Introducing Robustness into Message Authentication

Natasa Zivic

Institute for Data Communications Systems, University of Siegen, Siegen, Germany

Summary
Message Authentication Codes are very sensitive to any change of the message they are appended to. If one or more bits of the message change, Message Authentication Codes change about 50\% of their bits, making the message useless. The successful verification of Message Authentication Codes demands equality of all of bits of the received Message Authentication Code and that one recalculated of the received message. Such a hard condition for the successful verification of messages protected by Message Authentication Codes is not suitable for some applications. The introduction of a softer condition for the successful verification can enable the correction and improve the successful of messages corrupted by transmission over a noisy channel. An algorithm is presented, which introduces robustness into the verification of messages protected by Message Authentication Codes together with a correction of messages corrupted due to the noisy channel. Results show how promising the algorithm is for correction of messages, which have an error rate despite of the use of channel codes.

Key words:
Message Authentication Codes; robustness; soft decision; reliability values; Hamming distance.

1. Introduction

Message Authentication Codes (MACs) [1] apply symmetric cryptographic algorithms, which provide data integrity and authentication of data origin. Data integrity provides the recognition of any modification or manipulation of the message during transmission. Authentication of data origin means the confirmation that the message originates by the sender, who shares the used secret key with the receiver.

MACs are used very often in communication systems, which demand secure message transfer. They are appended to the message in the same way as Cyclic Redundancy Codes (CRCs), and are transferred with the message over a communication channel to the receiver. MACs, as well as CRCs, can recognize if the received message is errorless or erroneous. The main characteristic of MACs, unlike CRCs, is a protection against forgeries. For that reason MACs are constructed in such a way, that any modification of the message results in changing about 50% of bits of a MAC. This effect, known in cryptography as "avalanche effect", implies that every modified message produces an incorrect MAC at the verification. If the verification fails, the message cannot be regarded as authentic and is useless.

A strong condition of the verification of message authentication is a good protection against forgeries. Nevertheless, there is a number of applications, like multimedia or voice transmission, where the digital content is continuously modified and manipulated as a result of compression and conversion. Any of these modifications would be considered as a forgery in case of MAC verification. Therefore, it would be suitable for some applications that the modifications of a single message bit or a few bits do not result in any modification of a MAC. This conclusion leads to the main question and subject of this paper: can the message be accepted as authentic, if only one, two, or a few bits of the message and/or MAC are modified? In other words, is it possible to make message authentication more robust, than it is by using standard Message Authentication Codes?

Several algorithms [2, 3, 4, 5] have been developed in the last decade for the construction of "robust" Message Authentication Codes, which are less sensitive to modifications of messages. These algorithms are designed to calculate such authentication codes, which are more flexible to small changes of message bits. If an erroneous message has been verified and accepted as authentic, it is forwarded without correction to the next entity of the communication system.

An algorithm for correction of messages will be presented, which uses standard Message Authentication Codes, but with a different verification. The received Message Authentication Code and the one recalculated of the received message are compared, as by regular verification, but they will not have to be equal for a successful verification. The verification will be successful also, if one, two, or few bits of both compared Message Authentication Codes are different.

Manuscript received June 5, 2012
Manuscript revised June 20, 2012

This algorithm will be called Threshold based Soft Input Decryption, using as a basis an algorithm of Soft Input Decryption [6]. Both algorithms are iterative and use earlier ideas from [7, 8]. They combine channel decoding and cryptographic verification in such a way, that the message gets corrected using both channel decoding and cryptographic redundancy, i.e. MACs.

The paper is organized as follows: Chapter 2 presents existing algorithms of Approximate Message Authentication Codes and Noise Tolerant Message Authentication Codes with their variants. Chapter 3 describes the algorithm of Soft Input Decryption, as an introduction for Threshold based Soft Input Decryption. The algorithm of Threshold based Soft Input Decryption with an analysis of setting the threshold is presented in Chapter 4. In Chapter 5 a lower and an upper limit of the threshold used in Threshold based Soft Input Decryption are defined and calculated. Chapter 6 shows probabilities of non detection and miscorrection of the presented algorithm. Results of simulations of messages corrected by Threshold based Soft Input Decryption compared to results of correction by Soft Input Decryption and to the standard communication scheme without these correction algorithms are shown in Chapter 7. Security aspects are analyzed in Chapter 8. Chapter 9 is focused on possible application of Threshold based Soft Input Decryption. Finally, conclusion is given in Chapter 10.

2. Approximate Message Authentication Codes and Noise Tolerant Message Authentication Codes

The motivation for new algorithms for message authentication is the sensitivity of standard MACs, whose verification fails even if only one bit is modified.

Approximate Message Authentication Codes (AMACs) [2] were published in 1999. The applications, which are the main motivation for this algorithm, are voice and image communications, where an incidental noise or lossy compression would modify the message and lead to the unsuccessful MAC verification. The design principles of AMACs are:

- AMACs of two messages, that are slightly different, should be the same;
- AMACs of two messages, that have slightly larger difference, should only be slightly different;
- by changing the key, the AMACs should be affected just like the MAC i.e., each bit of the AMAC should change in 50\% of the cases.

