Input Output Chaining (IOC) AE Mode Revisited

January 2014 -
Francisco Recacha[1] -
e-mail: frecacha@gmail.com -

ABSTRACT: *Input Output Chaining (IOC) is an authenticated encryption (AE) mode that can be used with any block cipher. IOC main interest is that each message block is ciphered just once, as when only implementing confidentiality, while the added complexity by the accompanying integrity service is negligible. The core integrity concept in IOC is based on a novel, minimal and appealing chaining mechanism already presented by the first published IOC mode proposal* [6], *which so far has resisted public scrutiny. This paper upgrades some details of the former IOC specification and its security demonstration concluding that whatever resources could be spent to forge IOC integrity mechanism, assuming an ideal block cipher, the success probability of such attack will not be higher than* $2^{-(n-1.25)}$, *where* n *is the cipher block size.*

1. Introduction

The author proposed in 1996 an AE mode called IOBC [1], [4], [6] as part of his PhD Thesis but the author changed jobs then from the academic field to the private sector and IOBC received a very limited dissemination. Zúquete and Guedes proposed later a supposedly enhancement of IOBC called EPBC [2] that received much broader dissemination and Mitchell published a cryptanalysis of EPBC in 2007 showing that the integrity offered by EPBC was in fact quite easily broken [3]. This cryptanalysis arrived to the knowledge of the IOBC author several years later and he contacted to Mitchell who, in February 2013, performed an in-depth IOBC cryptanalysis that was presented in ACISP 2013 [5] concluding that IOBC was a little bit weaker than as estimated by the author in his PhD thesis. The author proposed as response to Mitchell conclusions a new AE mode called IOC based on IOBC principles but much stronger and more lightweight than its parent mode. IOC was then published on NIST encryption modes development web page for public scrutiny [6] and although several issues have been raised by the specialized community since then (see web link in [4]), the core integrity mechanism proposed by IOC proposal has resisted so far all the critical reviews known by the author.

This paper, where most of the contents from [6] have been replicated for completeness, comes as a revision of IOC former specification given in [6] that has been made upon the feedbacks received from its public review. The main characteristics of this IOC mode revision are:

- The same chaining method is maintained: an appealingly simple and novel chaining of x-or and modular additions is the basis for its lightweight integrity mechanism;
- The message Modification Detection Code (MDC) generation is simplified in a more elegant way making feasible a simpler analytical characterization to be used in formal security demonstrations, specially with cipher algorithms which block size is different from the key size;
- A new method to generate fresh IVs when needed is included now as part of IOC mode specification, avoiding the flaw identified in the method just recommended as a guideline for IOC or similar ones that could made by wrong implementations;
- This IOC specification is accompanied by a formal security demonstration indicating that, if an ideal block cipher is assumed, whatever resources an attacker could spend to forge IOC integrity mechanism the success probability of such attack will not be higher than $2^{-(n-1.25)}$, being n the block size[2].

To finalize with this introduction, the author wants to thank all the comments received on the initial IOC proposal [6]. They made possible to consolidate this new IOC mode specification. Moreover, special thanks also to NIST encryption modes development team for publishing IOC on their webpage which provided me with the personal motivation to spend a part of my spare time to go on with this work.

[1] To the knowledge of the author no IPRs apply to IOC mode at any region of the world. IOC can be freely used without - any restriction imposed by the author except to be fairly credited. -

[2] This demonstration is based on an exhaustive analysis of all possible attacks but is not alligned, at least regarding used - notation, with current trends in the provable security field. Nonetheless, a demonstration adopting a more 'canonical' - approach in the 'provable security' field would be of much interest. The author kindly invites to who may be interested - to produce such a demonstration. -

2. The IOC Authenticated Encryption Mode

2.1 Mode Specification

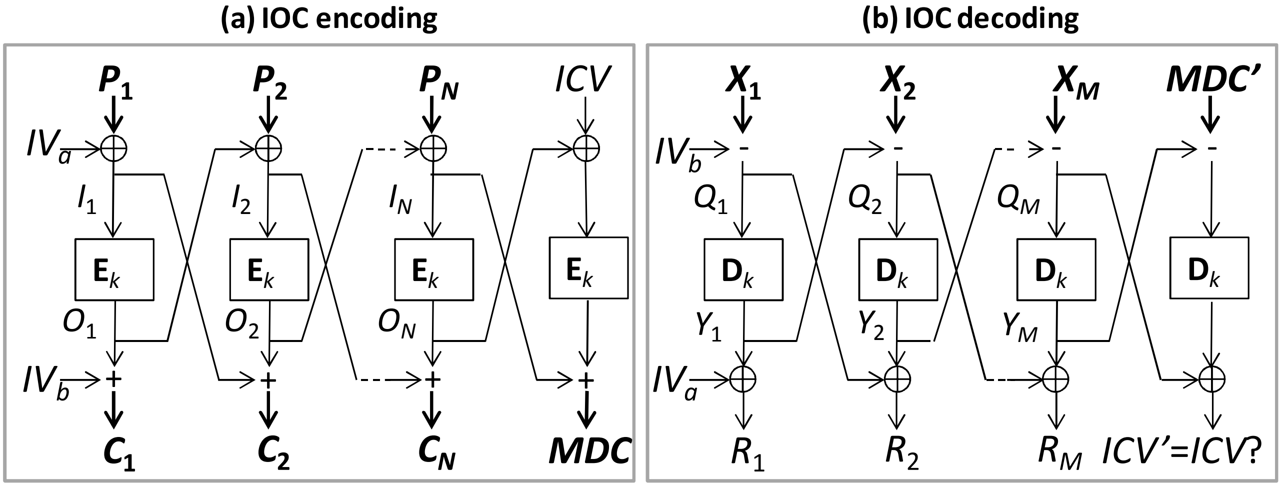


Figure 2: IOC authenticated encryption mode

Figure 2 illustrates IOC encoding operation mode that can formally be defined as it follows:

$$\left.\begin{aligned} C_i &= O_i + I_{i-1} \\ O_i &= E_k\{I_i\} \\ I_i &= P_i \oplus O_{i-1} \end{aligned}\right\} \text{ for } i = 1, \dots, N+1; \qquad (1)$$

where

a) P_i is a plaintext block of n bits of the plain message[3] and C_i its corresponding cipher-text block; -
b) N is the length, in n-bit blocks, of the plain message; -
c) $O_0 = IV_a$ is a secret and random n-bit value changed for each message; -
d) $I_0 = IV_b$ is a secret and random n-bit value changed for each message and different from IV_a;
e) P_{N+1}=ICV (Integrity Check Vector) is a secret random n-bit value changed for each message; -
f) C_{N+1}=MDC (Modification Detection Code) is an cryptogram authentication n-bit tag; -
g) $E_k\{ X \}$ is the result of the block encryption of a n bit vector X, using the key k; -
h) $\oplus$ is the x-or binary operator applied bit by bit to the two input n-bit vectors; -
i) + is the regular arithmetic addition modulo 2^n; -

