cdot \frac{1}{2^n}, \quad (11)$$

which proves the *Lema*.

In the case of an unsuccessful verification, CCV′ and CCV″ can be considered as independent bit arrays A and B according to *Definition 1*, and HD(CCV′, CCV″) will have the probability distribution function as shown in *Lema 1*:

$$pdf_2(d) = \binom{n}{d} \cdot \frac{1}{2^n}, \quad 0 \le d \le n, \qquad (12)$$

Equation (12) can also be explained in simpler way. Namely, when the message is not verified, the expected value of HD(CCV′, CCV″) is equal to the expected value of HD between CCV′ and any other fixed array of bits of the same length. If for simplicity, we choose an array of bits X = 00...0, the HD(CCV′, X) will also have Binomial distribution B(*n*,*p*), where *p*=1/2 since every bit in CCV′ is expected to be 0 or 1 with equal probability, so the *pdf* of HD(CCV′, CCV″) can be written as in (12).

By combining equations (4), (5), (6) and (12) in (3), for the parameter values *m*=160, *n*=160 and P_e=0.01, the probability distribution of *d* = HD(CCV′, CCV″) will have the shape as shown in Fig.3. Two regions are clearly distinguished: the left one for the case of successfully decoded message (i.e., M = M′) and the second − when the decoded message M′ is wrong (i.e., M ≠ M′). It is obvious that the probability distribution over *d* is zero for a great range of values between these regions, which means that the decision threshold d_{max} in the process of verification might take any value from this middle area.

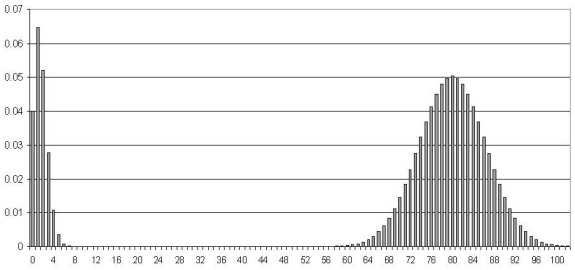

Fig. 3 Probability distribution of d=HD(CCV′, CCV″)

The rate of acceptance of decoded and by flipping corrected messages within verification process will be greater if the d_{max} is set to a greater value. The lower limit of the threshold (d_{max_low}) can be chosen in relation to wanted acceptance rate of message, i.e., regarding the predefined probability of message rejecting. In columns 3, 5 and 7 of Table 1 (see Appendix) the values of d_{max_low} are shown for different lengths of cryptographic check value (*n* = 160, 128, 64) and message (*m* = 160, 192, 256) so that *m*+*n* = 320 and for different BERs (depending on E_b/N_0). The criteria for choice of $d_{max,low}$ was that the probability of message rejecting (when M′ is correct) is less than 10^{-k}, i.e.,

$$\sum_{d=d_{\max\_low}+1}^{n} P_{M'corect} \cdot pdf_1(d) < 10^{-k}, \qquad (12)$$

where $P_{M'correct}$ and $pdf_1(d)$ are defined in (8) and (10), and the condition for message to be accepted as correct is,

$$d = HD(CCV', CCV'') \le d_{\max}. \qquad (13)$$

By setting the parameter *k* to an appropriate value, wanted minimal acceptance rate can be achieved and the matching values of d_{max_low} can be calculated from (12) and (13). The values of d_{max_low} obtained in this way are shown in Table I, where each cell contains four different values: for *k* = 4, 6, 10 and 15 respectively.

On one hand, greater d_{max} and higher acceptance rate of messages means speeding up the verification process, since the expected number of bit-flipping iterations leading to successful verification is smaller. Greater d_{max}, on the other hand, will increase the probability of false verification − the event when the decryptor wrongly decides that a decoded message M′ (or the corrected message M″ after a number of bit-flipping iterations) is correct. This happens when CCV″, which acts as a random variable (since calculated from wrong decoded message M′ and the secret key K), satisfies the condition (7). The probability of false verification becomes significant when the value of decision threshold d_{max} is getting closer to the region on the right side in Fig.3. Similarly as by choosing the lower limit, the upper limit of the threshold ($d_{max,high}$) can be found with regard to the probability of false verification which can be tolerated. This probability can be also defined by the use of parameter *k*, while $d_{max,high}$ will be the maximal integer that satisfies the following condition:

$$\sum_{d=0}^{d_{\max\_high}} P_{M'incorect} \cdot pdf_2(d) < 10^{-k}, \qquad (14)$$

(in (3) and (5) are the definitions of $P_{M'incorrect}$ and $pdf_2(d)$).

