Method for Improvement of Soft Input Decryption Performances

NATASA ZIVIC
CHRISTOPH RULAND
Institute for Data Communications Systems
University of Siegen
Hölderlinstraße 3, Gebäude E, D-57076 Siegen
GERMANY

Abstract: - In this work, the strategy for correction of bits using Soft Input Decryption is analyzed. Soft Input Decryption is used for correction of cryptographic check values, which are very fragile when they are transmitted over noisy channels: only one wrong bit at input of the decryptor causes about 50% of errors at output of the decryptor. Soft Input Decryption corrects wrong bits of cryptographic check values in most of cases using combination of SISO channel decoding and decrypting. Soft Input Decryption is an iterative process and therefore it is needed to minimize duration of iterations. This paper suggests an algorithm for optimization of Soft Input Decryption, which enables restriction of needed time to, theoretically, about 15% of it. Suggested optimization is performed for ECDSA digital signatures.

Key-Words: - Soft Input Decryption, L-values, optimization, verification, SID block, digital signatures, control matrix

1 Introduction

This This paper studies communication systems which use cryptographic mechanisms adding cryptographic check values to support data integrity and authentication of data origin.

The idea is to use the soft output (other names are reliability values, L-values) of SISO (Soft Input Soft Output) channel decoding to correct 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). 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] – a method, which will be explained in this paper.

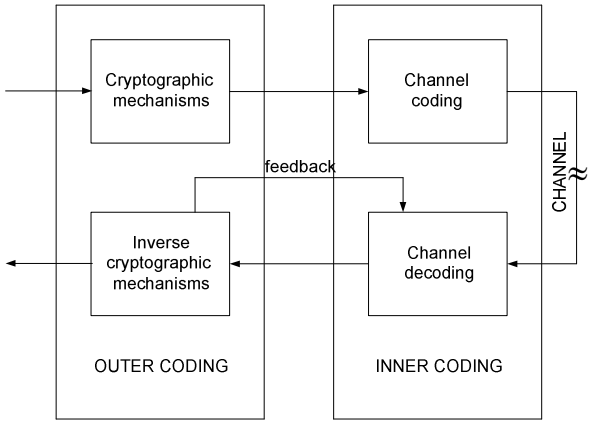

Fig. 1 Representation of channel coding and cryptographic mechanisms as inner and outer codes

Soft output values are valuable information about decoded bits, which are used in today's most efficient decoders: turbo decoders [2]. In this work *L*-values are used in a different way: as an information to the next following entity – the decrypting mechanism. Concatenation of codes, presented as an outer and inner code was already devised by Forney in 1966 [3,4]. In literature, it is known as concatenated codes [5], general concatenated codes [6] or codes of a superchannel.

The idea of inversion of the least probable bits (with the lowest reliability values) is used in Soft Input Decryption. This idea originates from Chase decoding algorithms [7] in 1972, which were the generalization of the GMD (Generalized Minimum Distance) algorithms from 1966 [4]. The similarity of Chase decoding algorithms to the method of the Soft Input Decryption, which is the subject of this paper, is the use of *L*-values reordered and iteratively tested. The difference is that Soft Input Decryption uses two decoders (inner and outer) and a non-linear block code.

Joint source channel coding is also related to this paper. The cooperation between the source and channel decoder enables a better use of information of both decoders and better decoding results [8, 9]. It is based on the turbo – principle, and performed as Softbit - Source Decoding [10, 11] and Iterative Source – Channel decoding [12, 13]. The similarity to Soft Input Decryption is the use of iterative information exchange between the two elements of the receiver: channel and source decoder, in case of joint source channel coding, rsp. channel decoder and decryptor in case of Soft Input Decryption

2 Soft Input Decryption

The basic technique which is described and used in this work is called Soft Input Decryption (SID). It consists of a decryptor which uses soft output of the channel decoder as soft input [1].

The algorithm of Soft Input Decryption (Fig. 2) is as follows:

The decryption is successfully completed, if the verification of the cryptographic check value is successful, i.e. the output is "true". If the verification is negative, the soft output of the channel decoder is analyzed and the bits with the lowest $|L|$-values are flipped (XOR "1"), then the decryptor performs the verification process and proves the result of the verification again. If the verification is again negative, bits with another combination of the lowest $|L|$-values are changed. This iterative process will stop when the verification is successful or the needed resources are consumed.

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

It may happen that the attempts for correction of SID block succeed, but the corrected cryptographic value is not equal to the original one: a collision happens. This case has an extremely low probability when cryptographic check values are chosen under security aspects. Collision aspects of cryptographic check values in Soft Input Decryption are analyzed in [20].

Soft Input Decryption is block oriented. The block which is taken from sequential input bits to the channel encoder and should be corrected by Soft Input Decryption after channel decoding is called SID block (Soft Input Decryption block).

