2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks

Reliability of Soft Verification of Message Authentication Codes

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

Abstract— **It is known that Message Authentication Codes are extremely sensitive to any change of the message they are appended to. Even one or more bits of the changed message invert about 50% of bits of Message Authentication Codes, making in such a way the message useless. The hard condition for the successful verification of Message Authentication Codes is that all bits of the received Message Authentication Code and that one recalculated of the received message have to be equal. This condition for the successful verification of messages protected by Message Authentication Codes is not suitable in many applications. Therefore an introduction of a softer condition for the successful verification would enable the correction and improvement of the verification rate of messages corrupted by transmission over a noisy channel. The following paper is based on an algorithm for Soft Verification 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. A soft output of the algorithm gives information about the reduction of the security level of the successful authentication to the source decoder. This information can be compared to reliability values as soft output of SISO channel decoder.**

Keywords— ***Soft Output; Soft Input Soft Output; Message Authentication Codes; Hamming distance; Decision Threshold; Symmetric Cryptography***

I. INTRODUCTION

Message Authentication Codes (MACs) [1] are symmetric cryptographic algorithms, providing data integrity and the authentication of data origin. Data integrity enables the recognition of any modification or manipulation of the message during transmission. The authentication of data origin gives a confirmation that the message originates by the sender, who shares the used secret key with the receiver.

MACs are often used in communication systems, which demand secure message transfer. 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 is known in cryptography as "avalanche effect": every modified message produces an incorrect MAC at the verification. If the verification fails, the message is not authentic and it is regarded as useless.

This strong verification condition for message authentication is used as a protection against forgeries. Nevertheless, for many applications, like multimedia or voice transmission, where the digital content is continuously modified and manipulated as a result of compression and conversion, this strong verification condition is not suitable. Any of these modifications would be considered as a forgery in case of MAC verification. Therefore it would be suitable for such applications that the modifications of a single or a few message bits do not result in any modification of MAC.

There are a number of algorithms [2, 3, 4, 5] which 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. So, 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, which uses standard Message Authentication Codes, but with a different verification, was presented in [6, 7]. The received Message Authentication Code and the one recalculated of the received message are compared, as by regular verification, but they do not have to be equal for the successful verification. The verification is successful also, if one, two, or few bits of both compared Message Authentication Codes are different. We will call this algorithm Soft Input Soft Verification (SISV). It uses as a basis an algorithm of Soft Input Decryption [8]. Both algorithms are iterative and 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 analysis of the statistic characteristics of the threshold used in the SISV algorithm was detailed given in [9, 10]. Therefore problems referring definition and setting of the threshold of the SISV algorithm are only mentioned in this paper, without deeper explanations.

The aim of this paper is to solve the problem of decreased security as a consequence of the introduced soft verification. Therefore a new output of the SISV algorithm is presented under the name of Soft Output. This point is important for the introduction of the soft verification into the Joint Source-Verification-Channel Model of the communication system and, at the same time, a novelty

978-0-7695-5042-8/13 $26.00 © 2013 IEEE
DOI 10.1109/CICSYN.2013.19

given in this paper compared to the results published in [6]-[10].

II. Soft Input Soft Verification

Soft Input Soft Verification (SISV) is the algorithm of a receiver, which uses the sensitivity of cryptographic MACs for the improvement of the decoding results. The new verification process is introduced, so called soft verification, which is not as hard as the standard one [6]. SISV uses standard verification of MACs, whereby the verification is based on the condition, that the Hamming distance HD between the received cryptographic check value CCV' and recalculated cryptographic check value CCV” of the corrected message M”, CCV” = CCF (M"), whereby CCF stands for cryptographic check function (for example MAC function), has to be smaller than a defined threshold d_{max} [7, 8, 9, 10] (see Fig. 1).