Requirements for the construction of AMACs are a one-time shared key (after one usage the key will be discarded), a cryptographically strong pseudo-random generator and a family of pseudo-random permutations. AMACs are constructed by several operations as partitioning of the document into “pages”, permutation, encryption (“whitening”) by using the secret key, generation of pseudo-random bits and calculation of majorities of zero's and one's of the “pages”. An AMAC is changed only if majorities from several “pages” of the document are changed, i.e. if modifications of the document are significant. Vice versa, if modifications of the message are local, i.e. not significant of the whole document, they won't influence the AMAC. In that way, robustness is introduced into message authentication.

A variation of AMACs, Approximate Image Message Authentication Codes (IMACs) [3], was published in 2001. The algorithm is specialized for soft image authentication, whereby IMACs can tolerate small to moderate image compression. IMACs are designed for such modification scenarios as JPEG compression, deliberate image tampering and influence of AWGN.

The algorithm for Noise Tolerant Message Authentication Codes (NTMACs) [4] was published in 2006. It is meant for image and other multimedia communications, similarly as AMACs. It tolerates a few errors (typically between 1 and 32), so it is less strict then the standard MAC, but it does not tolerate as many errors as AMAC. NTMAC behaves between standard MACs and AMACs. Construction of NTMACs achieves division of the message into partitions and partitions into blocks. The bits of the partitions are assigned to different blocks, using secret sub-keys, and standard MACs are calculated for each block and then punctured. Concatenation of punctured MACs forms an NTMAC. NTMACs have the capability to detect erroneous blocks, which are then discarded, claiming the other blocks to be authentic. The security is supplied by using secret keys in two places: for computing MACs and for pseudo-random secret partitioning. The probability of forgeries increases by puncturing of MACs. Interested readers are invited to read more about security aspects of NTMACs in [4].

A variant of NTMACs, which uses CRCs instead of punctured MACs, and enciphering of concatenated CRCs, is published in [5] under the name of CRC-NTMACs. Authors of the algorithm were motivated by the lower probability of non detection of errors in

messages if CRCs are used, compared to MACs of the same length.

Algorithms mentioned above provide robustness by different construction of the message authentication codes. Messages are accepted as authentic, if only small portions are altered. Nevertheless, the accepted message is not corrected by those algorithms, but delivered to the next entity of the communication system (source decoder) with existing errors.

Threshold based Soft Input Decryption (TSID), which will be introduced in Chapter 4, uses standard Message Authentication Codes, but a non standard verification. TSID supports robustness only by the receiver, using an error tolerating authentication. An error tolerating verification means that the received MAC and the MAC recalculated of the received message may differ from each other for a successful authentication, whereby the difference corresponds to the noise of the channel. This algorithm is iterative and tries to correct the received message, and then to authenticate it.

Threshold based Soft Input Decryption uses Soft Input Decryption as a basic method, which will be described briefly in the next Chapter.

3. Soft Input Decryption

Soft Input Decryption (SID) algorithm [6] exploits the soft output and reliability values of SISO (Soft Input Soft Output) channel decoding for the correction of the input of the verification of cryptographic check values (CCVs) (Note: CCV is used as a generalization of MACs).

Knowing that only one or a few erroneous bits of the input of verification will cause the fail of successful verification, and the message becomes useless, the idea is developed to try to correct all of the bits, which are erroneous after, rsp. despite of SISO channel decoding.

The correction is tried through iterations.

SISO is a concept of channel decoding, which was originally used in iterative and turbo coding, because soft output is fed back internally [9]. Soft output of the channel decoder is used here as soft input of the cryptographic verification process. Soft output of the channel decoder is usually expressed as a reliability value LLR (Log Likelihood Ratio), short L-value, of each output bit u' (see Fig. 1).

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

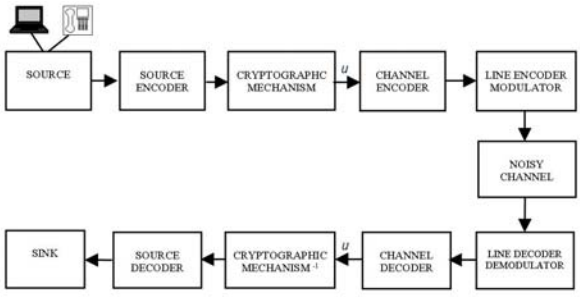

Fig. 1 Communication System.

L(u') expresses the reliability of the decision of the channel decoder, if the sent bit u was 1 or 0. The sign of the LLR value shows the hard output of bit u' (1 or 0) and its absolute value LLR is used as the reliability value of the hard decision. For example: if LLR is positive, the hard output is 1, otherwise 0. As higher LLR, as more reliable is the hard decision, and vice versa. When the LLR value is equal to 0, the probability of the correctness of the decision is 0.5.