Columns 4, 6 and 8 of "Table 1" contain values of $d_{max,high}$ calculated from (8) for *k*=4, 5, 10 and 20 respectively. There is a lot of values within range [$d_{max,low}$+1, $d_{max,high}$] which could be taken as the threshold, for different values of parameters E_b/N_0, P_e, *m* and *n*.

Both SID and TSID methods have been simulated with the message and its CCV / HMAC tag, both of length of 160 bits. HMAC tag has been calculated

using RIPEMID160 hash function. Simulations have been performed using a Convolutional encoder of code rate r (= ½) and a constraint length m = 2, BPSK modulation, AWGN channel and SISO decoding using MAP algorithm.

The results of simulations are expressed through Cryptographic Check Error Rate (CCER), already defined by (1), as the ratio between the number of incorrect CCVs after (T)SID and the whole number of simulations for given set of parameters. In both methods 16 bits with smallest $|L|$-values were being flipped, i.e., maximally 2^{16} trials of soft correction (bit flipping) had been performed in each simulation. The value of decision threshold within TSID had been set to 20% of the CCV length d_{max}= 32.

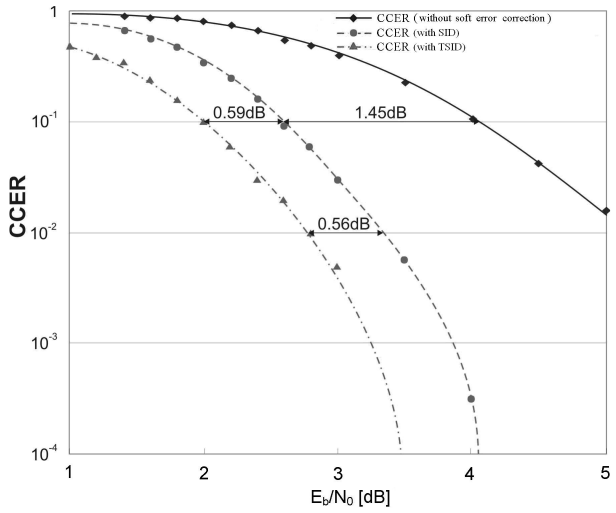

Fig. 4 Achieved coding gains of SID and SID with threshold (TSID)

The results are presented in Fig. 4, showing the achieved coding gain in comparison to standard 1/2 convolutional coding. Using original SID method, more than 1.4 dB of gain is achieved, while TSID obtains additional 0.5-0.6 dB.

4 Authentication Of Images Using TSID And Noise Tolerant MACs (NTMACs) Based On The Discrete Cosine Transform

4.1 Introduction

In this Section the application of TSID algorithm in combination with the Noise Tolerant Message Authentication Code (NTMAC) is investigated in image authentication. For images, the basic features are authenticated rather than authenticating the image itself. For this purpose, the Discrete Cosine Transform is used to extract the block by block features and authenticate the image block-wise. This is also beneficial for the TSID algorithm, which works well over small blocks of data as compared to big ones.

4.2 Discrete Cosine Transform in Image Processing

Discrete Cosine Transform is one of the most widely used techniques in image processing for basic feature extraction and compression. Its application in image processing was pioneered in [16]. Due to its better reconstruction capability, DCT is more suitable to images than other relevant transforms, such as Discrete Fourier Transform (DFT). DCT, like other transforms, tries to eliminate the correlation from image data. After de-correlation, each transform coefficient can be encoded independently without devitalizing the compression efficiency. Many well known image and video compression standards like JPEG and MPEG-1/2/4/H.26x, are based on 2-D DCT.