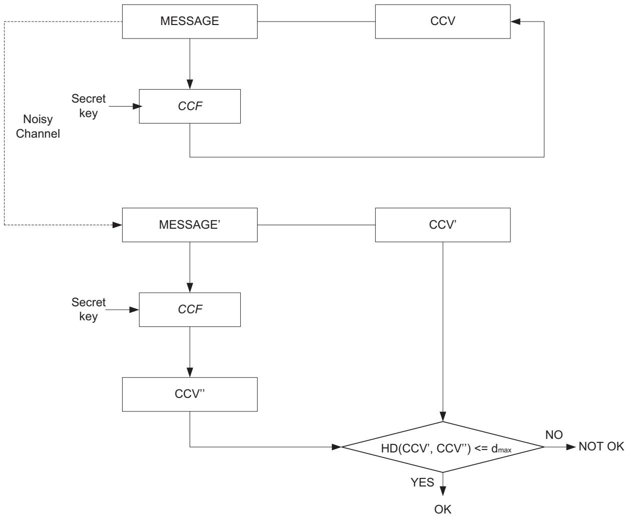

Figure 1. Soft Verification.

The background for the success of the algorithm is the “avalanche criterion” [11, 12] 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.

Soft Input Soft Verification works as follows:

Input: the received message M’, the received cryptographic check value CCV’, reliability values (LLR values) of the message and its cryptographic check value after SISO channel decoding, the maximal number of iterations i_{max}, threshold value d_{max}.

Output: the 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 step (v); else increment i, go to (iii);

(v) Output M” and CCV”.

The statistical distribution of d = HD (CCV’, CCF (M”)) was studied in [10], 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:

$$pmf(d) = pmf_1(d) \cdot P_{correct} + pmf_2(d) \cdot P_{wrong} \qquad (1)$$

where $P_{correct}$ and P_{wrong} are the probabilities that M’ does not contain errors, i.e. that M’ contains errors respectively:

$$P_{correct} = (1 - BER)^m \qquad (2)$$

$$P_{wrong} = 1 - (1 - BER)^m \qquad (3)$$

In case of the successful verification the Hamming distance d is expected to be small - smaller than the decision threshold d_{max}. In that case, CCF (M’) is equal to the original CCV of the sent message 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) [10]:

$$pmf_1(d) = \binom{n}{d} BER^d \cdot (1 - BER)^{n-d} \qquad (4)$$

with the mean value n·BER and the standard deviation σ^2 = n·BER·(1 - 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: 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 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 [10]:

$$pmf_2(d) = \binom{n}{d} \cdot \frac{1}{2^n} \tag{5}$$

III. Probabilty of a Wrong Decision

The wrong decision is an event, that the received message M' or a corrected message M" is a wrong one, although the condition for the successful verification is fulfilled: HD (CCV', CCF (M')) <= $d_{\max}$. This event can happen after receiving the message M' (before the iterations started), as well as after each iteration of the Soft Input Soft Verification algorithm. Since there are $i \le i_{max}$ iterations in the Soft Input Soft Verification algorithm, the number of iterations should be considered before calculating the probability of the wrong decision P_{wd}.

The iterations are performed according to the reliability of received bits (LLR-values), in order to find the correct message, i.e. the message which is believed to be correct. Also, there is an intention for the number of iterations to be minimised – not only because of the lower complexity, but also to avoid the unnecessary increase of the probability of the wrong decision.

It should be kept in mind that an attacker could affect the S/N of the received signal and thus the LLR-values as well, e.g. in case of the Man-in-the-middle attack or an additional noise. The attacker might be also taking care that a large number of iterations are performing, among which a wrong decision can happen. If maximally i_{max} message-MAC pairs are to be verified, the attacker can set up, by affecting the LLR-values, whose messages will be promoted to the candidates for (successful) verification. In this case, we are talking about a non verifiable forgery attack.

In the following text the increase of the wrong decision probability depending on the maximal number of possible iterations and on the threshold value d_{max} will be considered. The probability of a wrong correction using the algorithm is also considered for the „worst case“, i.e. when a solution (successful verification) is found in the last possible iteration and with the maximal allowed Hamming distance.

The probability that the verification result of a changed message is TRUE is $1/2^n$. With the standard verification process this false affirmative result can happen only once.

With the Soft Verification there are not only a lot of iterations, but also the difference between check values by at most d_{max} bits is tolerated.

