Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 73 (2013) 62 – 67

Procedia
Social and Behavioral Sciences

The 2nd International Conference on Integrated Information

Improved Cryptographic Protection of Information using Robust Authentication Algorithms

Natasa Zivic*

University of Siegen, Hoelderlinstrasse 3, 57076 Siegen, Germany

Abstract

Information is typically transmitted in telecommunication systems in form of messages. Many applications nowadays demand protection of their content and/or authentication of the sender and recipient of the message. Message Authentication Codes are commonly used for the authentication purposes. They 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. A successful verification of Message Authentication Codes demands that all of bits of the received Message Authentication Code and of the bits of the recalculated Message Authentication Code from the received message are equal. Such a hard condition is not suitable for some applications. The introduction of a softer condition for the successful verification can improve the successful authentication of messages corrupted by transmission over a noisy channel. Algorithms are presented, which introduce robustness into the verification of messages protected by Message Authentication Codes, as well as an algorithm which not only verifies, but also corrects the messages corrupted due to the noisy channel.

Keywords: Information, Transmittion, Message Authentication Codes, Authentication, Verification, Noisy Channel, Algorithms

1. Introduction

Message Authentication Codes (MACs) [1] apply symmetric cryptographic algorithms, which provide data integrity and authentication of data origin. 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 erroneous. The main function of MACs is the protection against forgeries. For that reason MACs are designed in such a way, that any modification of the

* Corresponding author. Tel.: +492717403322; fax: +492717402536.
E-mail address: natasa.zivic@uni-siegen.de

doi:10.1016/j.sbspro.2013.02.020

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 firm condition of the verification of message authentication is a good protection against forgeries. Nevertheless, there are some 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 that the modifications of a single message bit or a few bits do not result in any modification of a MAC, i.e. in an unsuccessful verification. 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 message modifications.

An algorithm for correction of messages will be presented, which uses standard MACs, but with a different verification process. The received MAC and the one recalculated from 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 compared MACs are different. 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.

2. 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 next stage – the verification of cryptographic check values (CCVs) (Note: CCV is used as a generalization of MACs).

Knowing that even one erroneous bit at the input of verification will cause the verification fail, the idea was developed to try to correct the rest of the bits which remained erroneous after SISO channel decoding. Soft output of the channel decoder is used here as soft input of the cryptographic verification process (see Fig 1). 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′* from channel decoder:

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

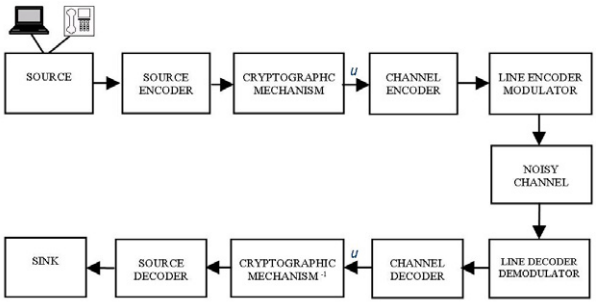

Fig. 1. Communication System

L(*u′*) expresses the reliability of the decision of the channel decoder, if the sent bit u was 1 or 0. The sign of *LLR* shows the hard output of bit u' (1 or 0) and its absolute value is used as the reliability value of the hard

decision. The higher |*LLR*|, the 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 the case when 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 equality of received cryptographic check value CCV' and the cryptographic check value CCV" which is recalculated from the received message M'("hard verification").

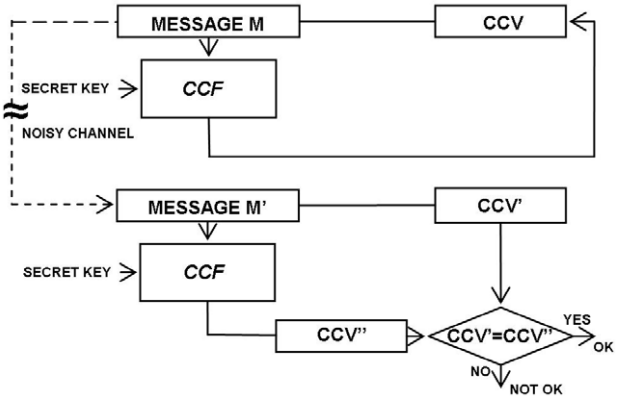

Fig. 2. Hard Verification.