The Discrete cosine transform of a 2-D vector is defined as follows [16-18]:

$$X(l,k)=\alpha(l)\alpha(k)\frac{2}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x(m,n)\cos\left[\frac{\pi(2m+1)}{2N}\right]\cos\left[\frac{\pi(2n+1)}{2N}\right]$$

where,

$$\alpha(i)=\begin{cases}1/\sqrt{2}\ , & i=0\\ 1\ , & 1\le i\le N-1\end{cases} \quad (15)$$

It is clear from (15) that the first coefficient (DC) represents the average intensity of the corresponding block and contains most of the energy and perceptual information. Also the inverse of 2-D DCT for l,k = 0, 1, ..., N-1 is defined as follows,

$$x(l,k)=\alpha(l)\alpha(k)\frac{2}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}X(m,n)\cos\left[\frac{\pi(2m+1)}{2N}\right]\cos\left[\frac{\pi(2n+1)}{2N}\right] \quad (16)$$

Besides the general characteristics of DCT which are defined for every Fourier-like transform, other properties of DCT like de-correlation, energy compactness, symmetry and separability make it a convenient tool for image processing purposes.

4.3 Introduction and Definitions

The algorithm introduced in this paper is based on the DCT transform. It protects the transmitted DCT components of an image by NTMAC and performs soft authentication on received (noisy) images. The algorithm is able to localize errors in the images and to correct a certain number of them if they are below a

certain threshold. The following definition is used in the description of the proposed algorithm.

Definition 3: Let H´ be the received n-bit MAC for a transmitted message M. Let M´ be the received message, H´´ be the MAC recalculated at the receiver and let d be a small non-negative integer ($d << n/2$); then (M´, H´) is said to be d-soft-verified if HD(H´, H´´) ≤ d, where HD is the Hamming Distance between H´ and H´´.

For the sake of simplicity, let's assume the image to be transmitted is N × N pixels. Let m be the block size such that m|N, where both N and m are integers. The sender divides the image into m × m-pixel disjoint blocks (typically m is equal to 8). This is followed by the calculation of DCT for each block.

4.4 Image Authenticating and Correcting Weighted Noise Tolerant MAC (IAC-WNTMAC)

IAC-WNTMAC achieves error localization using the concept of weights. IAC-WNTMAC is based on NTMAC [7] and identifies the locations of potential erroneous blocks with a high probability.

The NTMAC algorithm [7] works by splitting a message / image into smaller components. A MAC is calculated for each of the smaller components and truncated to obtain a sub-MAC. The sub-MACs corresponding to these message components are concatenated to form the NTMAC. The IAC-WNTMAC tag calculation can either operate row-wise or column-wise on the image blocks. It is assumed here that the tag calculation is done row-wise. A DCT matrix is obtained for each block in the source image. Thus there are as many DCT matrices as the number of blocks in the source image. NTMAC is calculated based on the DC components of the DCT matrices taken row-wise. There are N/m such DCT matrices in each row and therefore N/m DC components are used to get one NTMAC (against a row).

The same step is repeated for all the rows, giving N/m NTMACs. This process is also repeated for the first minor diagonal after DC coefficient (called as first minor diagonal for the sake of simplicity) of the DCT matrices, giving another set of N/m NTMACs. This produces a total of 2(N/m) MACs, i.e., N/m + N/m. The usage of NTMAC improves the error localization, whereas the usage of the NTMAC for the minor diagonal increases the quality of reconstructed image at the receiver as explained next. All of these N/m + N/m NTMAC tags are appended together to obtain IAC-WNTMAC tag for transmission.

The image I´ and its IAC-WNTMAC´ tag are received over a noisy channel. The receiver recalculates IAC-WNTMAC tag on I´ to get IAC-WNTMAC´´. Now the received IAC-WNTMAC´ tag is compared with the recalculated IAC-WNTMAC´´ tag. This is done by comparing the corresponding sub-MACs. If the sub-MACs are d-soft-verified according to the definition given above, then the DC component is accepted as authentic and the message block corresponding to the DC component is declared as authentic. Otherwise, the block is marked as un-authentic / suspicious. All the blocks marked as un-authentic / suspicious will be tried for error correction using Chase like iterative error correction algorithm based on the bit reliabilities calculated using the MAP decoder at the receiver. This iterative error correction is repeated for the first minor diagonal as well, so that they can be reconstructed to get a better quality of the reconstructed image. However, the first minor diagonal has a lower weight than the DC component. Lower weight means that the threshold for the maximum number of iterations used for the recovery of the first minor diagonal is smaller than the threshold used for the recovery of the DC component, i.e., a variation of the EC-WNTMAC [10] is used. If T_{iterDC} is the iteration threshold used for the error correction of DC components and T_{itrFMD} is the same used for the first minor diagonal, then the total number of iterations are given by,