Soft Input Decryption uses standard verification of CCVs, as presented in Fig 2. in a simple way. In case that MACs are used as CCVs, the secret key K which is known to both sender and receiver is applied for calculation of CCVs using the cryptographic check function CCF. The verification uses the hard condition of equivalence of received cryptographic check value CCV' and the cryptographic check value CCV", which is recalculated of the received message M'("hard verification").

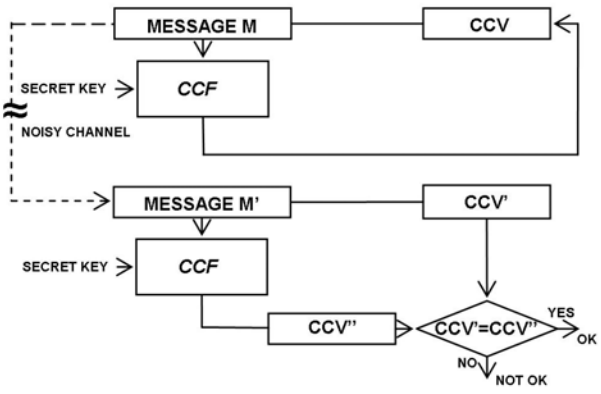

Fig. 2 Hard Verification.

The algorithm of Soft Input Decryption is block-based: each block consists of the received message M' and the received cryptographic check value CCV', as MAC [1] or H-MAC [10], for example.

Soft Input Decryption works as follows:

Input: received message M', received cryptographic check value CCV', LLR values of M' and CCV', maximal number of iterations i_max

Output: corrected message M" with its cryptographic check value CCV"; or FAILURE (optional: additional M' and CCV', so that the next entity gains the information, how the correction was made, according to the principle that no knowledge should be lost on the decoding path);

(i) Reorder the bits of M' and CCV' in increasing sequence of their absolute LLR values;
(ii) Verification: if CCV' = CCF (M'), then go to (v); i = 0;
(iii) If i <= i_max: invert bits of the next combination of lowest absolute LLR values of M' and CCV', resulting in M" and CCV" and go to (iv); else output FAILURE;
(iv) Verification: if CCV" = CCF (M"), go to (v); else increment i, go to (iii);
(v) Output M" and CCV".

The idea of inversion of the least probable bits originates from Chase decoding algorithms [7] in 1972, which were the generalization of the GMD (Generalized Minimum Distance) algorithms from 1966. [8] 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 [11].

SID and its application as feedback information for improvement of channel decoding [12] are the basis of the so called Joint Channel Coding and Cryptography concept published in [13].

Simulations of Soft Input Decryption have been performed using following parameters:
-message of length m=200 and MAC of length of 128 bits
-convolutional encoder (7,5) of a code rate r=1/2
-BPSK modulation
-AWGN channel
-SISO decoder using MAP algorithm [14].

The MAP decoder was programmed in such a way, that it supports the output of L-values. For each point of the resulting graphs, 10.000 simulations have been performed, programmed in C/C++. Up to 16 bits with smallest |L|-values have being inverted, i.e. maximal 2^{16} trials of bit inversions have been performed in each simulation.

For the measurement of the coding gain, a parameter named Cryptographic Check Error Rate (CCER) is used. CCER is a block error rate, whereby each block consists of a message and its CCV.

CCER is defined as follows:

$$CCER = \frac{number\ \ of\ \ incorrect\ \ blocks}{number\ \ of\ \ received\ \ blocks} \quad (2)$$

where an incorrect block is a block which did not pass the verification.

The results of simulations are presented in Fig. 3, showing the achieved coding gain.

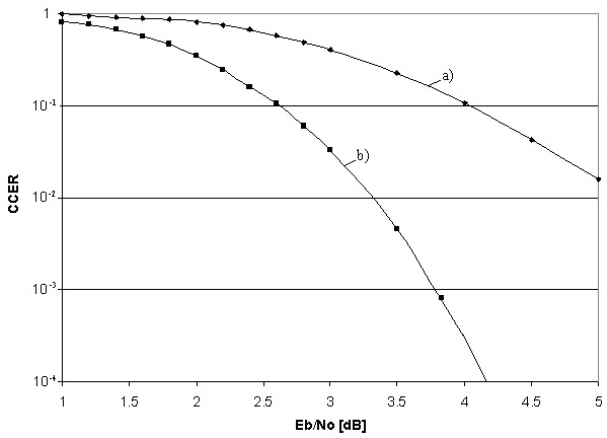

Fig. 3 Coding Gain of SID (b))compared to the Communication Systems in Fig. 1 without SID (a)).

4. Threshold based Soft Input Decryption

Threshold based Soft Input Decryption (TSID) enables further improvements of the coding gain. It uses the sensitivity of cryptographic MACs for the improvement of the decoding results. The new verification process will be introduced, which is not as hard as the standard one.

The differences between SID and TSID exist of two main points:

(i) SID uses iterative inversion of bits of the received message M´ and received CCV', whereby TSID is based on iterative inversion of the bits of M' only;
(ii) TSID uses standard verification of MACs, whereby the verification is based on the condition, that the Hamming distance HD between the received CCV' and recalculated cryptographic check value of the corrected message, CCF (M"), has to be smaller than a predefined threshold d_max (see Fig. 4).

