# Concatenation and Turbo Principle of Channel Coding and Cryptography

**Natasa Zivic**[†]

University of Siegen, Hoelderlinstrasse 3, Siegen, Germany

**Summary**

The transmission of data coded by a convolutional or turbo code and secured by cryptographic check values is improved by using of Soft Input Decryption. An additional coding gain can be reached, when coded data with cryptographic check values are interleaved, decoded and decrypted in an iterative process with feedback from decryptor to channel decoder. The coding gains are given complimentary, because the cryptographic redundancy is added by security aspects, which become more and more important and common.

This paper observes the method of iterative Soft Input Decryption with feedback for concatenated codes of an inner convolutional or turbo code and cryptography as an outer error recognizing code. In this case the code rate diminished by the outer code has to be considered if the advantages of this method are exhibited. The coding gain of this method is so great that the code rate of the outer code can be compensated by puncturing. The extension of this method to more than two iterations leads to a Turbo principle of concatenated codes

***Key words:***

*Concatenated Codes, Cryptographic Check Values, SISO Convolutional Coding, Decrypting.*

## 1. Introduction

The cooperation between channel coding and cryptography has been researched using channel decoding for the improvement of decryption results and, vice versa, using cryptography for the improvement of channel decoding [1], [2]. This concept is called Joint Channel Coding and Cryptography and is based on Soft Input Decryption with feedback. The main idea of Soft Input Decryption is to use the soft output (L-values [3][4]) of SISO (Soft Input Soft Output) channel decoding to correct the input of cryptographic mechanisms. The feedback from Soft Input Decryption to SISO channel decoding is used for a further reduction of BER of the SISO channel decoder. The channel code can be considered as an inner code, used for error correction, and the cryptographic mechanism as an outer code, used for recognition of modifications by errors or manipulation. The combination of an inner and outer code is known as concatenated codes [5] or general concatenated codes [6].

If security mechanisms are not applied, any other error recognizing code can be used instead of cryptographic mechanisms as an outer code ([7]). In this paper additionally puncturing is used in such a way, that the outer code becomes free of charge, as the overall code rate of the presented method is the same as the rate of the inner code.

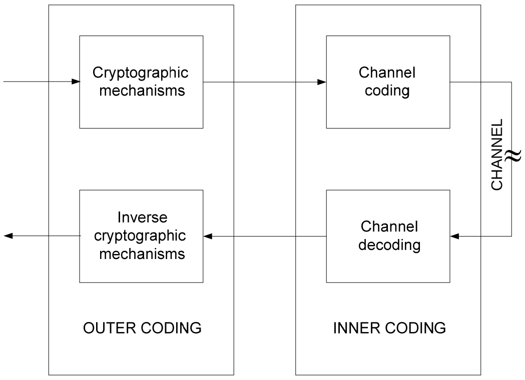

Fig. 1 Block model of Joint Channel Coding and Cryptography.

## 2. Joint Channel Coding and Cryptography

Soft Input Decryption (SID) [1] uses soft output of the channel decoder and cryptographic mechanisms, which cryptographic check values to data block, for example a message authentication code (MAC) or a digital signature. The input of Soft Input Decryption is called SID block.

The verification of the SID block is successful if the received redundancy check value is equal to the redundancy check value calculated of the received message. If the verification is not successful, the bits with the lowest |L|-values are inverted. The verification process is repeated and the result of the verification is checked. If the verification is again not successful, bits of another combination of the lowest |L|-values are changed. This iterative process is finished if the verification is successful or the needed resources (number of trials or memory

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

capacity) are consumed. Different strategies can be used to choose the next candidate for verification.
The idea of inversion of the least probable bits (with the lowest reliability values) originated from Chase decoding algorithms [3] in 1972, which were the generalization of the GMD (Generalized Minimum Distance) algorithms from 1966 [4]. These algorithms are referenced as LRP (Least Reliability Positions) algorithms. The similarity to the method of the Soft Input Decryption, which is used in this work, is the use of *L*-values reordered and iteratively tested. The difference is that Soft Input Decryption uses two decoders (inner and outer).