Analogously, the IOC decoding operation is defined as follows:

$$\left.\begin{aligned} R_i &= Y_i \oplus Q_{i-1} \\ Y_i &= D_k\{Q_i\} \\ Q_i &= X_i - Y_{i-1} \end{aligned}\right\} \text{ for } i = 1, \dots, M+1; \qquad (2)$$

where

a) $Q_o = IV_a$ and $Y_o = IV_b$ as in the encryption procedure;
b) $D_k\{\}$, the inverse operator of $E_k\{\}$ (i.e. $D_k\{ E_k\{X\} \} = X$); -
c) the decoded plain message is accepted as authentic only if $ICV' = R_{M+1}= ICV$; -
d) - is the regular arithmetic subtraction modulo 2^n. -

It is immediate the decoding operation for any authentic cryptogram is just the inverse of the encoding one (i.e. $R_i = P_i$ for i=1 … N and $ICV' = ICV$, with M=N).

[3] If the length of the plain-text message is not multiple of n, then additional padding bits shall be added till the last block, P_N, is completed.

2.2 Specification of Operational Parameters

The Initialization Vectors IV_a and IV_b and the Integrity Check Vector ICV needed to process each message shall comply the following operational requirements: (3)

1. - IV_a, IV_b and ICV shall be shared between the sender and the receiver and shall be secret;
2. - IV_a, IV_b and ICV shall be random, or pseudorandom and probabilistically different among them;
3. - IV_a, IV_b and ICV shall be renewed for each one of the messages;
4. - IV_a, IV_b and ICV generation shall avoid any weak composition with IOC chaining logics.

Given the above requirements, the following specifications are part of IOC mode to manage ICV, IV_a and IV_b along a *security session*[4] (see figure 3): (4)

a) For the first message of a security session a "fresh" pair of IV_a and IV_b is computed as $IV_a = E_{k'}\{ 0 \}$ and $IV_b = E_{k'}\{ IV_a \}$, where $k' = k + S$ (modular addition with the smallest value of $2^{|k|}$ and 2^n);

b) For subsequent messages, S is incremented and IVs can be taken simply as the last inner vectors, O_{N+1} and I_{N+1} respectively, from the previous message (i.e. $IV_{a,S+1} = O_{N+1,S}$ and $IV_{b,S+1} = I_{N+1,S}$);

c) - Alternatively to method (b), IV_a and IV_b can be reset with "fresh" values at any particular message using the method specified in bullet (a)). This alternative method can be automatically triggered once a particular total number of messages or total data volume is surpassed, or can be forced by means of the security session control signaling protocol. In this last case, this IVs reset may help, for instance, for resynchronization in case of message losses in applications tolerant to data-loss;

d) - For each message, ICV is computed as $ICV = (S \oplus IV_a) + (N \oplus IV_b)$;

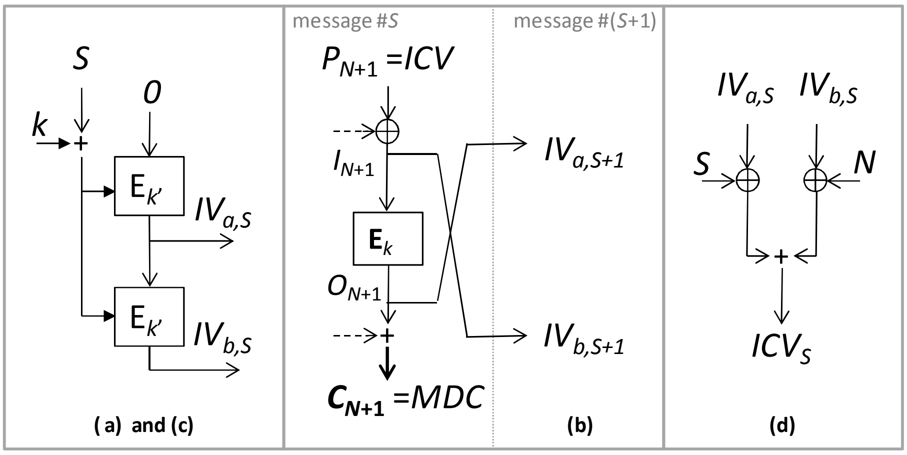


Figure 3: Generation of IOC Initializing Vectors and Integrity Check Vector

Regarding the message counter, S, the applicable operational requirements for this IOC parameter are:

4. - Each time a security session is initiated the message counter S shall be reset to 1 and a new cipher key, k, shall be set between the sender and the receiver. Note: value 0 shall not be used for S;
5. - S value shall be incremented synchronously for each message both by the sender and the receiver;
6. - S value shall be exchanged between the sender and the receiver for resynchronization at least each time a new pair of 'fresh' IVs is to be established;
7. - A security session can be terminated at any moment (e.g. when reaching an specific data volume threshold) but in any case a security session shall not take more than min{ 2^n-1, $2^{|k|}$-1} messages in order $k'=k+S$ is not repeated during that session (thence, neither the ICV value nor fresh IVs).

Finally, plaintext message padding shall be implemented in accordance to NIST recommendation as specified in Annex A of [7]: a single '1' bit shall be appended to the last plaintext bit and as many as required '0' bits will be appended, if necessary, to complete the last block P_N.

To sum up, IOC uses S as a nonce from which all the supplementary and internal keying material (IVs and ICV) are derived from the secret cipher key, k, and bit padding produced as per above specs.

[4] A *security session* is defined as the chain of plain messages that are encoded using a same ciphering *session* key k and numbered each one with a particular sequence number $1 \leq S \leq \min\{ 2^n-1, 2^{|k|}-1\}$.