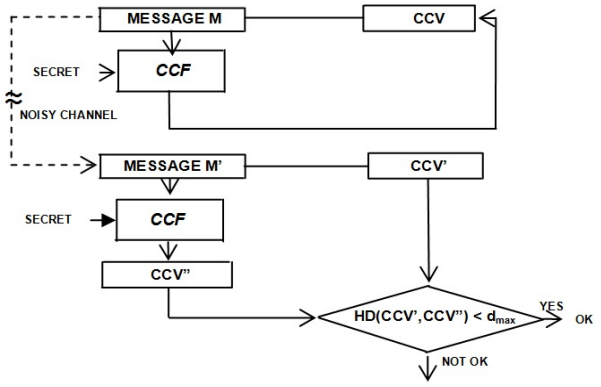

Fig. 4. Soft Verification.

The background for the success of the algorithm is the "avalanche criterion" [14, 15] of CCF: If M' and CCV' do not result in a positive verification, M' or CCV' or both of them are modified during transmission. If M' is correct and CCV' has been modified by the noise of the channel, the Hamming distance HD(CCV', CCV") will correspond to the BER (Bit Error Rate) after the channel decoder. If M' is not correct, around 50\% of the bits of CCV" are different.

If both M and CCV have been modified during the transmission, then the behavior is as in the case of a modified M.

The background for the success of the algorithm is the "avalanche criterion" [14, 15] of CCF: If M' and CCV' do not result in a positive verification, M' or CCV' or both of them are modified during transmission. If M' is correct and CCV' has been modified by the noise of the channel, the Hamming distance HD(CCV', CCV") will correspond to the BER (Bit Error Rate) after the channel decoder. If M' is not correct, around 50\% of the bits of CCV" are different. If both M and CCV have been modified during the transmission, then the behavior is as in the case of a modified M.

Threshold based Soft Input Decryption works as follows:
Input: received message M', received cryptographic check value CCV', L-values of the message and its cryptographic check value, the maximal number of iterations i_max, threshold value d_max;
Output: corrected message M" with its cryptographic check value CCV"; or FAILURE; (optional: additional M' and CCV', so that the next entity gains the information, how the correction was made, according to the principle that no knowledge should be lost on the decoding part)

(i) Reorder the bits of M' in increasing sequence of their absolute LLR values;
(ii) Verification: if HD (CCV', CCF (M')) <= d_max, then go to (v); i = 0;
(iii) If i <= i_max: invert bits of the next combination of lowest absolute LLR values of M', resulting in M" and go to (iv); else output FAILURE;
(iv) Verification: if HD (CCV', CCF (M")) <= d_max, go to (v); else increment i, go to (iii);
(v) Output M" and CCV".

The statistical distribution of d = HD (CCV', CCF (M")) has to be studied, in order to determine the appropriate value for the decision threshold d_max. The probability mass function pmf of different values of d for BER after channel decoding with length m of the message is given by:

$$pmf(d) = pmf_1(d) \cdot P_{correct} + pmf_2(d) \cdot P_{wrong} \quad (3)$$

where $P_{correct}$ and P_{wrong} are the probabilities that M' doesn't contain errors, i.e. that M' contains errors respectively:

$$P_{correct} = (1 - BER)^m \quad (4)$$

$$P_{wrong} = 1 - (1 - BER)^m \quad (5)$$

The Hamming distance d is expected to be small if the verification is successful - smaller than the decision threshold d_max. In that case, CCF (M') is equal to the original CCV of M (because M' is equal to original M) and d is equal to the number of errors in CCV' only. d_max should be defined in such a way, that it is not smaller than the expected number of errors in CCV'. Since the remaining errors after SISO channel decoder are assumed to be uniformly distributed over CCV' (with the

length of n bits), the number of errors in CCV' has a binomial distribution B(n, BER) given as:

$$pmf_1(d) = \binom{n}{d} BER^d \cdot (1 - BER)^{n-d} \qquad (6)$$

with the mean value nBER and the standard deviation sigma^2 = nBER(1 - BER).
HD(CCV', CCF (M')) has a large value in the case of unsuccessful verification, which is above the decision threshold d_max. The reason is: if the message is wrongly decoded (M' is incorrect, i.e. contains one or more errors) the number of errors in CCF (M') is expected to be n/2 due to the "avalanche criterion". In this case, CCF (M') can take any of 2n values of the same probability.
The expected value of HD(CCV', CCF(M')) is, if the message is not correct, equal to the expected value of HD between CCV' and any other fixed bit pattern of the same length. Therefore $pmf_2(d)$ has also a binomial distribution B(n,BER), where BER = ½ since every bit in CCV' is expected to be 0 or 1 with the same probability:

$$pmf_1(d) = \binom{n}{d}^d \cdot \frac{1}{2^n} \qquad (7)$$

Two regions can be clearly distinguished:

(i) D_1 for 0 <= d <= d_1 - if M' is correct (M = M') and
(i) D_2 for d_2 <= d <= n - if M' is wrong (M ≠ M')

where d_1 and d_2 are Hamming distances which define boundaries of regions D_1 and D_2.