The next mechanism of Joint Channel Coding and Cryptography uses Soft Input Decryption with feedback [2]. The input of the encryptor is a data block, which may be part of a data stream. The data block is split in two parts of the same length, message *ma* and message *mb*, both of length of *m*. Each of both messages is extended by a cryptographic check value *na* and *nb,* both of length *n*, using a crypgraphic check function RCF (generation of a digital signature, MAC/H- MAC or CRC) – see Fig. 1.

Generally, the lengths of message parts *ma* and *mb* don't have to be the same, as well as the lengths of cryptographic check values *na* and *nb* [2]. In [2] it is shown, that different lengths of *ma*, *mb*, *na* and *nb* have only marginal influence on BER and that equal lengths for *ma* and *mb* as well as for *na* and *nb* show the best results. For that reason, equal lengths of message parts as well as cryptographic check values are used in this paper.

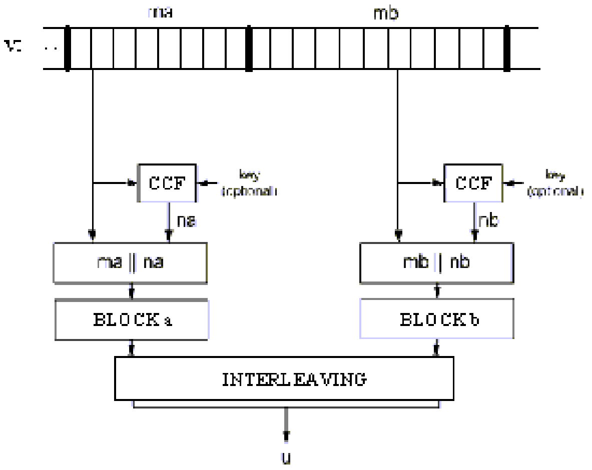

Fig. 2 Forming of a message *u*.

Block *a* consists of the message part *ma* and the redundancy check value *na*:

$$a = a_1a_2\ldots a_{m+n}= ma_1ma_2\ldots ma_m\; na_1na_2\ldots na_n \qquad (1)$$

Block *b* consists of the message part *mb* and the redundancy check value *nb*:

$$b= b_{1b}b_2\ldots b_{m+n}= mb_1mb_2\ldots mb_m\; nb_1nb_2\ldots nb_n \qquad (2)$$

Interleaving of block *a* and block *b* forms the assembled message *u*:

$$u= a_1b_1\; a_2b_2\ldots a_{m+n}\; b_{m+n} \qquad (3)$$

*u* is encoded by a convolutional or turbo code (inner code), modulated and transferred over the noisy channel.

After demodulation of the received message, Joint Channel Coding and Cryptography is applied in 3 steps (Fig. 2).

Step 1:
- channel decoding with resulting $BER_{cd1}$
- segmentation and de-interleaving of the output *u'* of the decoder into block *a´* and block *b´*, and
- parallel Soft Input Decryption with feedback of block *a´* and block *´b*.

The following steps depend on the results of step 1:

CASE 1
the results of the first Soft Input Decryption (1. SID) of block *a'* and Soft Input Decryption of block *b'* are correct, i.e. BER after 1. SID is 0:

$BER_{1.SID} = 0$

*u* is corrected and no other actions are necessary.

CASE 2
the result of Soft Input Decryption of block *a'* is correct, but block *b'* could not be corrected. So, a half of bits are corrected (belonging to block *a'*), and another half of bits (belonging to block *b'*) have *BER* as after channel decoding:

$BER_{1.SID} = ½\, BER_{cd1}$

Step 2 of CASE 2
The second step consists of feedback [2] from block *a* corrected by Soft Input Decryption to block *b*. L-values of block *a* block are set to ±∞, L-values of block *b* are set to 0, which represent unknown bits. The SISO decoder decodes *u* again with these L-values as input. Resulting *BER* after step 2 is $BER_{feedback}$.

