Improvement of Channel Decoding using Block Cypher

Natasa Zivic
Institute of Data Communications Systems
University of Siegen
Siegen, Germany
natasa.zivic@uni-siegen.de

Ayyaz Mahmood
Institute of Data Communications Systems
University of Siegen
Siegen, Germany
ayyaz.mahmood@uni-siegen.de

***Abstract*—This paper introduces two methods for the improvement of performance of channel coding using cryptography, based on concatenation of codes. Cryptography as an outer code is combined with channel coding as an inner code. The first method improves decoding of cryptographic functions. The second one uses the first method for improvement of information decoding using a block cipher Advanced Encryption Standard. Computer simulation results are included.**

Keywords- *Advanced Encryption Standard, Cryptography, Concatenated Codes, Soft Input Decryption, Encryption, Maximum A Posteriori Probability (MAP)*

I. INTRODUCTION

This paper researches the interoperability between channel coding and cryptography in order to reduce BER of the channel decoding. Therefore, soft output or so called *L*-values of SISO (Soft Input Soft Output) channel decoding are used for correction of the input of inverse cryptographic mechanisms. The channel code can be considered as an inner code and the output of the cryptographic mechanism as an outer code (Fig. 1).

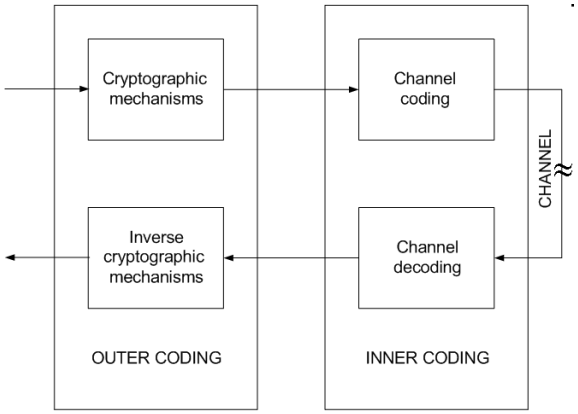

Figure 1. Cryptography and channel coding as concatenated codes.

Cryptographic mechanisms are used for the recognition of modifications by errors or manipulation. Soft output of the channel decoder enables cryptographic mechanisms to perform error corrections by Soft Input Decryption [1].

The following problems are investigated and solutions for them are proposed:

1. Improving cryptography using *L*-values of channel decoding (solution: Soft Input Decryption) – explained in Chapter II of the paper
2. Improving channel decoding using *L*-values and avalanche effect of error spreading [2] by wrong input of a decryptor (solution: Improving channel decoding using block cipher) – explained in Chapter III of the paper. AES is used as a block cipher because it is one of the most widely accepted block ciphers [3].

II. SOFT INPUT DECRYPTION

Soft Input Decryption (SID) improves decrypting mechanisms using soft output of the channel decoder [1]. A decryptor is used for verification of cryptographic check values.

Algorithm of SID is as follows:

The security mechanism is successfully completed on the receiving side if the verification results is positive. In case of negative verification, the decryptor analyzes soft output of the channel decoder, changes the bits with the lowest $|L|$-values, performs the verification process and checks the result of the verification again.

If the first verification after starting Soft Input Decryption is not successful, the bit with the lowest $|L|$-value flipped, assuming that the wrong bits are probably those with the lowest $|L|$-values. If the verification is again negative, the bit with the second lowest $|L|$-value is changed. The next try will flip the bits with the lowest and second lowest $|L|$-value, then the bit with the third lowest $|L|$-value, etc. The process is limited by the number of bits with the lowest $|L|$-values, which should be tested. The strategy follows a representation of an increasing binary counter, whereby the lowest bit corresponds to the bit with the lowest $|L|$-value, etc.

If the attempts for correction of cryptographic check values fail, the number of errors is too large as a result of a very noisy channel or an attack, so that resources are not sufficient to try enough combinations of flipping bits of low $|L|$-values.

III. CHANNEL CODING USING BLOCK CYPHER

This chapter explains the method of improving channel decoding using *L*-values and avalanche effect of error proposed error correction improvement scheme.