Two cases are separately observed: the case of the first verification, where the received MAC is compared with the new calculated MAC, and the second – the verifications during the iterative correction process:

1. First verification

The wrong decision can only happen if the message M differs from message M' and if $d = HD$(CCV', CCV'') $\le d_{max}$:

$$P_{wd} = (1-(1-BER)^m)\sum_{d=0}^{d_{\max}} \frac{\binom{n}{d}}{2^n} \le \sum_{d=0}^{d_{\max}} \frac{\binom{n}{d}}{2^n} \tag{6}$$

2. and each following verification i (2,… i_{max}), (iterations 1, …,i_{max} - 1) - a wrong decision can happen if the message M' differs from the message M'' and if $d = HD$(CCV', CCV'') $\le d_{max}$:

$$P_{wd} = \frac{1}{2^n}\sum_{d=0}^{d_{\max}}\binom{n}{d} \tag{7}$$

Since after (i_{max} – 1) attempts of verification was no wrong decision, the probability of a wrong decision after i_{max} verifications is:

$$P_{wd} = a \cdot \sum_{i=0}^{i_{\max}-1}(1-a)^i \tag{8}$$

where:

$$a = \sum_{d=0}^{d_{\max}} \frac{\binom{n}{d}}{2^n} \tag{9}$$

After the queue expansion:

$$P_{wd} = 1-(1-a)^{i_{\max}} \tag{10}$$

If the binomial expansion is applied in (10) and the higher expansion members which tend to zero are neglected, then the following approximation for P_{wd} is valid:

$$P_{wd}(n, d_{\max}, i_{\max}) \approx i_{\max} \cdot a = i_{\max} \cdot \frac{\sum_{d=0}^{d_{\max}}\binom{n}{d}}{2^n} \tag{11}$$

It's obvious that:

$$P_{wd} \ge \frac{1}{2^n} \tag{12}$$

Therefore, security level is reduced. For the compensation of the wrong decision probability increase, the length n of the CCV should be enlarged to n$_{new}$, so that the probability of a wrong decision is less than or equal to the same probability in case of standard verification:

$$P_{wd} \leq \frac{1}{2^n} \tag{13}$$

$$\frac{1}{P_{wd}} = \frac{1}{P_{wd}(n_2, d_{\max}, i_{\max})} \geq 2^n \tag{14}$$

IV. Simulation Results

In order to value the results of Soft Verification in an objective way, the coding gain should be observed, in case of keeping the same level of the forgery complexity even with a CCV length of n$_{new}$.

The parameters of the simulations are as follows:

- The length of M is 192 and of CCV 128 bits
- As CCF the H_MAC is used
- Convolutional 1/2 (7, 5) encoder
- BPSK modulation with Soft Decision
- MAP decoding algorithm for convolutional
- Maximal number of iterations $i_{max} = 2^8$ i.e. $i_{max} = 2^{16}$
- d_{max} is defined to minimize the probability of a wrong decision (see [6, 10])
- 50 000 simulations for each curve point in graphics

As the measure of the correction and verification efficiency a parameter named Cryptographic Check Error Rate (CCER) is defined:

$$CCER = \frac{Num berof\ nonverified\, m essages}{Num berof\ received m essages} \tag{15}$$

Each of following figures shows CCER of the standard communication system using hard verification, i.e. CCER of standard hard verification according to ISO-Standard (curve a)), CCER of SISV algorithm with the standard length n of CCV (curve b), see [10]) and CCER of SISV algorithm with the extended length n$_{new}$ of CCV (curve c)).

By the calculation of the values of curve d) for each E$_b$/N$_0$ different values of d$_{max}$ are used (see [6, 9, 10]. For low E$_b$/N$_{0,}$ a greater value of d$_{max}$ is needed for correction and, on the other hand, a greater increase of the CCV length causes a lowering of coding gain. For higher E$_b$/N$_0$ the coding is smaller. Despite the loss of coding gain due to elongation of CCV, a significant coding gain still remains thanks to Soft Verification.

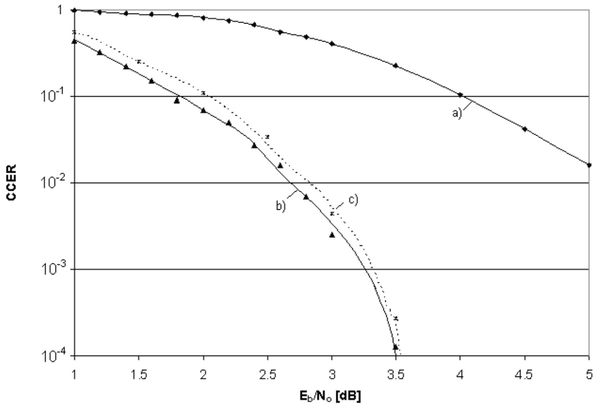

Figure 2. CCER for *n* = 128 and *m* = 192.

