IOC: The Most Lightweight Authenticated Encryption Mode?

March 2013 -
Francisco Recacha -
e-mail: frecacha@gmail.com -

ABSTRACT: *This paper presents*[1] *a new Authenticated Encryption (AE) mode, called IOC (Input and Output Chaining)*[2]*, that guarantees data confidentiality and integrity when used with any block cipher algorithm. The main interest of IOC is that each block of the message is only ciphered once to implement simultaneously both services while the added complexity is almost negligible and, possibly, significantly below any other AE mode. IOC is a simplification of IOBC mode (Input and Output Block Chaining) proposed by the author in 1996, but obtaining now in IOC a much stronger and lightweight AE mode. This paper presents: (a) an introduction of IOBC from which IOC is derived; (b) the specification of IOC mode; (c) an analytical model for IOC analysis together with a characterization of its core properties; and (d) an exhaustive IOC cryptanalysis, indicating that this method is secure since the best probability an attacker has to by-pass IOC integrity mechanism is not higher than* $2^{-(n-5/4)}$ *independently of whatever amount of computing resources are spent (being n the block size of the cipher algorithm used, assuming the used cipher algorithm is a 'perfect' secret randomizer).*

1. Introduction

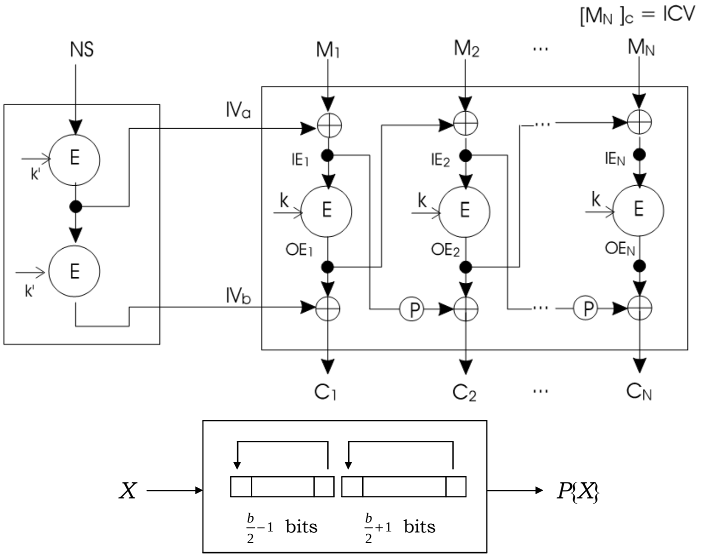
In year 1996, the author proposed as part of his Doctoral Thesis an AE mode called IOBC. IOBC was based on a cross chaining of the input and output of each encryption for consecutive steps and appending at the last message block a fixed bit stream called Integrity Check Value (*ICV*, see figure 1). IOBC is adequate to be used with any block cipher, disregarding its block size, and it was assessed then that the confidentiality strength of the used cipher algorithm was preserved and that it exhibited an integrity strength, for messages of maximum length of $n^2/2-1$, bounded by a best probability for any forgery attack around $2^{-n/2}$, being n the block size of the used cipher algorithm. Unfortunately, IOBC proposal received very little public dissemination since then: the idea was published in the corresponding PhD Thesis report and just a public presentation was given during "IV Reunión Española sobre Criptología" organized by Universidad de Valladolid in 1996 [1]. Afterwards the author changed jobs from the academic field in Universitat Politècnica de Catalunya to the private sector and no publication was made additionally to the two already mentioned publications that were written in Spanish.

Fortunately, like in a message in a bottle, IOBC presentation was attended by Zúquete and Guedes in Valladolid in 1996, who proposed later a supposedly enhancement of IOBC called EPBC [2] that received much broader international dissemination than IOBC itself. In 2007, Mitchell published a cryptanalysis of EPBC showing that the integrity offered by this encryption mode was in fact quite easily broken [3], but the analysis could not conclude anything on IOBC since Zúquete paper didn't include any description of it. This cryptanalysis arrived to the knowledge of the IOBC author several years later and he translated to English in 2013 the original Spanish paper and delivered it to Mitchell and it was published in Internet [4].