The decision threshold d_max can be any value of the middle area between regions D_1 and D_2, i.e. d_1 < d_max < d_2. In that case, d_1 and d_2 can be considered as a lower and an upper limit of the threshold value d_max respectively. In the next Chapter it will be explained, how these two values, which are essential for the choice of the threshold d_max, are calculated.

5. Criterions for the Choice of the Lower and Upper Limit

Values d_1 and d_2 are important for setting the threshold value d_max, which is necessary for the TSID algorithm. d_1 and d_2 depend on the probability of non detection and of miscorrection respectively. These probabilities define the criterions for the choice of d_1 and d_2.

Criterion for the Calculation of d_1

The acceptance rate of messages by the verification process is greater, if d_max is set to a greater value. Non detection of a message is the event, that the correct decoded or by bit flipping iterations corrected message could not been verified. The reason for that can be that the value of d_max is set to a low value, so that the condition for the successful verification is not fulfilled. Therefore, the value of d1 should not be too low, so that the threshold value d_max is also not too low and non detection is avoided.
The criterion for the choice of the lower limit d_1 is that the probability of non detection is less than $10^{-k}1$.

$$d_1 = \min_{0 \le d_1 \le n} \{ d_1 \mid \sum_{d=d_1+1}^{n} P_{correct} \cdot pmf_1(d) \le 10^{-k_1} \} \qquad (8)$$

where the parameter k_1 can be chosen by the system designer.

The values of d_1 and the region D_1 for k_1 = 4, MAC of length n = 128 and the message of length m = 200 bits are presented in Fig. 5 in dependence on BER after MAP SISO channel decoding with the same parameters as for SID (Chapter 3), for different signal-to-noise ratios of the channel E_b/N_0. Values of d_1 for different parameters k_1 and E_b/N_0 are given in Table 1.

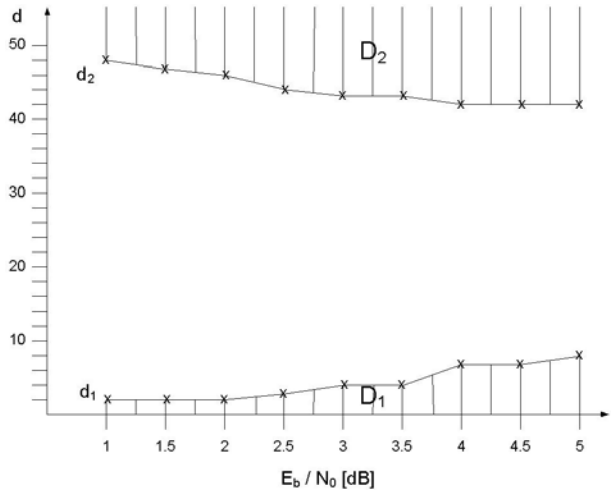

Fig. 5. Regions D_1 and D_2 with d_1 and d_2 for $k_1 = k_2 =$, m = 200 and n = 128.

E_b/N_0 [dB]	BER	$k_1 = k_2 = 4$		$k_1 = k_2 = 6$		$k_1 = k_2 = 10$	
		d_1	d_2	d_1	d_2	d_1	d_2
1	0.036	8	42	12	39	14	28
1.5	0.0234	7	42	11	39	12	28

2	0.0149	7	42	10	39	10	28
2.5	0.00681	4	43	7	40	8	28
3	0.00376	4	43	6	40	6	29
3.5	0.00142	3	44	5	41	5	29
4	0.00037	2	46	3	43	3	30
4.5	0.00024	2	47	3	43	3	31
5	0.00012	2	47	2	44	3	31

Table 1. Values of d_1 and d_2 for different values of k_1 and k_2, m = 200 and n = 128 in dependence on signal-to-noise ratio of the channel

Criterion for the Calculation of d_2

On one hand, greater d_max causes a higher acceptance rate of messages and speeds up the verification process, because the number of bit-flipping iterations leading to successful verification is smaller. On the other hand, a greater d_max increases the probability of miscorrections. A miscorrection is the event that the verification algorithm decides, that the wrongly decoded message or not corrected message is correct after bit-flipping iterations. The probability of miscorrection increases, when d_max increases. Therefore, the upper limit d_2 should be set to a value, which reduces the probability of miscorrections. The upper limit d_2 of the threshold can be determined regarding the probability of miscorrection which can be tolerated. This probability is defined by the use of parameter k_2, while d_2 will be the maximal integer that satisfies the following condition:

$$d_2 = \max_{0 \le d_1 \le n} \{ d_2 \mid \sum_{0}^{d_2} P_{wrong} \cdot pmf_2(d) \le 10^{-k_2} \} \quad (9)$$

The values of d_2 and the region D_2 for $k_2 = 4$, MAC of a length n = 128 and the message of length m = 200 bits are presented in Fig. 5 in dependence on signal-to-noise ratios of the channel E_b/N_0. Values of d_2 for different parameters k_2 and different values of E_b/N_0, which impacts the BER after MAP SISO channel decoding, are given in Table 1.