3. Some Observations on IOC Specification

3.1 Confidentiality and Algebraic Characterization of the Inner Vectors

Before characterizing the security of IOC mode, this section is aimed at establishing: (a) an analytical model; and (b) the relevant IOC mode properties that are later used as basis for that security characterization.

First of all, if A and B are two random numbers of n bits, it is a well known fact from Number Theory that both ($A \oplus B$) and (A+B) operators maintain the maximum randomness exhibited by A or B. From that, if we assume the initializing vectors IV_a and IV_b are random and secret, it can be easily demonstrated that all the I_i and O_i vectors are also random and secret even in the case a potential attacker knows all the plain and cipher blocks. Let's see why. Assuming the operator $E_k\{\}$ is a 'perfect' block cipher algorithm (i.e. a perfect secret randomizer) and it does not provide any useful knowledge to the attacker about the deterministic relation between any couple of input and output vectors (I_i, O_i) better than a brute force analysis could provide, then, the main information from IOC definition available to the attacker are equations (1) rewritten as it follows:

$$\left.\begin{aligned} C_i &= O_i + I_{i-1} \\ P_i &= I_i \oplus O_{i-1} \end{aligned}\right\}, \text{ for } i = 1, \ldots, N+1 \qquad (5)$$

Complemented with: -

$$O_i = E_k\{I_i\}, \text{for } i = 1,\ldots, N+1; \qquad (6) \text{ -}$$

and with the generation methods for ICV, IVs, S and padding as specified in section 2.2.

In the above equations, the simultaneous use of the two types of sum operators (the conventional arithmetic addition and the x-or one) introduces some burden in order to get an easily intelligible characterization for a clear analysis. Thus, let's try to rewrite it in a more 'comfortable' but equivalent form. For that purpose we will rewrite addition modulo-2^n operations in terms of regular x-or ones:

$$A + B = A \oplus B \oplus \Delta(A,B)\,; \qquad (7)$$

where $\Delta(A,B) = (A \oplus B) \oplus (A + B)$ is the n-bit difference vector containing the up to (n-1) bit carries that can appear for the most (n-1) significant bits at the arithmetic sum. Observe, in particular, that the less significant bit of $\Delta(A,B)$ is 0 in all cases and for other positions 0 and 1 values will not follow a uniform probability since the accumulated propagation of carries making that some values of $\Delta(A,B)$ more probable than others[5]. Using (7), equations (5) can be rewritten now in a more intelligible and manageable form:

$$\left.\begin{aligned} C_i &= O_i \oplus I_{i-1} \oplus \Delta(O_i, I_{i-1}) \\ P_i &= I_i \oplus O_{i-1} \\ \Delta(O_i, I_{i-1}) &= (O_i \oplus I_{i-1}) \oplus (O_i + I_{i-1}) \end{aligned}\right\}, \text{ for } i = 1, \ldots, N+1.$$

Or in a more compact writing:

$$\left.\begin{aligned} C_i &= O_i \oplus I_{i-1} \oplus \Delta_i \\ P_i &= I_i \oplus O_{i-1} \end{aligned}\right\} \text{ for } i = 1, \ldots, N+1. \qquad (8)$$

where $\Delta_i = \Delta(O_i, I_{i-1}) = (O_i \oplus I_{i-1}) \oplus (O_i + I_{i-1})$ are the bit carry-delta vectors of the addition modulo-2^n of I_i and O_{i-1}, that differentiates the result of the modulo-2^n addition with respect to the bit-wise x-or addition.

Assuming that all the plaintext and cipher blocks are known by an attacker (in fact, P_{N+1}=ICV, could be assumed unknown to the attacker but we do need this assumption by the moment), it is immediate to show that (8) is an indeterminate system of 2N+2 independent linear equations and 3N+5 unknown terms (I_0= IV_a, I_1, …, I_{N+1}, O_0= IV_b, O_1, …, O_{N+1}, and Δ_1, Δ_2, …, Δ_N, Δ_{N+1}) where no solution exists for any of the unknown terms[6]. Therefore, it can be finally concluded that, provided IV_a and IV_b are random and secret, a potential attacker cannot determine any value neither for the I_i and O_i inner vectors, neither for the Δ_i ones, even in the case she/he knows the whole plain message and its corresponding cryptogram. Hence, if any of the plaintext blocks is unknown there will be no way to gain access to it.

[5] These seem to be bad news for IOC mode but, as shown later, it is not the case. -

[6] In any case, a complete algebraic IOC model is formed by equations (8) together with the cryptographic relations (6). -

A very useful alternative writing of equations (8) is its corresponding matrix form (9) presented below. For instance, this matrix expression makes immediately evident the independent nature of the equations since each row contains at least one '1' that stands alone in its corresponding row. It also makes really simple the derivation of the rest of the conclusions stated above, and shows itself as a very intuitively tool for IOC insides comprehension and analysis.

$$\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}
\times
\begin{bmatrix}
I_0 = IV_a \\ O_0 = IV_b \\ \Delta_1 \\ I_1 \\ O_1 \\ \Delta_2 \\ I_2 \\ O_2 \\ \Delta_3 \\ I_3 \\ O_3 \\ \Delta_4 \\ I_4 \\ O_4 \\ \ldots \\ \ldots \\ \ldots \\ \Delta_{N-2} \\ I_{N-2} \\ O_{N-2} \\ \Delta_{N-1} \\ I_{N-1} \\ O_{N-1} \\ \Delta_N \\ I_N \\ O_N \\ \Delta_{N+1} \\ I_{N+1} \\ O_{N+1}
\end{bmatrix}
=
\begin{bmatrix}
C_1 \\ P_1 \\ C_2 \\ P_2 \\ C_3 \\ P_3 \\ C_4 \\ P_4 \\ C_5 \\ P_5 \\ \ldots \\ \ldots \\ \ldots \\ C_{N-1} \\ P_{N-1} \\ C_N \\ P_N \\ C_{N+1} = MDC \\ P_N = ICV
\end{bmatrix}
\tag{9}$$

3.2 Information Entropy in the Carry-Delta Vectors, or Δ_i Randomness

Provided the encryption algorithm can be considered a perfect randomizer and that the initializing vectors are produced by an equivalent method (as the (4.a) method proposed as IOC guideline to generate 'fresh' IVs), then it is straightforward that the inner vectors I_i and O_i for both IOC encoding and decoding processes will be perfect random n-bit vectors in the sense that they conserve the maximal information entropy exhibited by the IVs and the one generated by the encryption algorithm. Nonetheless, and as already mentioned, the carry-delta vectors, Δ_i, do not exhibit the same property since for instance their less significant bit will be always 0 and for subsequent ones the probability distribution for 0 and 1 values depend on the position of each specific bit. Therefore, it is evident from the beginning that such Δ_i could, in principle, constitute a weak aspect of IOC upon which a forgery attack could be designed if this entropy was too much low.