$$T_{itr} = T_{itrDC} + T_{itrFMD} \quad (17)$$

The pseudo-code of the IAC-WNTMAC tag generation and verification algorithms is given below for an N × N image. It can be easily extended to the general case where the image is not square. Also here weights are assigned based on the DC and the first minor diagonal elements, which can be easily extended to other minor diagonals. The notation DC is self explanatory, whereas MD represents the Minor Diagonal of the DCT matrix.

Algorithm: IAC-WNTMAC Tag Generation Algorithm

Inputs:

- Source Image (I)
- Image width / height in pixels (N)
- Block length (m)

Algorithm:

blocks = splitImageIntoBlocks(I, N, m)

for i = 1 to N/m

for j = 1 to N/m
DCT = $blocks_{i,j}$
DC = $DCT_{1,1}$
$subMAC_{DCj}$ = calcSubMAC(DC)
$subMAC_{MDj}$ = calcSubMAC($DCT_{1,2}$ || $DCT_{2,1}$)
end
$subMAC_{DCi} = subMAC_{DC1} || \dots subMAC_{DCN/m}$
$subMAC_{MDi} = subMAC_{MD1} || \dots subMAC_{MDN/m}$
end

$NTMAC_{DC} = subMAC_{DC1} || subMAC_{DC2} || \dots || subMAC_{DCN/m}$
$NTMAC_{MD} = subMAC_{MD1} || subMAC_{MD2} || \dots || subMAC_{MDN/m}$

Output:

$NTMAC_{DC} || NTMAC_{MD}$

Pseudo code for tag verification at the receiver:

Algorithm: IAC-WNTMAC Tag Verification at the Receiver

Inputs:

- Received Image (I´)
- Received NTMAC´
- Image width / height (N)
- Block length (m)
- Block LLRs (blockLLRs)

Algorithm:

I´´ = decompressImage(I´)
$subMAC´_{DC}$ = makeDCSubMACs(NTMAC´)
$subMAC´_{MD}$ = makeMDSubMACs(NTMAC´)
dc_LLRs = blockLLRsToDCLLRs(blockLLRs)
md_LLRs = blockLLRsToMDLLRs(blockLLRs)
blocks = makeBlocks(I´´, W, H, m)

for i=1 to N/m
for j=1 to N/m
DCT = $blocks_{i,j}$
DC = $DCT_{1,1}$
$subMAC_{DCi,j}$ = calcSubMAC(DC)
$subMAC_{MDi,j}$ = calcSubMAC($DCT_{1,2}$ || $DCT_{2,1}$)
if(HD($subMAC´_{DCi,j}$, $subMAC_{DCi,j}$) ≤ d)
performErrorCorrection(DC, $DC\_LLRs_{i,j}$)
end
if(HD($subMAC´_{MDi,j}$, $subMAC_{MDi,j}$) ≤ d)
performErrorCorrection($DCT_{1,2}||DCT_{2,1}$, $MD\_LLRs_{i,j}$)
end
end
end