V. Soft Output

In the case that n, d$_{max}$ and i$_{max}$ fulfill all security demands, like it is shown in Chapter 5, a complexity of a forgery attack is not reduced by Soft Verification. As that security level is the same in that case as in case of a standard or hard verification, a reliability value (or an LR-value) of ±∞ (for bits “0” and “1”) can be output as a Soft Output value for all bits of the verified message. And vice versa, for all bits of the message a Soft Output value of “0” can be output in the case that the message cannot be successfully verified.

Alternatively, the source decoder, as the next stage in the receiver after the verification module can receive reliability values, which the SISO channel decoder outputs to the verification module. Source decoder could be improved in the future in such a way, that it receives reliability values LR from the verification module, as well as from the SISO channel decoder (LR values are forwarded to the verification module and then from the verification module to the source decoder).

How the length n of CCV can be extended to n$_{new}$ in order to satisfy standard security requirements was shown in Chapter 3.

In case that the new i.e. extended length of n is chosen in such a way, that it is lower than n$_{new}$, the security risks are higher, but under control: the probability of a forgery attack can be calculated for each message.

The limes value of P$_{wd}$ is calculated for the (theoretically) infinite i$_{max}$ (if the length of CCV is also, theoretically, infinite) and is given by:

$$\lim_{i_{ma}x \to \infty} P_{wd} = 1 \tag{16}$$

This defines boundary values of the probability of a wrong decision:

$$\frac{1}{2^n} \le P_{wd} \le 1 \tag{17}$$

As i_{max} has always a finite value because of finite system recourses, it is: $P_{wd} < 1$.

In the case of a successful SISV, it is known in which verification (round of iterations) the message was corrected. We will call the number of that successful iteration $i_{success}$. And even more, the Hamming distance between the received CCV' and the recalculated CCV" is also known for the iteration round of a successful verification. We will call this Hamming distance $d_{success}$.

The probability of a wrong decision will be calculated in a round of a successful verification as:

$$P_{wd,success} = 1-(1-a)^{i_{success}} \tag{18}$$

with:

$$a = \sum_{d=0}^{d_{success}} \frac{\binom{n}{d}}{2^n} \tag{19}$$

Each bit of the message is wrong / forged with the probability $P_{wd,success}$ given in (18) and vice versa, each bit of the message is correct with the probability of 1 - $P_{wd,success}$.

Trustworthiness of the message can be defined as an opposite value of the probability of a wrong decision: each bit of the message is trustworthy with the probability of 1 - $P_{wd,success}$. This means that each bit u of the message has a Soft Output or reliability value RV(u). In the case of a minimal probability of a wrong decision ($1/2^n$), the message is maximally trustworthy and it should have the maximal RV. If P_{wd} ($d_{success}$, $i_{success}$) increases more than $1/2^n$, the value of RV should decrease. As long as the verification is successful, the probability $P_{wd} < 1$ (although the value P_{wd} comes very close to 1, if i_{max} is very high). Therefore, $|RV(u)| \neq 0$ in the case of a successful verification.

In case that all verification trials fail, i.e. the verification is not successful after i_{max} iterations, all of message bits u should have the reliability value RV(u) = 0.

Reliability values (LLR-values) in SISO channel decoding are for each bit u of the message M defined as:

$$L(u) = \ln\frac{P(u=1)}{P(u=0)} \tag{20}$$

and can get, theoretically, boundary values of +∞ (in case that "1" is sent, i.e. P(u = 1) = 1 and P(u = 0) = 0), and -∞ (in case that "0" is sent, i.e. P(u = 1) = 0 and P(u = 0) = 1). If we want to use reliability values RV in the way that reliability values in SISO channel decoding are used, they should be normalized in the following way:

RV(u) = 0 for $P_{wd,success} = 1$

RV(u) = +∞ for $P_{wd,success} = \frac{1}{2^n}$ (max trustworthiness of the hard verification) and u = 1

RV(u) = -∞ for $P_{wd,success} = \frac{1}{2^n}$ (max trustworthiness of the hard verification) and u = 0.