As presented in [1], [4] and [5], in IOBC AE mode, the equivalent equations to (8) for IOC, provide the attacker with a mean to try to build fake cryptogram blocks adding a series of known plain and cipher text blocks to enforce that such false cryptogram block equals the sum of a particular couple of inner vectors, $O_j \oplus I_{i-1}$, making possible in that case to substitute from the position i-th the authentic cryptogram blocks C_i, C_{i+2}, C_{i+2}, … by other ones $X_i \neq C_i$, $X_{i+1}=C_{j+1}$, $X_{i+2}=C_{j+2}$, … that would elude IOBC integrity mechanism.

Keeping that fact in mind, one can intuitively guess at this point that IOC integrity strength will be based, on its turn, on the entropy characteristics of these carry-delta vectors, Δ_i since no combination of the rows of the matrix equation (9) makes feasible to get rid of them[7] in order to enforce any particular Q_i vector in the decoding process to replicate some non-authentic O_j value, as will be shown later. That is, if the Δ_i vectors exhibit small entropy, then it could be relatively easy to guess the equivalent value of such $O_j \oplus I_{i-1}$ sums with a significant probability and to come out successfully with a forgery attack. On the contrary, if they exhibit high information entropy, then it will be practically impossible to implement such forgery attack. Then, let's have a look on which is actually the information entropy contained in those Δ_i.

[7] That statement is based in the fact that each Δ_i appears at most once at any column.

Let A and B two maximum entropy random n-bit binary numbers. As already mentioned, it is a well known fact from Number Theory that both $(A \oplus B)$ and $(A + B)$ maintain the entropy / randomness of A and B. Now, let define Δ as:
$\Delta = \Delta(A, B) = (A \oplus B) \oplus (A + B)$.

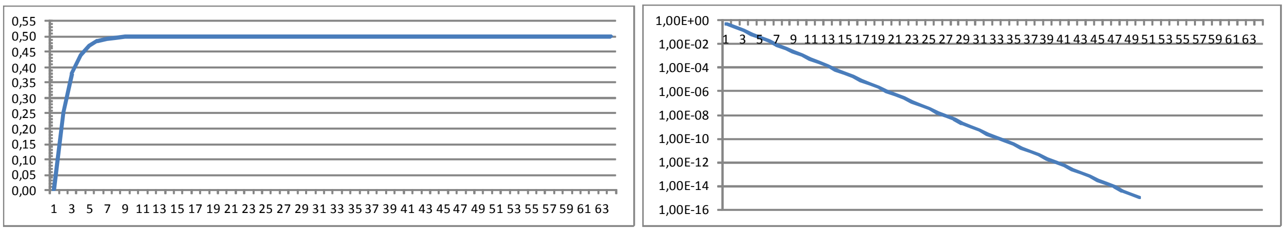
Then, the only difference between the bits in the i-th positions of $(A \oplus B)$ and $(A+B)$ comes from whether a carry from the bit addition in the (i-1)-th position has to be applied for the sum of the i-th position. Thus, it is immediate that the i-th bit of Δ will be 1 if such carry bit occurred and 0 otherwise. Let's see which is this probability for any of the bit positions $[\Delta]_i$. That is, which is the probability, $\mathrm{P}_i = \mathrm{P}\{[\Delta]_i = 1\}$, of having a bit carry from previous position at the addition modulo-2^n of A and B:

- $\mathrm{P}_1 = 0$, since being the first added bit position, no bit carry has to be applied from previous one;
- $\mathrm{P}_2 = \dfrac{1}{4}$, since only if the first two added bits were simultaneously 1 the bit carry is produced;
- $\mathrm{P}_3 = \dfrac{1}{4}(1 - \mathrm{P}_2) + \dfrac{3}{4}\mathrm{P}_2$, since if no carry was applied in the previous position then there would be a probability of ¼ that the bit sum in that position produces a carry and otherwise such probability would be ¾.
- and $\mathrm{P}_i = \dfrac{1}{4}(1 - \mathrm{P}_{i-1}) + \dfrac{3}{4}\mathrm{P}_{i-1} = \dfrac{1}{4} + \dfrac{\mathrm{P}_{i-1}}{2}$ for the general case.

Since P_i is a monotonously increasing sequence and, as a probability, it is bounded by 1 then it will converge for $i \to \infty$ towards a specific P value that will be given by:

$$\mathrm{P} = \frac{1}{4} + \frac{\mathrm{P}}{2} \quad \Rightarrow \quad \mathrm{P} = \frac{1}{2}.$$

Figure 4 below illustrates that the convergence of P_i towards the ½ is actually extremely fast. For instance, for the 10th bit, P_{10} approximates ½ just with an error of 10^{-3}. On other words, we can conclude that, except for a very few of the less significant bits, the carry-delta vectors exhibit very good entropy/randomness as the inner vectors do for all of their bits. Thus, let's quantify the total entropy contained in Δ_i vectors, or on other terms, the equivalent length of Δ_i in terms of random bits.



(a) (b)

Figure 4: (a) P_i vs i; (b) log (1/2- P_i) vs i

According to Shannon definition of Information Entropy, each bit of Δ_i exhibits an entropy defined by:
$\mathrm{H}([\Delta]_i) = -\mathrm{P}_i \cdot \log_2 \mathrm{P}_i - (1 - \mathrm{P}_i) \cdot \log_2 (1 - \mathrm{P}_i)$.