Output: authenticMessageBlocks

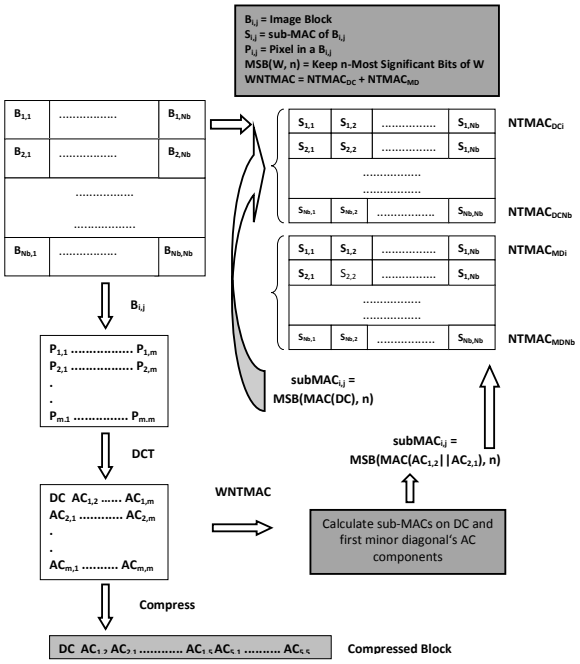

Fig. 5 IAC-WNTMAC Transmitter

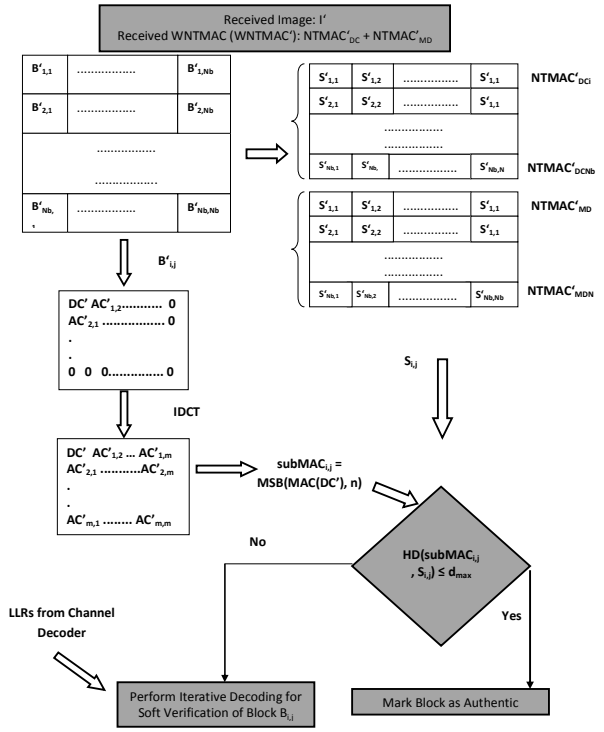

Fig. 6 IAC-WNTMAC Receiver

5 Analysis of the proposed Algorithm

IAC-WNTMAC algorithm is a variant of WNTMAC given in [10]. Therefore the analysis of the IAC-WNTMAC algorithm is based on WNTMAC. It is assumed that an ideal n-bit MAC (n ≥ 256) algorithm is used for each row and each column of concatenated selected DCT elements of the blocks of an image. The total number of concatenated DC coefficients is $(N/m)^2$ of k-bit each. Let the bit error rate of the channel be denoted as BER.

5.1 Performance Study

d-soft-verification is successful, if the difference between the received MAC and the recalculated one is not greater than the threshold value. Two types of errors can exist: "false rejection" (correct image is discarded) and "false acceptance" (wrong image is accepted).

The probability of a false rejection of the whole image (P_{FR}) depends on the policy of the application and the nature of the image blocks. False rejection is not as bad as false acceptance, which causes communication overhead and reduces the efficiency and therefore the security. Therefore the probability of false acceptance will be discussed.

False acceptance happens when there are error(s) in the received image I´, but the received and recalculated tag pair is d-soft-verified. The probability of false acceptance on the block level is:

$$P_{FA}(Block) = \left[1-(1-BER)^k\right]\left(\sum_{i=0}^{d}\binom{n}{i}BER^i(1-BER)^{n-i}\right)^2 \quad (18)$$

5.2 Security Considerations

The most important integrity threat against data is message substitution and forgery. It refers to any attempt for adding, removing and manipulating objects into data in order to fool the receiver to accept the wrong message. This threat in image data can lead to image tampering.