The following equation fulfills all previous conditions:

$$RV(u) = \operatorname{sign}(0.5-u)\cdot \ln\frac{(P_{wd,success}-\frac{1}{2^n})(P_{wd,success}+\frac{1}{2^n})}{1-\frac{1}{2^{2n}}} \tag{21}$$

or, in a simpler form:

$$RV(u) = \operatorname{sign}(0.5-u)\cdot \ln\frac{P_{wd,success}{}^2-\frac{1}{2^{2n}}}{1-\frac{1}{2^{2n}}} \tag{22}$$

These reliability values RV have a role of reliability values (LLR-values). SISV algorithm outputs reliability values as a Soft Output to the source decoder. If the source decoder wants to feedback reliability values, than RV values will be fed back to the verification module and from the verification module to the SISO channel decoder. The architectural model of the communication system, which supports transparency of the verification module, will be integrated in a Joint Source-Channel Model in this way (see Fig. 3).

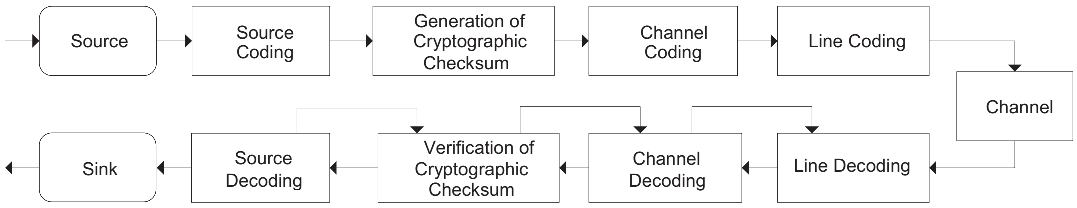

Figure 3. Communication model with feedbacks.

VI. Influence of System Parameters on Soft Output

Fig. 4 and 5 show Soft Output values (reliability values RV) by two examples if $i_{success}$. $d_{success}$, $i_{success}$ and n are parameters which show how trustworthy a message is after Soft Verification. Therefore, the dependency of Soft Output on $d_{success}$, $i_{success}$ and n is shown.

Fig. 6 shows dependence of Soft Output values on $d_{success}$ and $i_{success}$ on example for one specific value of n_{new} = 163 (this is the extended length of n = 128 for d_{max} = 4).

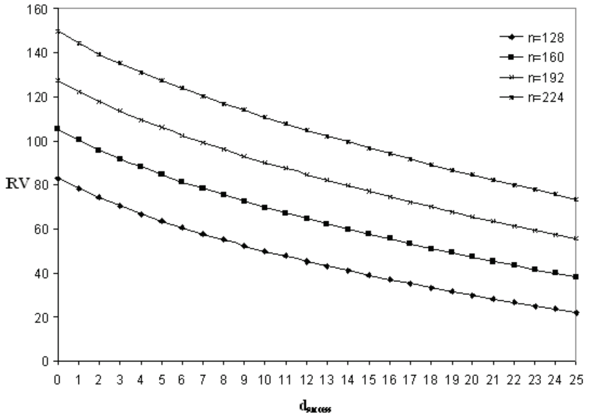

Figure 3. Soft Output for $i_{success}$ = 288 for different n.

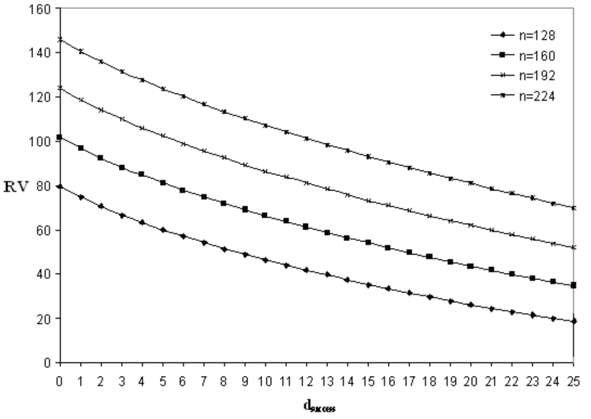

Figure 4. Soft Output for $i_{success}$ = 10.000 for different n.