6. Non Detection and Miscorrection Probability

Non detection happens, if the message is correct and the Hamming distance d > d_max. The probability of non detection P_{nd} is given by:

$$P_{nd} = \sum_{d=d_{\max}+1}^{n} P_{correct} \cdot pmf_1(d) \quad (10)$$

or, using equations (3) and (5):

$$P_{nd} = \sum_{d=d_{\max}+1}^{n} (1-BER)^m \cdot \binom{n}{d} \cdot BER^d \cdot (1-BER)^{n-d} \quad (11)$$

Miscorrection happens, if the message is wrong and the Hamming distance d ≤ d_max. The probability of miscorrection P_{mc} is given by:

$$P_{mc} = \sum_{d=0}^{d_{\max}} P_{wrong} \cdot pmf_2(d) \quad (12)$$

or, using equations (4) and (6):

$$P_{mc} = \sum_{d=0}^{d_{\max}} (1-(1-BER)^m) \cdot \binom{n}{d} \cdot \frac{1}{2^n} \quad (13)$$

Values of non detection and miscorrection probability are presented in Table 2 and Table 3 respectively, for different values of d_max in dependence on signal-to-noise ratio of the channel, i.e. BER after SISO channel decoding (same values as in Table 1).

d_{max}	E_b/N_0= 1 dB	E_b/N_0 = 2 dB	E_b/N_0 = 3 dB	E_b/N_0 = 4 dB	E_b/N_0 = 5 dB
0	$6.47 \cdot 10^{-4}$	$4.24 \cdot 10^{-2}$	$0.80 \cdot 10^{-1}$	$4.30 \cdot 10^{-2}$	$1.49 \cdot 10^{-2}$
1	$6.19 \cdot 10^{-4}$	$2.83 \cdot 10^{-2}$	$3.97 \cdot 10^{-2}$	$1.00 \cdot 10^{-3}$	$1.13 \cdot 10^{-4}$
2	$5.51 \cdot 10^{-4}$	$1.48 \cdot 10^{-2}$	$6.03 \cdot 10^{-3}$	$1.55 \cdot 10^{-5}$	$5.69 \cdot 10^{-7}$
3	$4.45 \cdot 10^{-4}$	$6.21 \cdot 10^{-3}$	$6.93 \cdot 10^{-4}$	$1.79 \cdot 10^{-7}$	$2.13 \cdot 10^{-9}$
4	$3.20 \cdot 10^{-4}$	$2.15 \cdot 10^{-3}$	$6.38 \cdot 10^{-5}$	$1.64 \cdot 10^{-9}$	$6.35 \cdot 10^{-12}$
5	$2.05 \cdot 10^{-4}$	$6.32 \cdot 10^{-4}$	$4.87 \cdot 10^{-6}$	$1.24 \cdot 10^{-11}$	$1.56 \cdot 10^{-14}$
6	$1.17 \cdot 10^{-4}$	$1.60 \cdot 10^{-4}$	$3.18 \cdot 10^{-7}$	$8.01 \cdot 10^{-14}$	$3.26 \cdot 10^{-17}$
7	$6.01 \cdot 10^{-5}$	$3.54 \cdot 10^{-5}$	$1.80 \cdot 10^{-8}$	$4.48 \cdot 10^{-16}$	$5.92 \cdot 10^{-20}$
8	$2.77 \cdot 10^{-5}$	$6.97 \cdot 10^{-6}$	$9.01 \cdot 10^{-10}$	$2.21 \cdot 10^{-18}$	$9.48 \cdot 10^{-23}$

Table 2. P_{nd} for m = 200 and n = 128 in dependence on signal-to-noise ratio of the channel for different values of d_max

d_{max}	E_b/N_0= 1 dB	E_b/N_0 = 2 dB	E_b/N_0 = 3 dB	E_b/N_0 = 4 dB	E_b/N_0 = 5 dB
0	$2.94 \cdot 10^{-39}$	$2.79 \cdot 10^{-39}$	$7.27 \cdot 10^{-40}$	$2.09 \cdot 10^{-40}$	$6.97 \cdot 10^{-41}$
1	$3.79 \cdot 10^{-37}$	$3.60 \cdot 10^{-37}$	$9.38 \cdot 10^{-38}$	$2.70 \cdot 10^{-38}$	$8.99 \cdot 10^{-39}$
2	$2.42 \cdot 10^{-35}$	$2.31 \cdot 10^{-35}$	$6.00 \cdot 10^{-36}$	$1.73 \cdot 10^{-36}$	$5.75 \cdot 10^{-37}$
3	$1.03 \cdot 10^{-33}$	$9.76 \cdot 10^{-34}$	$2.54 \cdot 10^{-34}$	$7.33 \cdot 10^{-35}$	$2.44 \cdot 10^{-35}$
4	$3.24 \cdot 10^{-32}$	$3.08 \cdot 10^{-32}$	$8.01 \cdot 10^{-33}$	$2.31 \cdot 10^{-33}$	$7.68 \cdot 10^{-34}$
5	$8.09 \cdot 10^{-31}$	$7.70 \cdot 10^{-31}$	$2.00 \cdot 10^{-31}$	$5.78 \cdot 10^{-32}$	$1.92 \cdot 10^{-32}$
6	$1.67 \cdot 10^{-29}$	$1.59 \cdot 10^{-29}$	$4.14 \cdot 10^{-30}$	$1.19 \cdot 10^{-30}$	$3.97 \cdot 10^{-31}$
7	$2.94 \cdot 10^{-28}$	$2.80 \cdot 10^{-28}$	$7.89 \cdot 10^{-29}$	$2.10 \cdot 10^{-29}$	$6.98 \cdot 10^{-30}$
8	$4.49 \cdot 10^{-27}$	$4.27 \cdot 10^{-27}$	$1.11 \cdot 10^{-27}$	$3.21 \cdot 10^{-28}$	$1.07 \cdot 10^{-28}$