The algorithms introduced in this paper are based on the standard ideal MAC, so the generic attacks on MACs are considered as potential threats. As in the given approaches, the algorithms may tolerate a modest number of error(s) as a consequence of soft verification. The security strength is decreased generally compared to hard authentication MAC schemes by allowing near collisions. This drawback is compensated in both approaches. In the first one each DCT element is supported by two MACs instead of one MAC and the attacker has to forge DCT elements in such a way that both row and column MACs become d-soft-verified. In the second algorithm, the attacker has even more difficult task, the forgery attack requires forgery on protected DC and AC coefficients so that IAC-WNTMAC authentication on both selected DC and AC coefficients becomes successful.

A common approach for approximating the required complexity (data/time) for forgery attack on MACs is given by a "birthday paradox" which is based on finding collisions. In case of soft authentication, the attacker attempts to launch a near-collision attack [19]. Near-collision refers to any message pair whose MACs differ only in few bits from each other. By extending the birthday paradox to the introduced soft verification scheme with threshold d, it is expected to have a near collision (with at most d-bit differences) with the data complexity (C) of,

$$C = \sqrt{\frac{2^{2n}}{\sum_{i=0}^{d}\binom{2n}{i}}} \quad (19)$$

In the presented algorithms, the minimum MAC length used to protect row and columns of DC coefficients is 256 bits. The threshold value is set in such a way that the probability of events like false acceptance and false rejection remains significantly low. There are other experimental methods to find a convenient safe threshold zone through image processing techniques. It can be easily concluded that the security strength compensated by double length of the MAC is much larger than the required security strength of a standard MAC, for low threshold values. The security of the second approach is even higher due to secret partitioning.

6 Simulation Results

The simulation results are presented for the proposed algorithm using image transmission over AWGN channel with BPSK modulation. The results are given in the presence of rate 1/3 Turbo Codes. The extrinsic Log Likelihood Ratios (LLRs) produced by the decoder for Convolutional Turbo Codes (CTC) are used for bit reliabilities values.

The source image is a grayscale image of 128 × 128 pixels (each pixel of 8 bits). The image is split into 8 × 8 pixel non-overlapping blocks, giving a total of 16 × 16 blocks. DCT for each of these blocks is calculated and the DC components of the DCT sub-matrices are protected using the corresponding MACs

as explained earlier. Each element in the DCT matrices requires 2 octets. If the original image is transmitted using the standard MAC based protection, then either the whole image will be authentic or non-authentic and so it will either be accepted or discarded. HMAC-SHA-256 is used as the MAC in the simulations, thus in total 128 × 128 × 8 bits of image data plus 256 bits of MAC needs to be transmitted, which is equal to 131328 bits in total.

Using IAC-WNTMAC, the number of data bits transmitted is calculated as follows. Each WNTMAC tag is 256 bits long. Thus, 256 bits for each of the DC components in the 16 rows as well as another 256 bits for the first minor diagonals for each row are used. Thus 256 × 16 × 2 = 8192 bits of WNTMAC tags are transmitted along the 61440 bits of the data. This is equal to 53% of the whole data transmitted in the standard MAC tag based image transmission.

Each figure shown in the following sub-sections is divided into five constituent sub-images. First, the source image is shown followed by the received image. In the next sub-image, the suspicious block positions (identified through the proposed algorithms) are highlighted in white followed by these suspicious blocks highlighted in the received image. Finally, the resultant image is shown, which is obtained by applying the proposed error correction algorithm over the erroneous image based on the localized errors.

6.1 Simulation Results for IAC-WNTMAC

Images protected using IAC-WNTMACs have better error localization capabilities and so they can be reconstructed in a better manner as compared to the previous algorithm. The simulation results are presented in Fig. 7.

Fig. 7 IAC-WNTMAC at SNR 2.5 with Turbo Codes of rate-1/3

6.2 Image Error Rate (IER)

IER for both the algorithms is shown in Fig 8 at different values of E_b/N_0. The curves represent the IER in the presence of a standard MAC tag based protection scheme and then in the presence of IAC-WNTMAC. Fig. 4 shows that IAC-WNTMAC achieves a coding gain of 1.2 dB at IER of 10^{-4}. Its performance is due to dual error protection and recovery using weighted NTMAC.