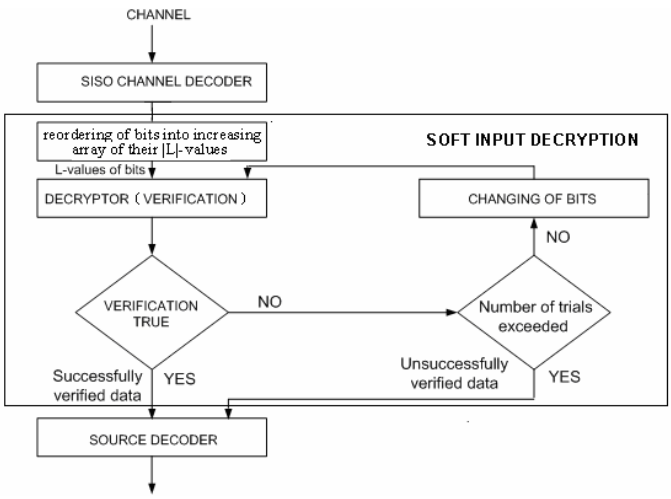

Fig. 2 Algorithm of the Soft Input Decryption

3 Strategy of Correction of Bits by Soft Input Decryption

If the first verification after starting Soft Input Decryption is not successful, the bit with the lowest $|L|$-value of the SID block is flipped, assuming that the wrong bits are probably those with the lowest $|L|$-values. If the verification is again not successful, 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.

The strategy considers only the sequence of positions of increasing $|L|$-values. Therefore, at the beginning of the algorithm the bits of the SID block are sorted increasingly according to the $|L|$-values. To control the strategy a matrix $\boldsymbol{C} = (c_{i,j})$, $i = 1, \ldots 2^{Nmax} - 1$, $j = 1, \ldots, N_{max}$ (Fig. 3) is used, where the rows are defined by the numbers from 1 to $2^{Nmax} - 1$ in binary presentation:

$$i = \sum_{j=1}^{N_{\max}} c_{i,j} 2^{j-1} \qquad (1)$$

N_{max} is the maximum number of bits to be flipped, rsp. $2^{Nmax} - 1$ is the maximum number of trials, if all verifications fail.

The algorithm flips those bits whose $|L|$-values are in positions j of coefficients c_{ij} which are marked by "1" in Fig. 3. Each row of the matrix describes one trial. Those bits are flipped, which are marked by the binary "1" in each row. The index j of those coefficients c_{ij} indicates the positions of the bits to be flipped. These indices j are used as indices j in the sorted sequence P_j. Therefore, the sequence P_j can be limited to P_{Nmax}. At the beginning of Soft Input Decryption i is reset to 0.

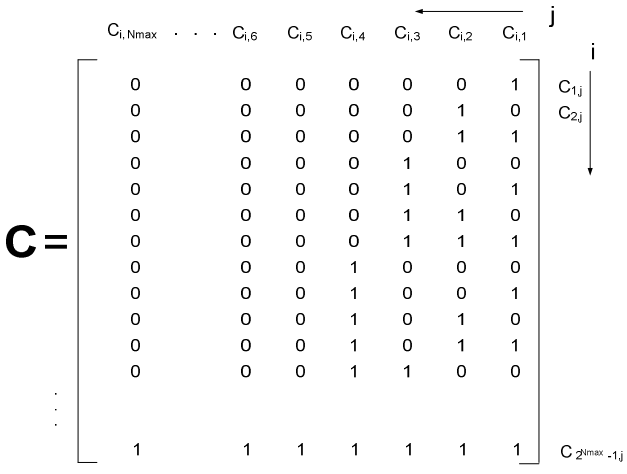

Fig. 3 Control matrix C of the correction strategy

4 Improvement of Soft Input Decryption Implementation

The optimized algorithm for verification of digital signatures uses intermediate results of previous verifications to save time needed for the next verification. Intermediate results are used to avoid re-computations, i.e. repetition of arithmetical operations.

Here it is assumed, that verification can be executed very fast, if there is only one bit of the input of the decryptor changed compared to one of the preceding verifications, and intermediate results of the preceding verifications are available. Using strategy of bit correction schedule which is explained in Chapter 2, a preceding verification always exists, which differs only by one bit from the actual signature to be verified. So the assumption is correct. Digital signatures based on ECC [16] (e.g. ECDSA) are suitable for Soft Input Decryption. Therefore, the considerations of optimization are focused on ECDSA.

An ECDSA digital signature consists of two parts, r and s, and both parts are used separately in the verification process. Therefore, a re-computation is necessary either for part r or part s, if one bit of the signature has been changed. This simple fact results in a time saving of around 50 %.

The main complex arithmetic computation of ECC is the scalar multiplication of points, which is implemented by (ld $|k|$) point additions, where $|k|$ is the length of the scalar k. Most of the point additions can be reused if one bit of k is changed, for example by one correcting point addition or point subtraction. For ld $|k| = 160 = 7.32$, for example, the verification would be 7.32 times faster. Together with the previous time saving of 50 %, the resulted time of verification would be around 15 % of the non-optimized verification time. Other ECC variants may be suitable, as for example ECKDSA, because it does not need the calculation of the inverse element for verification.