From where the total entropy for the whole carry-delta vector is:

$$\mathrm{H}(\Delta) = \sum_{i=1}^{n} \mathrm{H}([\Delta]_i) = \sum_{i=1}^{n} \left(-\mathrm{P}_i \cdot \log_2 \mathrm{P}_i - (1 - \mathrm{P}_i) \cdot \log_2 (1 - \mathrm{P}_i)\right). \tag{10}$$

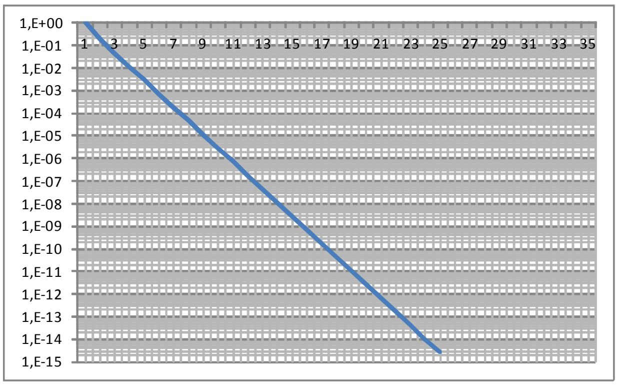


Figure 5: log(1 - H($[\Delta]_i$)) vs i

Figure 5 above shows that the entropy per bit is quickly maximized since it converges very rapidly towards 1 bit of information entropy per each 'physical' bit. For instance, for the 6-th bit of Δ its randomness entropy is above 1-10^{-3}. On the other hand, table 1 below indicates according to (10) the total entropy, H(Δ), for different values of n, that for n≥3 the total entropy is above (n-5/4). That is, almost equal to the block size for any practical n size (64, 128, 256, 512, 1024, 2048 ...) since only 1,25 equivalent bits do not exhibit any entropy. These are really good news for IOC since we can conclude that the carry delta vectors are almost completely random and, thus, unpredictable meanwhile one does not have access to the IOC inner vectors.

n **(physical bits)**	**1**	**2**	**3**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
H(Δ) (random bits)	0	0,811278	1,765712	2,75441	6,750667	14,75065	30,75065	62,7506	126,7506	254,7506

Table 1: Total entropy of the carry-delta vector, Δ, for different block sizes

4. IOC Confidentiality Strength

Let's see that even in the worst scenario where the attacker would know all the contents of a plain message (i.e. P_1, P_2, …, P_N), except for a particular block P_k (as it is the case for $P_{N+1} = ICV$), she/he will be unable to determine that P_k assuming the initializing vectors IV_a and IV_b are random and secret. If an attacker wants to determine a particular plain message block P_k, even knowing all the other blocks of the message, he will face the system of linear equations (8), or its equivalent and more intuitive matrix form (9), but now having P_k as an additional unknown variable and, therefore, the confidentiality of such plaintext block will completely safe.

It is worth to remark that, as other encryption chaining modes, IOC shows also as collateral advantage that it makes more difficult the cryptanalysis of the core encryption algorithm $E_k\{\}$ since a potential attacker will not be able to compile any dictionary of plain/ciphered blocks (X, $E_k\{X\}$) in order to try to determine the encryption key k or to take profit of such dictionary as it happens, for instance, with ECB encryption mode.

5. IOC Integrity Strength

5.1 Integrity Threats Taxonomy

Any attack against data integrity can be classified in one of the following three classes: data creation, removal or modification where the attacker actively modifies the message chain during a security session. Further to the above first level breakdown, the following sub-classes can be further identified in order to have a more detailed case taxonomy to be used as check-list for an exhaustive analysis:

- Creation / insertion:
  - Of a complete cryptogram. This attack would consist in the insertion of a cryptogram (or a sequence of them) between two authentic ones.
  - Partial insertion of some data block(s) within a specific cryptogram.
- Removal:
  - Of a complete cryptogram (or a sequence of them).
  - Removal of some data block(s) within a specific cryptogram.
- Modification:
  - Reordering of a sequence of complete cryptograms (without modifying them). This attack can be also considered either under the insertion class or the removal one, depending on whether the first out-of-sequence cryptogram is inserted or removed, respectively.
  - Reordering of some cipher-text blocks within a cryptogram. This case can also be handled under the insertion or removal cases.
  - **Modification of an authentic cryptogram using all available knowledge of authentic plain and cipher data blocks**. This is the most relevant case since no restrictions are assumed on what the attacker can do (except to gain knowledge of the key k and the IVs vectors). In this attack, the attacker modifies totally or partially a given cryptogram using any information that she/he can have at hand even in the worst case (i.e. assuming that all the plain and cipher text blocks are known, or that the plaintext can be even chosen / induced by the attacker).

5.2 Some Remarks on the Inner Vectors, *MDC* Block and Modulo-2^n Subtraction.

It is immediate from (8), or its equivalent matrix form (9), that each *I* and *O* inner vector can be expressed in terms of the plain and cipher blocks, the carry-delta and the initializing vectors as follows: (11)

- Even blocks ($1 < i \leq N+1$)
  - $I_i = \overset{i-2/2}{\underset{k=0}{\oplus}} \left(P_{i-2k} \oplus C_{i-(2k+1)} \oplus \Delta_{i-(2k+1)}\right) \oplus IV_b$
  - $O_i = \overset{i-2/2}{\underset{k=0}{\oplus}} \left(C_{i-2k} \oplus \Delta_{i-2k} \oplus P_{i-(2k+1)}\right) \oplus IV_a$
- Odd blocks ($1 \leq i \leq N+1$)
  - $I_i = P_i \oplus \left(\overset{i-3/2}{\underset{k=0}{\oplus}} \left(C_{i-(2k+1)} \oplus \Delta_{i-(2k+1)} \oplus P_{i-(2k+2)}\right)\right) \oplus IV_a$
  - $O_i = C_i \oplus \left(\overset{i-3/2}{\underset{k=0}{\oplus}} \left(P_{i-(2k+1)} \oplus C_{i-(2k+2)} \oplus \Delta_{i-(2k+2)}\right)\right) \oplus IV_b$

It is here worth to highlight from (11) that either in the even or the odd cases, I_i and O_i depend on complete separate and different subsets of the Δ, *C* and *P* blocks as well as only on one of the *IV*s. With this respect, it is also worth to point out, that the equations (11) can be extended to previous cryptograms as long as no fresh renewal of the *IV*s breaks the chaining with those previous cryptograms.