7 Conclusion

An algorithm for approximate data authentication (SID) is presented first. This is extended further to TSID which more efficiently perform authentication by iteratively considering only the data part in authentication and doing a threshold number of comparisons till the match criteria is satisfied. Both the algorithms fall into the category of fuzzy authentication algorithms. The application of TSID together with the NTMAC using DCT is demonstrated in image authentication. Thus an algorithm for soft authentication, error localization and correction of images is presented. Soft authentication is performed using the standard MAC together with the threshold value. The main property of the algorithms is its ability of error localization and correction without compromise of security, which is shown in the analysis. Simulation results showing the high error recovery as well as relatively accurate error localization validate the theoretical analysis. In future it would be interesting to extend the proposed work by combining it with artificial intelligence based cooperative learning strategies, e.g., the one proposed in [20]. It is expected to get better content based image retrieval results using such approaches.

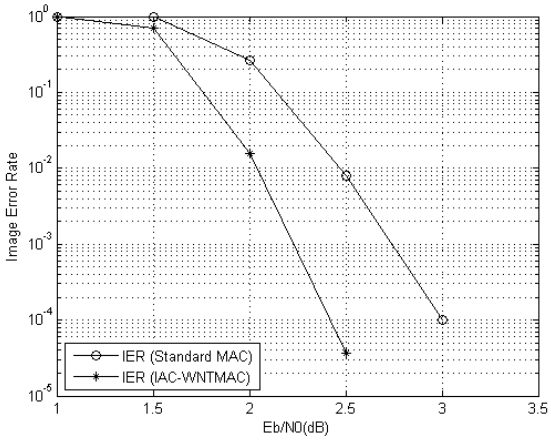

Fig. 8 IER over AWGN channel with BPSK modulation using Turbo codes of rate-1/3

References:

[1] N. Zivic, Joint Channel Coding and Cryptography, Shaker Verlag, Aachen (2008).

[2] C. Ruland, N. Zivic, Soft Input Decryption, 4th Turbocode Conference, 6th Source and Channel Code Conference, VDE/IEEE, Munich, Apr 2006

[3] D. Chase, A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, IEEE Trans. Inform. Theory, IT-18, pp. 170-182, Jan 1972

[4] G. D. Jr. Forney, Generalized Minimum Distance Decoding, IEEE Trans. Inform. Theory, IT-12, pp. 125-131, Apr 1966

[5] R. Gravemen, L. Xie, and G. R. Arce, Approximate image message authentication codes, in Proc. 4th Annu. Symp. Advanced Telecommunications and Information Distribution Research Program, College Park, MD, 2000.

[6] R. Graveman and K. Fu, Approximate message authentication codes, in Proc. 3rd Annual Fed lab Symp. Advanced Telecommunications / Information Distribution, vol. 1, College Park, MD, Feb 1999.

[7] C. Boncelet, The NTMAC for authentication of noisy messages, IEEE Trans. Info. Forensics and Security, vol. 1, no. 1, pp. 35-42, Mar 2006.

[8] D. Onien, R. Safavi-Naini, P. Nickolas and Y. Desmedt, Unconditionally secure approximate message authentication, in Proc. The Second International Workshop on Coding and Cryptology, Springer, 2009.

[9] R. Ge, G. R. Arce and G. D. Crescenzo, Approximate message authentication codes for N-ary alphabets, IEEE Transactions on Information Forensics and Security, vol. 1, no. 1, 2006.

[10] O. Ur-Rehman, N. Zivic, S. Amir Hossein A. E. Tabatabaei, C. Ruland, Error Correcting and Weighted Noise Tolerant Message Authentication Codes, 5th Int. Conference on Signal Processing and Communication Systems (ICSPCS) / IEEE Conference, Hawaii, USA, Dec 2011.

[11] N. Zivic, M. Flanagan, On Joint Cryptographic Verification and Channel Decoding via the Maximum Likelihood Criterion, IEEE Comm. Letters, vol. 6, no. 5, pp. 717-719, May 2012.

[12] O. Ur-Rehman, A. Tabatabaei, N. Zivic, C. Ruland, Soft Authentication and Correction of Images, 9th International ITG Conference on Systems, Communications and Coding (SCC 2013), Jan 2013, Munich, Germany.