After IOBC was analyzed by Mitchell in February 2013, he concluded that the best probability an attacker has to success with a forgery attack to IOBC is actually bounded by $2^{-n/3}$ instead of $2^{-n/2}$ basically because the cross-feeding permutation used by IOBC is composed by two bit sub-rotations of lengths ($n/2$-1) and ($n/2$+1), respectively, and either one or the other are multiple of 3 [5]. Equivalently, if the forgery probability can be maintained down to $2^{-n/2}$ but then the maximum message length has to be reduced by a factor of 3. As Mitchell also explains in his paper, IOBC pertains to a family of <<*'special' modes of operation for block ciphers, designed to offer 'low cost' combined integrity and confidentiality protection by combining encryption with the addition of special redundancy to the plaintext*>> and, therefore, his analysis could be closing a long story of over 30 years that has showed that all of the proposals pertaining to that family of operation modes have exhibited one by one some kind of weakness.

[1] IOC and IOBC can be freely used without any restriction imposed by the author except to be fairly credited by its - invention when used. To the knowledge of the author no patent neither other type of Industrial Property Rights applies - neither to IOC nor IOBC at any region of the world. -

[2] IOC name intends to point out two basic ideas: while it is a simplification of IOBC, IOC maintains its core concept. -



NOTE: *b*, the block size, is assumed a 2 power (as usual).

Figure 1: IOBC authenticated encryption mode for block ciphers.

Despite Mitchell conclusions in [5], we show in this paper that there is still some hope for such kind of modes of operation and that even the happy end of that long story has been achieved at the end. Particularly, we show that the basic skeleton of IOBC concept shows interesting crypto hashing properties that can be much better and simpler used than in its original design, leading to the design of a new and much stronger and lightweight IOC mode that works without the limitations of its parent mode.

The IOBC limitations solved, or strengthened, by IOC are the following ones:

- The size of the plain message shall be in IOBC smaller than $n^2/2$-1 (e.g. 16.376 bytes if n=64 bits). IOC has no message length limitation *per-se* (i.e. the only limitations will be the ones imposed by key management criteria driven mainly by the used block encryption algorithm);
- A separated key k' is required by IOBC to generate a pair of initializing vectors (IV_a and IV_b) for each message and this key shall be renewed at least each 2^n messages. IOC does not require such key for the initializing vectors an it is sufficient that they are random, different and secret. Moreover, although optional, they do not need to be generated for each encoded message (see 2.2 for more details);
- A secret and random *ICV* (Integrity Check Vector) of $n/2$ bits is required by IOBC to be appended at the end of the plaintext message. Although IOC also appends a *MDC* block of n bits to verify the integrity of the decrypted message, this vector is now computed by IOC as a crypto-hash of the plain and cipher texts that allows to the receiver to check whether the recovered message is authentic or not [3].
- An the last, but not the least, while both IOBC and IOC maintain, or even improve, confidentiality strength provided by the used block encryption algorithm, regarding the integrity strength, IOBC it is bounded by a best probability for any forgery attack of $2^{-n/3}$, while in IOC this probability is reduced down to $2^{-(n-5/4)}$.

To finalize with this introduction, the author wants to acknowledge to Chris Mitchell the discussions maintained around IOBC and the cryptanalysis he performed. These discussions enabled the author to realize that at some point during IOBC design in 1996 a wrong way was taken introducing a function into the backwards feedback instead of substituting a pre-fixed ICV by a random value intrinsically computed by the core mechanism of this mode of operation and introducing an slight, but critical, un-balance in the chaining sum operators that substitute the bit permutation used by IOBC. Now with this lesson learnt, it has been possible to roll back to the essence of IOBC and to take now the right way to come out with IOC design, possibly the most light-weight AE mode ever proposed.