Table 3. P_{mc} for m = 200 and n = 128 in dependence on signal-to-noise ratio of the channel for different values of d_max

7. Simulation Results

Simulations have been performed using the same parameters as for those of SID (Chapter 3).

The value of the decision threshold of TSID has been set to d_max = d_1 = 8, as the worst case in Table 1 for d_1 if k_1 = 4.
The results of simulations are presented in Fig. 6, showing the coding gain in comparison to channel decoding and Soft Input Decryption.

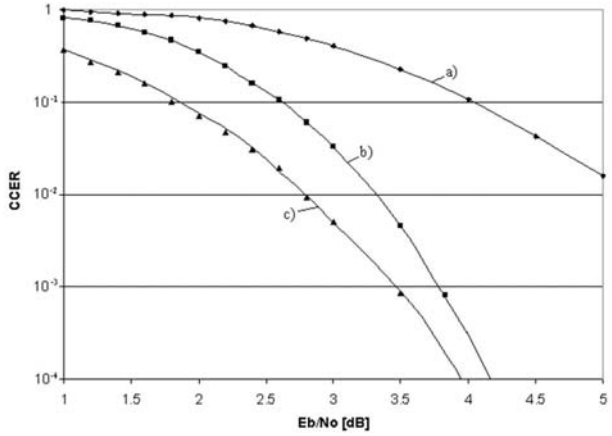

Fig. 6. Coding gain of SID (b)) and TSID (c)) in comparison to the standard communication systems (a))

8. Security Aspects

The security of the transmitted message is obviously reduced by the algorithm of TSID: an attacker can modify or generate a message appending any bit string as CCV without knowledge of the secret key and forward it to the receiver. The receiver calculates the CCF of the message from the attacker and compares it with CCV' received from the attacker. If the condition for the successful verification is fulfilled, i.e. if the Hamming distance between the recalculated and received CCV' is less than the threshold d_max, the message from the attacker will be accepted as being authentic. Additionally, it is assumed, that the attacker is able to influence the signal-to-noise and to control by this way the LLR values, which determine the bit flipping of the message during the iteration process of the verification algorithm.

The probability that an attack is successful in a single iteration is given by:

$$P_{sa,1iteration} = P_{mc} = (1-(1-BER)^m)\sum_{d=0}^{d_{\max}} \frac{\binom{n}{d}}{2^n} \quad (14)$$

The probability of a successful attack P_{sa}, if the maximal number i_{max} of iterations is performed, is:

$$P_{sa} = P_{sa,1iteration} \cdot \sum_{i=0}^{i_{\max}-1} (1-P_{sa,1iteration})^i \quad (15)$$

The resulting security level reduced from the original security level of $1/2^n$ is presented in Table 4 for m = 200, n = 128, $i_{max} = 2^{16}$ and some typical values of d_max.

In order to compensate for the reduction of the security level, the length of MAC can be extended to n_1, such the original security level of a CCV length of n = 128 is re-established:

$$P_{sa}(n_1) \le \frac{1}{2^n} \quad (16)$$

The extended length n_1 of CCV is also presented in Table 4.

The coding gain of the worst case requiring an extended length of n_1 =193 instead of n=128 bits of CCV for compensation is shown by the dashed line d) in Fig. 7. Despite of the reduction of the coding gain compared to c), a remarkable coding gain still remains.

d_max	$P_{sa}(n)$	n_1
0	2^{-105}	151
1	2^{-99}	157
2	2^{-93}	163
3	2^{-87}	169
4	2^{-82}	174
5	2^{-77}	179
6	2^{-72}	184
7	2^{-67}	189
8	2^{-63}	193

Table 4. Loss of security level P_{sa} and extended length m_1 of MAC for compensation for different d_max