The algorithm of Soft Input Decryption works on data blocks where each block contains the received message M' and the received cryptographic check value CCV', as for example MAC [1] or H-MAC [9]. Beside this, SID algorithm also uses *LLR* values for each bit of M' and CCV', taking the maximal number of iterations i_{max} as a parameter. After an iterate process, if successful, SID at the output gives corrected message M" and its (also corrected) cryptographic check value CCV". If, after i_{max} iterations, the message and/or its CCV are not corrected, SID gives a FAILURE information. The iterate process contains the following steps:

1. Reorder the bits of M' and CCV' in increasing sequence of their absolute *LLR* values;
2. Verification: if CCV' = CCF (M'), then go to step 5; i = 0;
3. If $i \le i_{max}$: invert the next combination of bits with the lowest |*LLR*| values of M' and CCV', resulting in M" and CCV" and go to step 4; else output FAILURE;
4. Verification: if CCV" = CCF (M"), go to step 5; else increment *i*, go to step 3;
5. 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 [10].

The numeric simulations of SID performed in [6] have shown a significant gain presented in Fig 3. As the measure, a parameter named Cryptographic Check Error Rate (CCER) which represents the ratio between the number of incorrect blocks after SID, and the whole number of received blocks, has been observed.

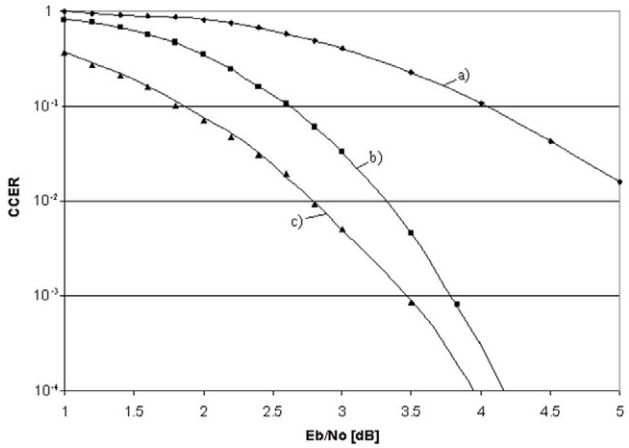

Fig. 3. Coding Gain of SID (b) and TSID (c) compared to the Communication Systems in Fig 1. without SID (a)

3. 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. Two main differences between SID and TSID are:

1. 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;
2. TSID uses standard 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).

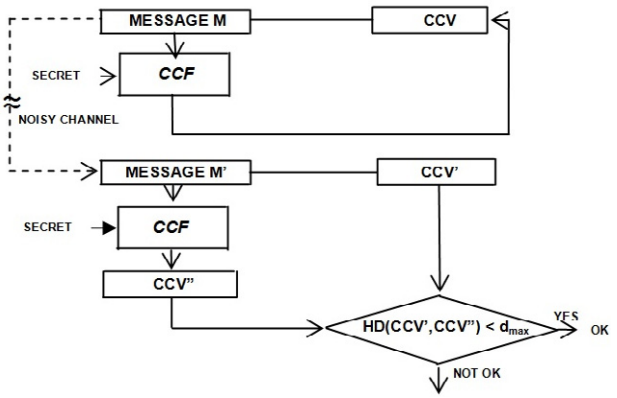

Fig. 4. Soft Verification.

The background of TSID algorithm is the "avalanche criterion" [11, 12] of CCF: if only CCV' has been modified during transmission the Hamming distance HD(CCV', CCV") is relatively small i.e. corresponds only to the Bit Eror Rate (BER) after channel decoding; if M' has been modified, then HD(CCV', CCV") will be around 50% of the block length regardless if CCV' was correct or not.