On the other hand, IOC integrity mechanism is based on the *ICV* vector separately computed by the sender and receiver according to $ICV = (S \oplus IV_a) + (N \oplus IV_b)$ and checked by the receiver in the last decrypted block.

Finally, just to mention that the modulo-2^n subtraction performed by the decoder could be rewritten, for analysis purposes, in terms of x-or sums as it follows: $Q_i = X_i - Y_{i-1} = X_i \oplus Y_{i-1} \oplus \Delta(Q_i, Y_{i-1})$.

5.3 Creation and Removal Attacks

This section analyses creation/insertion attacks either of complete or partial cryptograms where one, or several, arbitrary[8] cipher blocks are inserted in an authentic cryptogram (or, analogously, a complete arbitrary cryptogram, or several of them, is inserted in a sequence of authentic cryptograms). The case where the inserted cipher blocks / cryptograms are designed or selected by the attacker in function of all the information available is left for section 5.4.

Moreover, we also include removal attacks in this same section as a particular case of insertion of the first piece of authentic data not removed in the place of the removed one.

5.3.1 Insertion of a cryptogram, or a block within a cryptogram

It is completely straightforward that if an 'spurious' arbitrary cipher-block $C_i^{'}$ is inserted between two cipher-blocks of an authentic cryptogram, the attack will have only a success probability of 2^{-n} since it will produce completely random and unpredictable vectors $Q_i^{'}$, $Y_i^{'}$ and $R_i^{'}$ that will propagate uncontrolled over the subsequent decoding steps till producing at the end an also completely random and unpredictable vector *ICV**. In a similar manner, it is also completely straightforward that if a 'spurious' arbitrary cryptogram is inserted between the sender and receiver, the attack will have also only a success probability of 2^{-n}.

If instead a spurious cryptogram, the attacker inserts a complete authentic cryptogram (repeating some previous one or advancing the position of a subsequent one), then since, in general, the *IV*s of that cryptogram will not match the expected ones by the receiver, the insertion / reordering will be detected also with a probability of (1-2^{-n}). Nonetheless, it is directly derived from equation (11) that for a contiguous

[8] We understand here as 'arbitry' cipher data either any sinthetic or authentic cipher data selected whithout any special criteria (e.g. random/ spurious arbitrary values or authentic cipher blocks selected arbitrarily without analysing in advance which impact will they have on the decoding chain).

sequence of cryptograms[9], $\dot{C}_{i\text{-}1}$, $\dot{C}_i$, $\dot{C}_{i+1}$, …, $\dot{C}_{j\text{-}1}$, $\dot{C}_j$, if the total sums (from $\dot{C}_i$ to $\dot{C}_{j\text{-}1}$) of the corresponding plain, cipher and delta blocks are 0 then the *IV*s associated to $\dot{C}_i$ and $\dot{C}_{j+1}$ would be the same and, thus, if $\dot{C}_i$ is inserted after the cryptogram $\dot{C}_j$ (or the sequence from $\dot{C}_i$ to $\dot{C}_{j\text{-}1}$ is removed) then insertion would not be detected by IOC integrity mechanism (that's not actually right, since the sequence counter, *S*, used by the receiver would not match with the used in $\dot{C}_j$ for computing the *IVC* block, but at least the last inner vectors, and the *N* parameter used by the receiver would match with the used ones to compute that *ICV* and one could say, at least, that this forgery attack would be close to elude most of IOC integrity protections). Fortunately, although the attacker would know all those plain and cipher text blocks, the sums of all the involved carry-delta vectors, including the ones associated to the *ICV*, take a random value and for a given cryptogram sequence the probability of having such simultaneous combination in IV_a and IV_b , is not higher than $2^{-2(n\text{-}1,5)}$, since their respective values come from independent x-or sums of P_is, C_is, Δ_is and one of the *IV*s. Moreover, observe that although such coincidence on the IVs could happen, since the Δ_is are secret, the attacker has no mean to identify the event and cannot improve her/his chances by any computation and the attack would not improve the chances of a spurious modification with a success probability of 2^{-n}.

Finally, if the attacker inserts an arbitrary authentic cipher text block, then there are two possible cases to have into account: either (a) a copy of one authentic cipher block is inserted in another specific position, or (b) two or more blocks are exchanged reordering their positions in the cryptogram. In any case, if an arbitrary cipher-text block is inserted in another position of the cryptogram without using any other additional consideration, the combination of this cipher-text block with the previous *I* inner vector will result in a random input *Q* vector to the deciphering operator (see figure 2) causing an uncontrollable error propagation that will be detected by the *ICV* mechanism with a probability of $(1\text{-}2^{-n})$. Moreover, observe also that if the previous *I* and *O* vectors of the inserted cryptogram coincide with the ones that it 'finds' in the insertion position, then no error will appear and the attacker will be able just to append to this block the ones that follow it in its original position till the end of its corresponding cryptogram, the *ICV* block. But observe that these event is equivalent to the repetition of both Initializing Vectors, that is, it will happen with a probability not higher than $2^{-2(n\text{-}1,5)}$, it will be unnoticeable to the attacker thanks to the carry-delta vectors and in any case would not pass the integrity check due to the *N* parameter used for *ICV* computation.

5.3.2 Removal of a whole cryptogram, or a sequence of them

In this case, the first cryptogram arriving to the receiver will have, in principle, de-synchronized initializing vectors not matching with the expected ones. Thus it is also immediate that an attack of this type will have only a success probability of 2^{-n}. Observe also that the same considerations for the possible simultaneous coincidences of the two *IV*s already presented in 5.3.1 apply in this case.

5.3.3 Removal of some blocks from a given cryptogram

First, if the attacker removes the last block of a cryptogram, the *MDC*, the receiver will take C_N as the *MDC* and its corresponding computed *ICV** value will match *ICV* value again only with a probability of just 2^{-n}.

Second and final, if the attacker removes an intermediate arbitrary sequence of the cipher blocks $<C_i, C_{i+1}, …., C_j>$, with $(i < j)$, $(1 \le i < N)$ and $(1 < j \le N)$ and delivers to the receiver the resulting false cryptogram, then the situation when deciphering the new block $C_i^{'}$ is equivalent to the insertion cases in section 5.3.1, and therefore the attack will only success with a probability again of 2^{-n} (and also the same considerations about simultaneous coincidence of $(I_{i\text{-}1},O_{i\text{-}1})$ and (I_j,O_j), and the message length parameter, *N*, apply here).