Step 3 of CASE 2
The third step is a second Soft Input Decryption (2. SID). block *b´* is tried to be corrected by Soft Input Decryption.

Resulting *BER* after this step is $BER_{2.SID}$. As step 3 is the last step of the algorithm, total *BER* is equal to $BER_{2.SID}$.

CASE 3

The result of Soft Input Decryption of block *b'* is correct, but block *a'* could not be corrected. As in CASE 2:

$BER_{1.SID} = ½\, BER_{cd1}$

Step 2 and 3 of CASE 3

These steps correspond to step 2 and 3 of CASE 2, but with difference that the symbols a´ and b´ are exchanged.

CASE 4

Neither the result of Soft Input Decryption of block *a'* nor the result of Soft Input Decryption of block *b'* is correct: BER is equal to BER of the convolutional or turbo decoder (BER of the inner code, $BER_{cd1}$).

No further actions are possible.

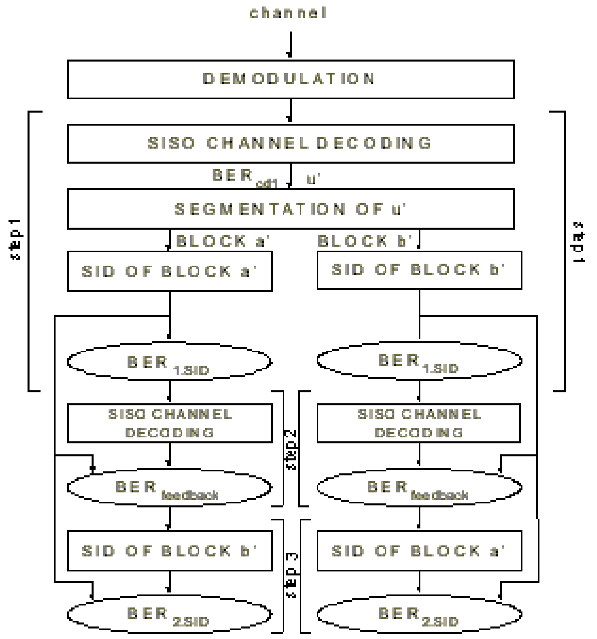

Fig. 3 Algorithm of Joint Channel Coding and Cryptography.

## 3. Puncturing

In the previous work ([1], [2]) the exploitation of cryptographic information for the improvement of decoding results was free of charge: the main function of cryptography is to secure the data transfer. In that sense, the decrease of code rate by additional redundancy – used check values - can not be avoided.

In the following the code rate is improved by puncturing. Nevertheless, the price of this improvement is a decrease of the coding gain or could even result in a negative coding gain. Puncturing is used here to enable a transfer of redundancy check values, rsp. the redundancy of an outer code, without degradation of the coding rate, i.e. the number of punctured bits is the same as the number of bits of redundancy check values, rsp. the outer code. The code rate remains the code rate of the inner code despite of the outer code.

Puncturing takes place after the inner coding before transmission over the channel and depuncturing is done before the inner decoder. All other steps of Joint Channel Coding and Cryptography are the same as in Fig. 1.

Puncturing is performed in equidistant intervals, no matter if punctured bits belong to the message part or redundancy check value.

Results of iterative SID with feedback using puncturing are presented in the next chapters.

## 4. Puncturing with Cryptographic Check Values

The simulations in this and the following chapters are performed using C/C++ programs and following parameters:

- convolutional (5,7) encoder of rate 1/2
- BPSK modulation
- AWGN channel and
- MAP decoder [8].

For each point of the curves shown in the following figures, 50 000 simulations have been performed, which are more than enough for getting a reliability of 99% of the results [9].