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

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

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

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

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

If the verification is successful the Hamming distance d is expected to be small, i.e. smaller than the decision threshold d_{max}. In that case, *CCF*(M') is equal to the original *CCV*(M) (because M' is equal to original M) and d is equal to the number of errors in CCV' only. The value of d_{max} should be determined 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:

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

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

In case of unsuccessful verification HD(CCV', *CCF*(M')) has a large value, which is above the decision threshold d_{max}. The reason is as follows: 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 2^n values with equal probability and the expected value of HD(CCV', *CCF*(M')) is equal to the expected value of HD between CCV' and any other fixed bit pattern of the same length. Therefore $pmf_2(d)$ is also a binomial function $B(n, p)$ but with $p = 0.5$, since every bit in CCV' is expected to be 0 or 1 with the same probability:

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

The choice of value of d_{max} which is used in the verification process should be the result of a compromise. Namely, if d_{max} is too small there is a large probability that some correctly decoded message M' won't be recognized as a correct one ($d > d_{max}$, verification fails) because of only "a few" errors in CCV'. This probability is called non-detection probability P_{nd}.

From the other side, if d_{max} is too large (e.g. near $n/2$) a "wrong detection" may happen. In this case some wrongly decoded message M' is taken as correct ($d \le d_{max}$, verification is successful although it shouldn't be) since the condition for the verification within TSID is "too loose". Both P_{nd} and the probability of wrong detection P_{wd} can be calculated using Equations (4)-(7) for different values of parameters m, n , BER and d_{max} where:

$$P_{nd} = P_{correct} \cdot \sum_{d=d_{\max}+1}^{n} pmf_1(d) \tag{8}$$

and:

$$P_{wd} \approx P_{wrong} \cdot \sum_{d=0}^{d_{max}} pmf_2(d) \tag{9}$$

Different criterions related to the tolerable levels of P_{nd} and P_{wd} can be set up and the proper value of d_{max} can be chosen so the criterions are satisfied. Nevertheless, numerous calculations showed that there is a wide area "in the middle", where the value of d_{max} can be taken from, whereby the probabilities P_{nd} and P_{wd} are "extremely small". For example, for parameter values m = 200 bits, n = 128 bits and BER = 0.036, any value of d_{max} between 14 and 28 will fulfill the condition that both P_{nd} and P_{wd} are less than 10^{-10}, and if d_{max} = 21 then both probabilities are even less than 10^{-35}.

Simulations have been performed using the same parameters as for those of SID (Chapter 2). The value of the decision threshold of TSID has been set to d_{max} = 8 (when P_{nd} is close to 10^{-4}). The results of simulations are also presented in Fig 3, showing a significant gain in comparison to channel decoding and Soft Input Decryption.

4. Conclusion

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 between 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 wrong detection on the other hand. Simulations show that a significant coding gain can be achieved by the use of the TSID algorithm. The 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. (1999). Approximate message authentication codes, in Proc. 3rd Annual Fedlab Symp. Advanced Telecommunications/Information Distribution, vol.1, College Park.
[3] Xie L, Arce G. R., & Graveman R. F. (2001). Approximate Image Message Authentication Codes, IEEE Trans. On Multimedia, vol.3, no.2.
[4] Boncelet C. G. Jr. (2006). The NTMAC for Authentication of Noisy Messages, IEEE Trans. On Information Forensics and Security, vol.1, no.1.
[5] Liu Y., & Boncelet C. G. Jr. (2005). The CRC-NTMAC for Noisy Message Authentication. IEEE Military Communication Conference, MILCOM.
[6] Ruland C., & Živić N. (2006). Soft Input Decryption, 4th Turbocode Conference, 6th Source and Channel Code Conference, VDE/IEEE, Munich.
[7] Chase D. (1972). A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, IEEE Trans. Inform. Theory, IT- 18, pp. 170-182.
[8] Forney G. D. Jr. (1966). Generalized Minimum Distance Decoding, IEEE Trans. Inform. Theory, IT-12, pp. 125-131.
[9] (2011). Information technology - Security techniques - Message Authentication Codes (MACs) - Part 2: Mechanisms using a hash-function, ISO/IEC 9797-2, 2nd edition waiting for publication.
[10] Lin S., & Costello D. J. (20047). Error Control Coding, Pearson Prentice Hall, USA.
[11] Bahl L., Jelinek J., Raviv J., & Raviv F. (1974). Optimal decoding of linear codes for minimizing symbol error rate, IEEE Transactions on Information Theory, IT-20, pp. 284-287.
[12] Hays H. M., & Tavares S. E. (1995). Avalanche characteristics of Substitution - Permutation Encryption Networks, IEEE Trans. On Computers, Vol. 44, Nr. 9.