5.4 'Intelligent' Modification Attacks

This section analyzes the most sophisticated attacks that can be designed inserting or modifying authentic cryptograms using all the information available to the attacker in the worst case (i.e. all the ciphered and plain message blocks, as well as IOC specification). To start with, it is required to highlight that in order any potential attack goes not beyond control, the attacker needs to be sure that each one of the inner *Q* vectors at the input of the decryption operator correspond with some O_j authentic inner vector associated to some known C_j cryptogram block. On the contrary, such value would produce a completely random value at the output of the $D_k\{\}$ operator that would propagate beyond any possible control and leading to a completely

[9] Ċ (a *C* with an dot over it) denotes a complete crytogram to differentiate with the notation used to denote specific cryptogram blocks.

uncontrollable error propagation towards the *ICV** value computed by the receiver. That is, **a necessary (although not sufficient) condition to build any forgery attack is that each input vector at the decipher operator has to correspond to someone obtainable at some step for an authentic cryptogram**.

In summary, the attacker needs to somehow enforce the above necessary condition. This could be tried to be implemented by two different ways: either (a) injecting a synthetic cipher block that makes that the input to the deciphering algorithm is an inner vector of a known authentic cipher block; or (b) taking benefit of any intrinsic repetition in the inner vectors that may happen eventually as consequence of IOC specification:

a) Synthetic injection of an O_j value defining a false $C_i^{'}$ from knowledge of authentic data;

b) 'Natural' injection of an O_j value taking profit of eventual repetitions in the inner vectors:
b.1) Eventual repetitions of the inner vectors used to substitute the subsequent blocks of the cryptogram by others taken from other authentic cryptogram.
b.2) Exploit known plain-text repetitions that have associated coinciding inner vectors because of the birthday paradox;

5.4.1 Synthetic injection of an O_j value in the *i*-th position of a cryptogram.

Let's see how the necessary condition of forcing a misplaced O_j could be tried generating a 'synthetic' $C_i^{'}$ using all the authentic material potentially known by the attacker and taking benefit from the linear nature of the equations (8), or its matrix equivalent form (9). The objective is to build a fake cryptogram, *C'*, which blocks from the *i*-th (for some 1 ≤ *i* ≤ *N*) till the *MDC* block are somehow defined in order to elude the integrity verification mechanism:

$$C^{'} = \left\langle C_1, C_2, ..., C_{i-1}, C_i^{'}, C_{i+1}^{'}, ... C_N^{'}, MDC \right\rangle .$$

In order the above necessary condition is guaranteed, $C_i^{'}$ shall comply:

$$C_i^{'} = I_{i-1} \oplus \Delta_i \oplus O_j \text{; for some } j \neq i.$$

If such condition could be implemented, then only replicating C_{j+1}, C_{j+2}, … C_N and *MDC* after $C_i^{'}$ would lead to the same last inner vectors that were used by the sender to compute the *MDC*, avoiding uncontrolled error propagation[10]. But, let's see that there is no way the attacker synthesizes such $C_i^{'}$ with the available information.

By a simple and quick inspection of matrix equation (9) is immediate that if |*j* – *i*| is odd then it is not feasible to combine the equation rows to make appear only one I_{i-1} and one O_j in the final combined equation. Moreover, the only way to make that such condition happens, |*j* – *i*| shall be even and then it is very simple to compute such $C_i^{'}$ that only depend on a I_{i-1} and a O_j :

- If (*j* > *i*), then

$$C_i^{'} = \bigoplus_{k=0}^{\frac{(j-i)-2}{2}} \left(C_{2k+i} \oplus P_{(2k+1)+i}\right) \oplus C_j = \left(\bigoplus_{k=0}^{\frac{(j-i)-2}{2}} \Delta_{2k+i} \right) \oplus I_{i-1} \oplus \Delta_j \oplus O_j ;$$

- If (*i* > *j*), then

$$C_i^{'} = \bigoplus_{k=0}^{\frac{(i-j)-2}{2}} \left(P_{(2k+1)+j} \oplus C_{(2k+2)+j}\right) \oplus Pi = \left(\bigoplus_{k=0}^{\frac{(i-j)-2}{2}} \Delta_{(2k+2)+j} \right) \oplus I_{i-1} \oplus O_j ;$$

Observe that in the first case (*j* > *i*), if the attacker substitutes the false cipher block C_i by $C_i^{'}$, then the input value delivered by the receiver to de deciphering block at the step *i*-th will be

[10] Observe that this would not be enough to forge IOC integrity mecanism since the message length, *N*, intervenes in the ICV computation. Thus additional cryptogram modifications would be required to complete the attack.

$$Q_i = \left(\bigoplus_{k=0}^{\frac{(j-i)-2}{2}} \Delta_{2k+i} \right) \oplus O_j ;$$

Since the first term, the x-or sum of the carry-deltas will be cero with a probability smaller than $2^{-(n-1.25)}$, O_j will be present at the input of the cipher algorithm only with the same probability and, therefore, uncontrollable error propagation will be unavoidable in practice since the attacker cannot guess in any manner the value of linear combination of the carry-delta vectors.

5.4.2 'Natural' injection of an O_j value taking profit of eventual repetitions in the inner vectors

Eventual repetitions on the Inner Vectors

In section 5.3.1 is already pointed out that given for a specific cryptogram there's some probability that its *IV*s coincide with the ones of another cryptogram. This fact could be of some use to build a forgery attack (especially if the parameters *N* and *S* would not intervene in the *MDC* computation). Fortunately, such eventual simultaneous coincidences happen only with a probability of 2^{-2n}. But more important indeed, although they could happen, according to equation (11) there's no manner to the attacker to detect the event thanks to the fact that the evolution of the inner vectors (and, thus, the *IV*s) from one from one cryptogram to following ones is obfuscated by the introduction of the carry-delta vectors at each ciphering step.

The same situation happens with eventual repetitions of the inner vectors within a cryptogram (or in different cryptograms): although rare events, they will happen sometime or the other but they will be unnoticeable for the attacker thanks again to the obfuscation introduced by the carry-delta vectors.

Exploitation of Eventual repetitions in the inner vectors caused by repeated plaintext sequences

If the value of a specific plain-text block is repeated, the probability that the inner vectors coincide is negligible (2^{-n} again) and no significant information can be collected in order to build a forgery attack.