 ISBN: 978-1-61208-100-7

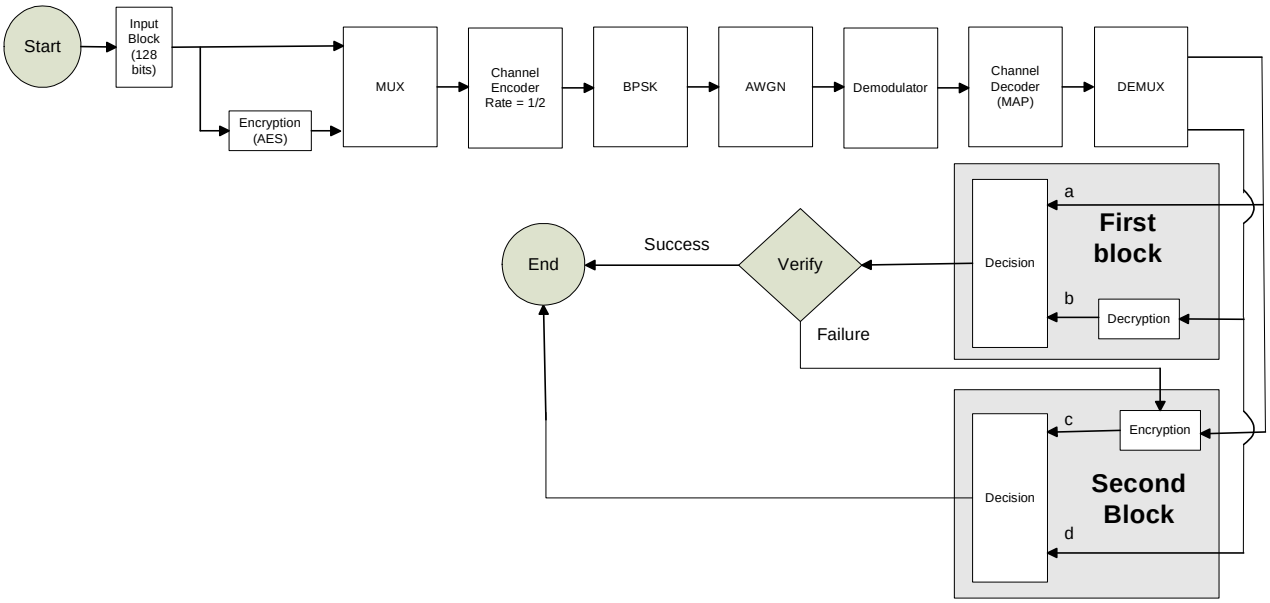

Figure 2. Error correction system using block cypher.

Fig. 2 shows the proposed error correction improvement scheme. The two blocks, which are named as first block and second block, are actually responsible for error correction improvement. The second block comes into operation in the case that first block is not able to do error correction.

Scheme in Fig. 2 includes decision blocks, which are used for making decision between two inputs applied to them. The decision block selects only one input which has the minimum number of errors.

The AES is a symmetric key (uses the same key for encryption and decryption) block encryption algorithm. The AES block size is 128 bits and that is the reason for using a block of 128 input bits in the simulations. Fig. 2 shows that in the case of the first block, decryption is used because the input data was encrypted. If the first block is not able to make error correction, the second block comes into operation. In the second block, the outputs 'c' and 'd' will also be compared to perform error correction in the case that first block is not able to do it.

A. Error Correction considering the First Block

The two outputs 'a' and 'b' applied to decision block have the following possibilities:

1) Both have errors
2) The output 'a' applied to decision block is error free whereas the output 'b' applied to decision block has about 50% errors (avalanche effect)
3) The output 'b' is error free whereas the output 'a' has errors.

The first block will be able to improve error correction considering all of the above possibilities. The output 'b' in lower branch exhibits avalanche effect because of the use of AES. It means that if MAP decoder is not able to correct all errors, then the output 'b' will have about 50% errors. The output 'a' will have significantly smaller number of errors as compared to the output 'b'. Therefore decision block will always compare output 'a' and output 'b' to check if this difference is above a certain value. This value depends upon the signal to noise ratio and is named threshold. The decision block calculates a value, which is called BER_compare for each iteration. It is calculated as a difference between BER of the output 'a' and BER of the output 'b'. If BER_compare for each iteration is higher than the threshold, then the output 'b' has about 50% errors. In this case SID is used for achieving error free output 'b' (if SID is successful). If BER_compare is lower than the threshold, then the output 'b' is error free; the decision block will select the output 'b'.

B. Soft Input Decryption using AES Block Cypher

Soft Input Decryption using AES is able to correct all errors (if it is successful) occurring after decryption at output 'b' by taking the output 'a' as a reference. It uses soft output of the channel decoder. As the magnitude of *L*-value gives the reliability of the decision, it can be used to correct all erroneous bits at output 'b'. Soft Input Decryption using block cipher AES uses the lowest sixteen *L*-values, which means that SID will have 65536 attempts for error correction. In each attempt a bit or a combination of bits are flipped (0 to 1 or 1 to 0) and then decryption is performed. For each attempt BER_compare is calculated and compared with the threshold until it becomes less than the threshold.

When BER_compare is less than the threshold, all of the errors at output 'b' are corrected. The decision block will then select the output 'b' because it is error free. It can also happen that within 65536 attempts, SID is not successful. In that case second block comes into operation.

 ISBN: 978-1-61208-100-7

C. Error Correction considering the Second Block

If Soft Input Decryption is not able to correct errors in the first block, the second block attempts to achieve it. In the case of the second block, upper branch is encrypted after MAP decoder, so the avalanche effect will be present at output 'c'. The decision block will therefore treat output 'c' exactly like output 'b' and output 'd' exactly like output 'a'. The error correction can be done in the same way as it was performed for the first block. Instead of SID, the second block performs Soft Input Encryption. If BER_compare is higher than threshold, the lowest sixteen $|L|$-values will be flipped at the input of the encryption block until all errors are corrected (if Soft Input Encryption is successful).