[13] L. Zhang, L. Xi, B. Zhou, Image Retrieval Method Based on Entropy and Fractal Coding, WSEAS Transaction on Systems, issue 4, vol. 7, Apr 2008.

[14] C. Aviles-Cruz, A. Ferreyra-Ramirez, J. J. Ocampo-Hidalgo, I. Vazquez-Alvarez, Structured-Image Retrieval invariant to rotation, scaling and translation, WSEAS Transaction on Systems, issue 8, vol. 8, Aug 2009.

[15] N. Doukas, Low Color-Depth Image Encryption Scheme for use in COTS Smartphones, WSEAS Transaction on Systems, issue 9, vol. 11, Sep 2012.

[16] A. Watson, Image compression using Discret Cosine Transform, Mathematical Journal, vol.1, no. 4, pp. 81-88, 1994.

[17] N. Ahmed, T. Natarajan, and K. R. Rao, Discrete Cosine Transform, IEEE Trans. Computers, vol. C-23, pp. 90-93, Jan 1974.

[18] P. Yip, and K. R. Rao, Fast Decimation-in-Time Algorithms for a Family of Discrete Sine and Cosine Transforms, Circuits, Systems and Signal Processing, Vol. 3, pp. 387-408, 1984.

[19] B. Preneel, P. C. van Oorschot, MDx-MAC and building fast MACs, from hash functions, Proc. CRYPTO 1995, LNCS 963, Springer-Verlag, pp. 1-14, 1995.

[20] F. Neri, Cooperative evolutive concept learning: an empirical study, WSEAS Transaction on Information Science and Applications,WSEAS Press (Wisconsin, USA), issue 5, vol. 2, pp. 559-563, May 2005.

Appendix

Table 1
Upper and Lower Limits of the Decision Threshold

E_b/N_0 [dB]	P_e	$\boldsymbol{n = 160}$, m=160		$\boldsymbol{n = 128}$, m=192		$\boldsymbol{n = 64}$, m=256	
		$\boldsymbol{d_{max\_low}}$ (k = 4, 6, 10, 15)	$\boldsymbol{d_{max\_high}}$ (k = 4, 5, 10, 20)	$\boldsymbol{d_{max\_low}}$ (k = 4, 6, 10, 15)	$\boldsymbol{d_{max\_high}}$ (k = 4, 5, 10, 20)	$\boldsymbol{d_{max\_low}}$ (k = 4, 6, 10, 15)	$\boldsymbol{d_{max\_high}}$ (k = 4, 5, 10, 20)
1	0.036	11, 15, 22, 29	56, 52, 40, 23	7, 12, 19, 25	42, 39, 28, 14	0, 6, 12, 17	16, 14, 7, -
1.5	0.0234	10, 13, 19, 25	56, 52, 40, 23	8, 11, 17, 22	42, 39, 28, 14	4, 7, 11, 16	16, 14, 7, -
2	0.0149	8, 11, 16, 21	56, 52 ,40, 23	7, 10, 14, 19	42, 39, 28, 14	4, 7, 10, 14	16, 14, 7, -
2.5	0.00681	6, 8, 12, 16	56, 53, 40, 23	4, 7, 11, 15	43, 40, 28, 14	4, 5, 9, 12	17, 14, 7, -
3	0.00376	5, 7, 10, 14	57, 53, 40, 24	4, 6, 9, 13	43, 40, 29, 14	3, 5, 7, 10	17, 15, 7, -
3.5	0.00142	3, 5, 7, 10	58, 55, 41, 24	3, 5, 7, 10	44, 41, 29, 14	2, 4, 6, 8	18, 15, 8, -
4	0.00037	2, 3, 5, 8	61, 57, 42, 25	2, 3, 5, 7	46, 43, 30, 15	2, 3, 4, 6	19, 16, 8, -
4.5	0.00024	2, 3, 5, 7	61, 57, 43, 25	2, 3, 5, 7	47, 43, 31, 15	2, 2, 4, 6	19, 17, 8, -
5	0.00012	2, 3, 4, 6	63, 58, 43, 26	2, 2, 4, 6	48, 44, 31, 16	1, 2, 4, 5	20, 18, 9, -