VII. Conclusion

The presented paper concentrates on the algorithm of Soft Input Soft Verification, which not only verifies but also corrects a corrupted message protected with a MAC. SISV is an iterative algorithm based on bit flipping correction and Soft Verification. Soft Verification uses Hamming distance between CCVs (MACs) for defining the criterion and the fulfillment of the criterion of the successful and unsuccessful CCV verification. Security criterions are not fulfilled anymore, because iterations and a new type of verification are used. Therefore, neutralization of the lost security level by the extension of the CCV length has been applied. The lost of security is measured by Soft Output or reliability values at the output of the verification.

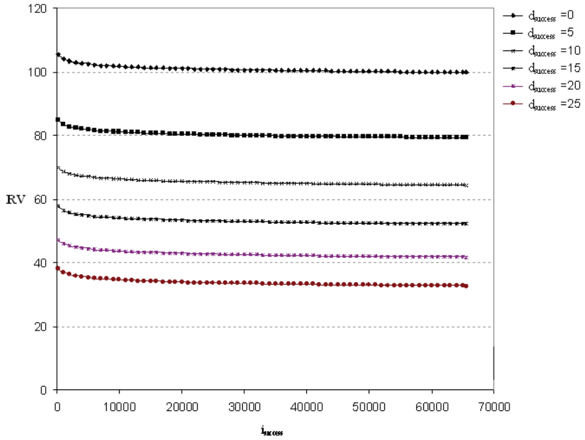

Figure 5. Soft Output for n_{new} = 163 and different values $d_{success}$.

References

[1] ISO/IEC 9797-1 Std., *Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher*, ISO, Geneva, Switzerland, 2011.

[2] R. F. Graveman, K. E. Fu., "Approximate message authentication codes", *Proc. 3rd Annual Fedlab Symp. Advanced Telecommunications/Information Distribution*, vol.1, College Park, MD, 1999.

[3] L. Xie, G. R. Arce, R. F. Graveman, "Approximate Image Message Authentication Codes", *IEEE Trans. On Multimedia*, vol.3, no.2, Jun 2001, p. 242-252.

[4] C. G. Jr. Boncelet, "The NTMAC for Authentication of Noisy Messages", *IEEE Trans. On Information Forensics and Security*, vol.1, no.1,, March 2006, p. 35-42.

[5] Y. Liu, C. G. Jr. Boncelet, "The CRC-NTMAC for Noisy Message Authentication", *IEEE Trans. On Forensics and Security*, vol. 1, no. 4, Dec. 2006., p. 517-523.

[6] N. Zivic, "Soft correction and verification of the messages protected by cryptographic check values", *Proc. 45th Annual Conference on Information Sciences and Systems*, p. 1-6, March 2011.

[7] N. Zivic, M. Flanagan, "On Joint Cryptographic Verification and Channel Decoding via the Maximum Likelihood Criterion", *IEEE Comm. Letters*, vol. 16, no.5, p. 717-719, May 2012.

[8] N. Zivic, C. Ruland, *Method for Transmitting and Receiving a Data Block and a corresponding transmitter and receiver*, Nr. 8,196, 015, US Patent, Patent and Trademark Office, Washington D.C., 2012

[9] N. Zivic: "Introducing Soft Verification as an Improvement of Hard Verification of Cryptographic Check Values", *Pro.c 35th IEEE Sarnoff Symposium*, N J, USA, p. 1-6, May 2012.

[10] N. Zivic "Soft Verification of Message Authentication Codes", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, vol. 3, issue 1, May 2012, p. 262-285

[11] H. M. Hays, S. E. Tavares, "Avalanche characteristics of Substitution - Permutation Encryption Networks", *IEEE Trans. On Computers*, Vol. 44, Nr. 9, p. 1131-1139, September 1995.

[12] F. S. Gomez, R. J. J. Andina, E. Mandado, "Concurrent Error Detection in Block Ciphers", *Proc. IEEE Int. Test Conf.*, Atlantic City, NJ, p. 979-984,2000.