Nonetheless, there are realistic scenarios in practice where an specific piece of plain-text appears identically repeated, possibly induced by the same attacker as a form of chosen-plaintext attack, on many pairs of plaintext (P_i, P_{i+1}), (P_j, P_{j+1}), … In this case, the attacker would be facing a regular birthday paradox problem, where $2^{n/2}$ of such pairs would be sufficient to have a very significant probability that for a particular couple of such consecutive pairs, let's say (P_k, P_{k+1}), (P_t, P_{t+1}) the inner vectors I_k and I_t coincide and, consequently, also (O_k, I_{k+1}, O_{k+1}, C_{k+1}) and (O_t, I_{t+1}, O_{t+1}, Ct+1), respectively. Such situation would be very easily identifiable by the attacker because of the coincidence in the C_{k+1} and C_{t+1} cipher blocks. Observe that any other potential cause where $I_k \neq I_t$, although possible, would have a comparatively negligible probability and, therefore, if for a particular repetition of two consecutive plain-text blocks the second cipher block also coincides the attacker can be almost absolutely certain that the 'birthday' paradox coincidence is taking place in the inner vectors.

If such repetition in two cipher blocks occur, then the attacker could simply substitute the cipher blocks C_{k+2}, C_{k+3}, …, C_N, *MDC* by C_{t+2}, C_{t+3}, …, C_N, *MDC* and the last inner vectors decoded by the receiver, Y_N and Q_N, would coincide with the last ones computed by the sender, I_N and O_N, respectively. At this point, two last considerations apply to assess definitely whether this birthday paradox approach can be of any use to implement a forgery attack:

- In order to progress with the attack, it is necessary that the repetitions in the couples of plaintext blocks take place in the same cryptogram. On the contrary, since the sequence number, *S*, of the two cryptograms differ, so will do the two *MDC* codes and the receiver check will reject the received cryptogram as false with a probability of ($1-2^{-n}$);
- Although the attack could be limited to use just the material associated to the authentic cryptogram, observe that it requires either to remove a certain part of the cryptogram, or to replicate it and therefore the total length of the cryptogram will be altered to a final effective length $N' \neq N$. Since the receiver will use N' to compute the *ICV* value while the sender used *N*, then the integrity check will reject the

cryptogram as false with a probability of (1-2^{-n}). Observe that at this point, the only way to make this attack strategy to progress is to find a second couple of 'birthday' coincidences for which the block distance is exactly the same than for the first 'birthday' coincidence, leading to the conclusion that the attacker will need to proceed recurrently till at least $N^2/2$ birthday coincidences are identified in order to have a significant probability to find a pair of them with the same block-distance to compensate one with the other. But observe that this composition of $N^2/2$ birthday problems introduces a subtle complication: if the message is short the attack will not be feasible simply because of lack of material and if N is big, let's say at least in the order of magnitude of $2^{n/2}$ blocks that are required for one 'birthday' coincidence, then the attacker will require to compound a number of such birthday problems in the order of 2^n .

To finalize, taking into account the above considerations, the construction of a forgery attack based on the repetition on plaintext segments can be discarded as completely unfeasible in practice.

6. Conclusion

This paper defines an Authenticated Encryption mode called IOC which implementation is extremely lightweight, possibly the most lightweight AE mode ever proposed. Moreover, IOC offers a very high security level according the exhaustive analysis presented in this paper: whatever resources could be spent to forge IOC integrity mechanism, and assuming an "ideal" block cipher, the success probability of such attack will not be higher than $2^{-(n-1.25)}$, where n is the cipher block size.

Most remarkable IOC characteristics are the following ones:

- It can be used with any block cipher algorithm of whatever block size;
- AE is implemented with a negligible computational cost compared with only-encryption modes;
- The only keying material required are the cipher key, k, and a message counter, S, used as a nonce.
- Bit padding is required for the last plaintext block;
- No practical limitation applies to message length as long it is kept below 2^n blocks;
- No practical limitation applies to the number of messages processed with the same key k as long the values of $k'=k+S$ are not repeated (i.e. at most min{ 2^n , $2^{|k|}$ } messages);
- If IOC is used with a symmetric block ciphering algorithm, then some, or all, of the plaintext blocks can be sent in clear provided that the decoding algorithm is re-adjusted for them (i.e. for these blocks the decoding shall be identical to the encoding process and the inner vectors flows plumbed accordingly at the transition point).

We can summarize that the combined use of x-or and modulo-2^n additions are the basis for IOC strength thanks to the unpredictable evolution of the random and secret vectors *I*s and *O*s caused by the combination of these different sums. That unpredictability makes impossible to a potential attacker to synthesize any false IOC cryptogram material with a sound probability of enforcing controlled inner vectors and, thus, uncontrolled error propagation till the last block, the *ICV*, is unavoidable.

7. References

[1] - F. Recacha. IOBC: Un nuevo modo de encadenamiento para cifrado en bloque. In Proceedings: IV Reunión Española de Criptologia, Valladolid, September 1996, pages 85–92, 1996, ISBN 84-7762-645-6.

[2] - A. Zuquete and P. Guedes. Efficient error-Propagating Block Chaining. In M. Darnell, editor, Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17–19, 1997, Proceedings, number 1355 in Lecture Notes in Computer Science, pages 323–334. Springer-Verlag, Berlin, 1997..

[3] - C. J. Mitchell. Cryptanalysis of the EPBC authenticated encryption mode. In S. D. Galbraith, editor, Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings, volume 4887 of Lecture Notes in Computer Science, pages 118–128. Springer-Verlag, Berlin, 2007.

[4] F. Recacha. IOBC: A new authenticated encryption mode. January 2012. (Note: it is a literal translation to English of [1]) https://inputoutputblockchaining.blogspot.com

[5] C. J. Mitchell. Analysing the IOBC authenticated encryption mode. 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings volume 7959 of Lecture Notes in Computer Science, pages 1-12, Springer-Verlag Berlin Heidelberg 2013.

[6] F. Recacha. IOC: The Most Lightweight AE Mode? http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html /

[7] Morris Dworkin, Recommendation for Block Cipher Modes of Operation: Methods and techniques, NIST Special Publication 800-38A, December, 2001