The BER after each step of the algorithm of Joint Channel Coding and Cryptography (Fig. 1) using puncturing is shown in Fig. 2, together with BER after channel decoding using puncturing ($BER_{punct}$).

Blocks *a* and *b* have the length of 192 bits each, with *m* = 128 and *n* = 64 bits (MAC). As the length of redundancy check values has to be great enough for security reasons, the number of bits which are punctured is high, i.e. the puncturing rate is high. The implication of high puncturing rate is lower coding gain.

Puncturing is performed corresponding to number of encoded bits of the inner code, i.e. 256 bits of 768 encoded bits are punctured (puncturing rate is 1/3).

The resulting code rate after puncturing is:

$$r = (m + m)/(2(2m + 2n) - 2(2n)) = ½ \qquad (4)$$

Fig. 3 shows that BER increases by puncturing and decreases after the second step in comparison to BER of convolutional decoding - $BER_{1/2}$. The resulting coding gain ($BER_{2.SID}$) reaches up to 1.59 dB in comparison to the convolutional decoding of the same code rate of ½.

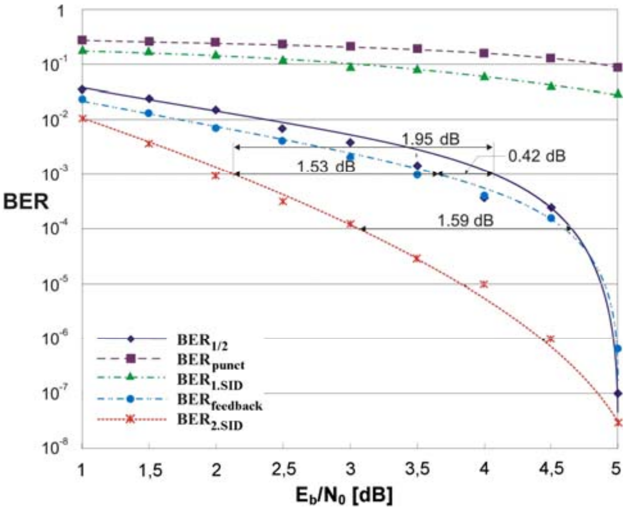

Fig. 4 BER after the SISO decoder ($BER_{1/2}$), puncturing ($BER_{punct}$) and step 2 and 3 ($m + n = 192$).

## 5. Turbo Principle of Concatenated Codes

The Turbo principle of decoding is an iterative decoding process between component decoders [10] and has been widely applied to various detection/decoding problems in recent years. The algorithm of iterative Soft Input Decryption with feedback, described in this paper, contains all characteristics of the Turbo principle: two (concatenated) codes an interleaver, SISO decoding, the exchange and feedback of L-values (extrinsic information) from one decoder to the other and iterations. Now it will be drafted, how this principle can be extended to more than one iteration, calling it Turbo Principle of Concatenated Codes.

The input of the outer code is a data block, which is split in more than two sequential parts of the same length *m* (i.e. message *ma*, *mb*, *mc*, *md*…), whereby each of messages is extended by a redundancy check valueof a length *n* (i.e. check value *na*, *nb*, *nc*, *nd*…):

$$a = a_1a_2\ldots a_{m+n} = ma_1ma_2\ldots ma_mna_1na_2\ldots na_n \qquad (5)$$

$$b = b_1b_2\ldots b_{m+n} = mb_1mb_2\ldots mb_mnb_1nb_2\ldots nb_n \qquad (6)$$

$$c = c_1c_2\ldots c_{m+n} = mc_1mc_2\ldots mc_mnc_1nc_2\ldots nc_n \qquad (7)$$

$$d = d_1d_2\ldots d_{m+n} = md_1md_2\ldots md_mnd_1nd_2\ldots nd_n \qquad (8)$$

............