IV. SIMULATION RESULTS

It is explained that the improvement in error correction can be achieved using Soft Input Decryption and Soft Input Encryption which depends upon the threshold. The simulated curve for threshold vs. E_b/N_0 is shown in Fig. 3. The curve shows that threshold decreases with the increase of E_b/N_0. The reason is that with the increase of E_b/N_0, the channel introduces fewer errors.

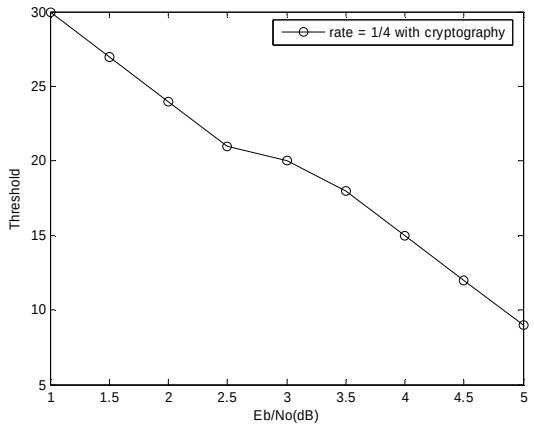

Figure 3. Threshold versus E_b/N_0 for error correction system using cryptography.

The proposed system shown in Fig. 2 has an overall code rate of 1/4, if it compared to a standard error correction system without cryptography having a convolutional encoder of rate 1/4. The convolutional encoder of rate 1/2 is a non-systematic (2,1,3) convolutional encoder and a convolutional encoder of rate 1/4 is a non-systematic (4, 1, 3) convolutional encoder [4]. These two encoders were selected because they have the same coding gain and the similar structure, which enables fair comparison of decoding results [4].

BPSK modulation, AWGN channel and MAP [5] convolutional decoder are used in simulations. For purposes of Soft Input Decryption / Encryption, maximum 16 lowest L-values are used (2^{16} correction trials).

Fig. 4 shows that the error correction system using cryptography achieves considerable coding gain over 1/4 convolutional decoder: 1.3 dB for BER of 10^{-6} and 1.85 dB for BER of 10^{-7}.

For E_b/N_0 higher than 2.4 dB, there is a coding gain of the error correction system presented in Fig. 3, which increases with increase of E_b/N_0 in comparison to 1/4 convolutional decoder. For E_b/N_0 lower than 2.4 dB, presented error correction system gives worse decoding results than the comparable 1/4 convolutional decoder.

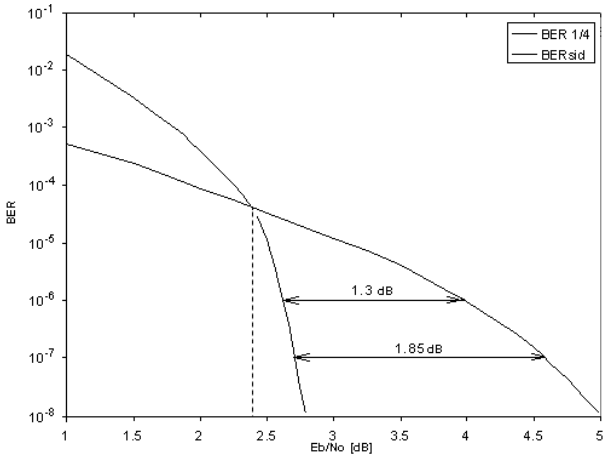

Figure 4. BER versus E_b/N_0 for error correction system with and without cryptography.

V. CONCLUSION

The paper introduces two methods of interoperability of channel coding and cryptography: improving decryption using Soft Input Decryption and improving decoding results using block cipher and principles of Soft Input Decryption. The characteristic of cryptographic check values to give about 50 % of wrong bits at output of decryptor if one or more bits at input of decryptor are wrong, is used for bit error correction of decoded information. Bit error correction scheme with two blocks (for Soft Input Decryption and Soft Input Encryption) is presented and simulated.

Simulation results show that, if 16 lowest L-values are used for Soft Input Decryption / Encryption, a remarkable coding gain in comparison to the standard 1/4 convolutional decoder can be achieved for E_b/N_0 higher than 2.4 dB: for BER of 10^{-6} coding gain is equal 1.3 dB and for BER of 10^{-7} coding gain achieves 1.85 dB. For E_b/N_0 lower than 2.4 dB, 1/4 convolutional decoder is stronger in error correction than the presented error correction scheme.

REFERENCES

[1] N. Zivic, C. Ruland, "Soft Input Decryption", 6th Source and Channel Code Conference, VDE/IEEE, Munich, vol. April 2006.

 ISBN: 978-1-61208-100-7

[2] S. Fernandez-Gomez , J. J. Rodriquez-Andina, E. Mandado, “Concurrent error detection in block ciphers”, in Proc. IEEE Int. Test Conf., Atlantic City, NJ, 2000, pp. 979-984

[3] National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standard FIPS PUB 197, November 26, 2001

[4] S. Lin, D.J. Costello, “Error Control Coding”, Pearson Prentice Hall, USA, 2004

 ISBN: 978-1-61208-100-7