9. Applications of Threshold based Soft Input Decryption

CCVs as MACs and H-MACs are used more and more in industrial applications to support data integrity of messages exchanged between sensors, robots, metering devices, and control units. Each of these messages is short

and exists only of a few octets (sometimes called bytes), but each message is secured by a CCV. A message transmitted in automotive applications, for example sent over a CAN bus has less than 20 octets, messages with metering information contain also only a few octets, typically consisting of destination address, meter id, timestamp or sequence number, type and length fields, and the meter value. These messages are transmitted very often in electrically and magnetically disturbed environments, sometimes wireless. Therefore, they are exposed to a high noise, which causes a low signal-to-noise ratio. The channel code can correct some of the errors caused by the noise, but there will always be a remaining bit error rate. The application of the method of Automatic Repeat Request (ARQ) in the case that CCV is not verified as to be correct, is not possible in many cases of industrial applications, because the transmission mode is connectionless, and/or real-time oriented, which does not allow for a repetition or even an iterative repetition till the received CCV fits to the received message. Therefore, the Threshold Based Soft Input Decryption is a very appropriate technique for industrial applications.

Conclusion

The interest in robustness of (secure) communication is increasing. Two approaches exist for message authentication over noisy channels:
(i) Provision of special error tolerant message authentication codes
(ii) Provision of error tolerant verification of standard message authentication codes by the receiver.
In both cases the security is affected i.e. reduced and have to be compensated, for example, by longer message authentication codes.
Using Threshold based Soft Input Decryption, cryptographic check values (MAC) can be used for the correction of messages modified due to the channel noise. The Hamming distance of the received MAC and the MAC of the corrected message corresponds then to the bit error rate after SISO channel decoding. The range of values of the decision threshold in the verification process has been determined under consideration of the risk of non detection on one hand, and of miscorrection on the other hand. Simulations show that a significant coding gain can be achieved by the use of the TSID algorithm. This loss of security, which is the price of the introduced algorithm can be compensated by using longer MACs. The result of such compensation means a minor loss of coding gain of TSID. Nevertheless, the final coding gain is even in the worst case still remarkable, recommending Threshold based Soft Input Decryption for a number of industrial applications.

References

[1] Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, ISO/IEC 9797-1, 2nd edition waiting for publication
[2] Graveman R. F., Fu K. E.: Approximate message authentication codes, in Proc. 3rd Annual Fedlab Symp. Advanced Telecommunications/Information Distribution, vol.1, College Park, MD, 1999.
[3] Xie L, Arce G. R., Graveman R. F.: Approximate Image Message Authentication Codes, IEEE Trans. On Multimedia, vol.3, no.2, 2001.
[4] Boncelet C. G. Jr.: The NTMAC for Authentication of Noisy Messages, IEEE Trans. On Information Forensics and Security, vol.1, no.1., 2006.
[5] Liu Y, Boncelet C. G. Jr.: The CRC-NTMAC for Noisy Message Authentication. IEEE Military Communication Conference, MILCOM, 2005.
[6] Ruland C., Živić N.: Soft Input Decryption, 4th Turbocode Conference, 6th Source and Channel Code Conference, VDE/IEEE, Munich, 2006.
[7] Chase D.: A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, IEEE Trans. Inform. Theory, IT- 18, pp. 170-182, 1972.
[8] Forney G. D. Jr.: Generalized Minimum Distance Decoding, IEEE Trans. Inform. Theory, IT-12, pp. 125-131, 1966.
[9] Kabatiansky G, Krouk E, Semenov S.: Error Correcting Coding and Security for Data Networks, Analysis of the Superchannel Concept, John Wily and Sons, 2005.
[10] Information technology - Security techniques - Message Authentication Codes (MACs) - Part 2: Mechanisms using a hash-function, ISO/IEC 9797-2, 2nd edition waiting for publication, 2011.
[11] Lin S, Costello D. J.: Error Control Coding, Pearson Prentice Hall, USA, 2004.
[12] Ruland C., Živić N.: Soft Input Decryption using feedback, 7th Source and Channel Code Conference, VDE/IEEE, Ulm, 2008.
[13] Živić N.: Joint Channel Coding and Cryptography, Shaker Verlag, Aachen, 2008.
[14] Bahl L, Jelinek J, Raviv J, Raviv F. Optimal decoding of linear codes for minimizing symbol error rate, IEEE Transactions on Information Theory, IT-20, pp. 284-287, 1974.
[15] Hays H. M., Tavares S. E.: Avalanche characteristics of Substitution - Permutation Encryption Networks, IEEE Trans. On Computers, Vol. 44, Nr. 9., 1995.
[16] Gomez F. S., Andina R. J. J., Mandado E.: Concurrent Error Detection in Block Ciphers, in Proc. IEEE Int. Test Conf., Atlantic City, NJ, pp. 979-984, 2000.

Dr.-Ing. Nataša Živić received diploma and Magister from the Faculty of Electrical Engineering of the Belgrade University in 1999 and 2002 respectively.She received Dr.-Ing. from the University of Siegen in 2007, where she is currently employed as a lecturer and works on a postdoctoral thesis. Her current research includes channel coding and cryptography, as well as combined applications of both. She is author of 3 monographs, about 90 publications of conference and magazine papers, two German and one USA patent and reviewer of conference and journal papers.