Multiplexing of all block forms of a message *u*:

$$u = a_1b_1c_1d_1a_2b_2c_2d_2\ldots a_{m+n}b_{m+n}c_{m+n}d_{m+n} \qquad (9)$$

*u* is encoded by a convolutional or turbo code (inner code), modulated and transferred over a noisy channel.

After demodulation of the received message, Joint Channel Coding and Cryptography is applied in several steps, depending on the number of blocks.

The first three steps are like in Chapter II (Fig.2):
- different blocksaretriedto be corrected by SoftInput Decryption
- the L-values of blocks corrected in the first step are used as feedback to the SISO channel decoder
- Soft Input Decryption of the blocks decoded by the SISO decoder with feedback.

In the next steps algorithm is achieved iteratively: the L-values of corrected blocks are used as feedback for improved SISO channel decoding of not corrected blocks, followed by Soft Input Decryption of the uncorrected blocks.

The algorithm stops, when Soft Input Decryption is not successful for any block in an iteration. The maximum number of iterations is the number of blocks, if in each iteration one block can be corrected.

## 6. Conclusions

This paper gives an overview over the evolution of Joint Channel Coding and Cryptography principle and generalizes iterative Soft Input Decryption with feedback. The usage of an outer code has a negative influence on the overall coding rate. For this reason puncturing is used: by puncturing the coding rate is increased to the code rate of the used convolutional encoder (inner code). In this case there is no cost for the outer code.

In the last chapter of the paper, an extension of two rounds of Soft Input Decryption applied to concatenated codes, to more than two, for example four rounds. The algorithm contains all characteristics of the Turbo principle: therefore the extended strategy is called Turbo Principle of Concatenated Codes.

## References

[1] C. Ruland and N. Živić, *Soft Input Decrzption,* 4th Turbocode Conference, 6th Source and Channel Code Conference, VDE/IEEE, Munich, Germany, April 3 – 7, 2006.

[2] C. Ruland and N. Živić, *Feedbaack in Joint Channel Coding and Cryptography*, 7th Source and Channel Code Conference, VDE/IEEE, Ulm, Germany, January 14 – 16, 2008.

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

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

[5] S. Lin and D. J. Costello, *Error Control Coding*, Pearson Prentice Hall, USA, 2004.

[6] M. Bossert, Kanalcodierung, B. G. Treubner, Stuttgart 1998.

[7] N. Živić and C. Ruland, *Channel coding as a cryptography enhancer*, WSEAS Transactions on Communications, http://www.worldses.org/journals/communications/communications-march2008.htm

[8] L. Bahl. J., Jelinek, J., Raviv and F., Raviv, *Optimal decoding of linear codes for minimizing symbol error rate*, IEEE Transactions on Information Theory, IT-20, pp. 284-287, 1974.

[9] M. Jeruchim, P. Balaban and K. S. Shanmugan, *Simulation of Communication Systems*, Kluwer Academic/Plenum Publ, New York, 2000.

[10] J. Hagenauer, N. Goertz: *The turbo principle in joint source-channel coding*, Proc. of Information Theory Workshop 2003, IEEE Vol. 31, pp. 275-278, 2003.

**Natasa Zivic, Dr.,** born 1975 in Belgrade, Serbia, graduated from the Faculty of Electrical Engineering (Electronics, Telecommunication and Automatics) of the Belgrade University in 1999. at the Telecommunication Department. After the Post diploma studies at the same Faculty (Telecommunications Division) she defended her Magister Thesis (Acoustics) in 2002.

From October 2004. she was scientific assistant at the University of Siegen in Germany at the Institute for Data Communications Systems as a DAAD and University of Siegen Scholarship holder. In 2007. she defended her Doctoral Thesis on the same University. The main course of her work in Siegen is Coding and Cryptography. From 2000. till 2004. she was working at the Public Enterprise of PTT "Serbia", Belgrade as the senior engineer. Currently she is employed as an Assistant Professor at the University of Siegen. She published more than 30 articles at international Conferences and Journals and two monographs.