5 Optimization Algorithm

Two blocks have been added for the optimized Soft Input Strategy: "Finding of SID block for comparison" and "Decryptor (optimized verification)" (Fig. 4). The implementation of a decryptor has been changed in such a way that intermediate results have been stored. Soft Input Decryption algorithm in Fig. 2 uses verification based on one bit difference in comparison to the verification, whose results are reused. Because of that, the block "Finding of a SID block for comparison" has been added to the algorithm of Soft Input Decryption presented in Fig. 2.

The strategy is as follows: Consider matrix C from Fig. 3. Assume that i is the index of the i^{th} trial. The row C_i indicates with ''1'' the positions of the lowest $|L|$-values and therefore the bits which have to be changed. In row C_i from right to left, the first bit ''1'' is searched from right to left and changed to ''0''. The result is index i' of row $C_{i'}$ with $i' < i$. So, it is guaranteed that the SID block from i-th trial has already been verified and intermediate results exist

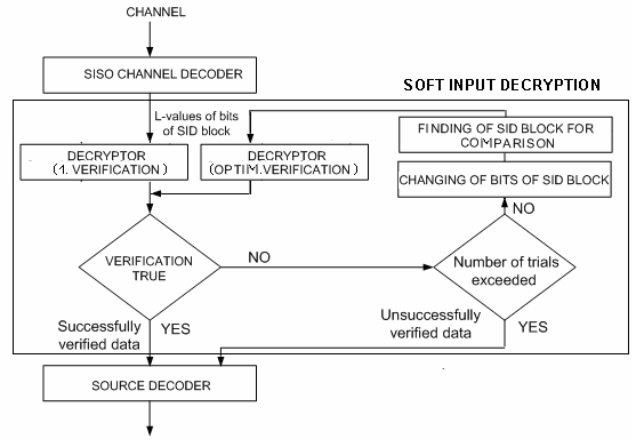

Fig. 4 Optimized scheme of Soft Input Decryption in case of digital signatures

6 Conclusion

This paper presents a method for increase of efficiency of Soft Input Decryption in case of digital signatures based on ECC, as ECDSA digital signatures. Using the mathematical background of digital signatures verifications, in combination with the binary strategy of bit correction by Soft Input Decryption, it is shown how the verification process can be reduced to about 15% of time needed for Soft Input Decryption without optimization.

The future work should realize the optimization method, showing the time optimization results. Also, optimization using other cryptographic algorithms should be investigated.

References:

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

[2] Giuiletti, A., Bougard, B., Perre, L.v.d., Turbo Codes: Desirable and Designable, *Kluwer Academic Publishers*, 2003.

[3] Forney, G.D.Jr.: Concatenated Codes, *MIT Press*, Cambridge, 1966.

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

[5] Bossert, M.: Kanalcodierung, *B. G. Treubner*, Stuttgart, 1998.

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

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

[8] Hagenauer, J.: Source-Controlled Channel Coding, *IEEE Trans. On Communications*, pp. 2449-2457, September 1995.

[9] Caire, G., Shamai, S., Verdu, S.: Almost-Noiseless Joint Source - Channel Coding-Decoding of Sources with Memory, *in Proc. for 5th international ITG Conference on Source and Channel Coding (SCC)*, Erlangen, January 2004.

[10] Adrat, M., Picard, J.-M., Vary, P.: Analysis of extrinsic Information from softbit-source decoding applicable to iterative source-channel decoding, *in Proc. for 4th ITG Conference 2002 – Source and Channel Coding*, Berlin, January 2002..

[11] Adrat, M., Picard, J.-M., Vary, P.: Efficient near-optimum softbit source decoding for sources with inter- and intra-frame redundancy, *in Proc. for IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 653-656, May 2004.

[12] Görtz, N.: Iterative Source-Channel Decoding by Channel Coded Optimal Estimation, *Proc. Of 3rd ITG Conference Source and Channel Coding*, Munich, Germany, pp. 267-272, January 2000.

[13] Adrat, M., Vary, P.: Iterative Source-Channel Decoding: Improved System Design Using EXIT Charts, *EURASIP Journal on Applied Signal Processing*, vol. 6, pp. 928-941, 2005

[14] Information technology – Security techniques – Digital signatures giving message recovery, 1997

[15] ISO/IEC 9796-3: Information technology – Security techniques – Digital signatures giving message recovery – Part 3: Discrete logarithm based mechanisms, 2006

[16] ISO/IEC 9797-1: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999

[17] ISO/IEC 9797-2: Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a hash-function, 2000

[18] ISO/IEC 9798-1: Information technology – Security techniques – Entity authentication mechanisms – Part 1: General, 1997

[19] ISO/IEC 13888-1: Information technology – Security techniques – Non–repudiation – Part 1: General, 2004

[20] Živić, N., Ruland, C.: Probability of collisions in Soft Input Decryption, American Conference on Applied Mathematics (Math 08), Harvard University, USA, March 24 – 26, 2008