[3] *MDC*: Modification Detection Code (see section 2.1).

2. The IOC Operation Mode

2.1 IOC Specification

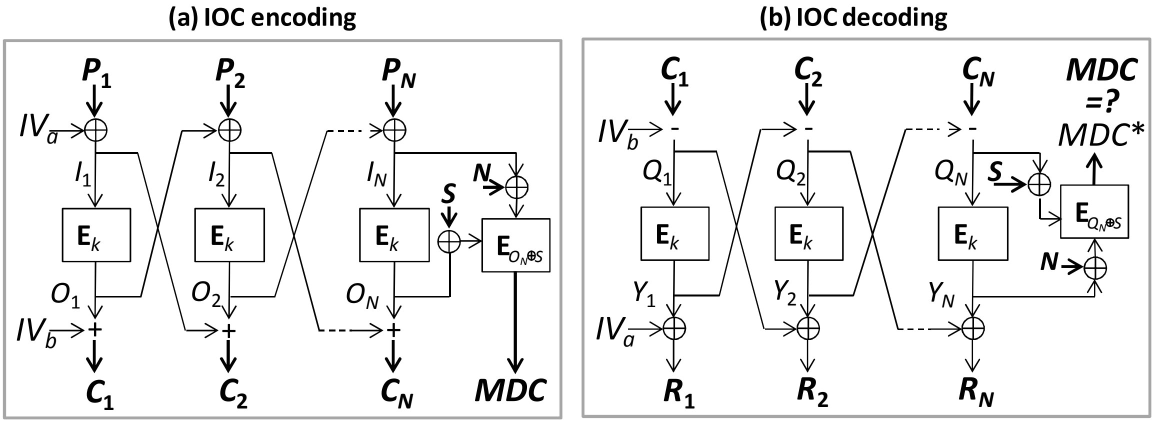


Figure 2: IOC authenticated encryption mode for block cipher.

Figure 2 illustrates IOC cipher operation mode and can formally be defined as follows:

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

where

a) P_i is a plaintext block of n bits of the original plain message and C_i its corresponding cipher-text block;

b) $O_0 = IV_a$ is a random and secret n bit vector;

c) $I_0 = IV_b$ is a random and secret n bit vector different from IV_a;

d) $MDC = E_{O_N \oplus S}\{I_N \oplus N\}$ is the Modification Detection Code;

e) $E_k\{X\}$ the result of the block encryption of a n bit vector X, using the key k;

f) $\oplus$ is the x-or binary operator applied bit by bit to the two input n-bit vectors;

g) + is the regular arithmetic addition modulo-2^n;

h) N is the length, in n-bit blocks, of the plain message[4];

i) S is a unique sequence counter assigned to each encoded/ transmitted message.

Analogously, the inverse IOC decryption operation is defined as follows:

$$\left.\begin{aligned} R_i &= Y_i \oplus Q_{i-1} \\ Y_i &= D_k\{Q_i\} \\ Q_i &= C_i - Y_{i-1} \end{aligned}\right\} \text{ for } i = 1, \ldots, N; \tag{2}$$

where

a) $O_o = IV_a$ and $I_o = IV_b$ as in the encryption operation;

b) $D_k\{\ \}$, the inverse operator of $E_k\{\ \}$;

c) $MDC^* = E_{Q_N \oplus S}\{Y_N \oplus N\}$, and the decoded plain message is accepted as authentic only if $MDC^* = MDC$;

d) - is the regular arithmetic subtraction modulo-2^n.

It is immediate to demonstrate that the decryption operation for any authentic cryptogram is just the inverse of the encryption one (i.e. $R_i = P_i$ for i=1 … N and $MDC^* = MDC$).

[4] If the length of the plain-text message is not multiple of n, then additional padding bits with whatever value shall be added till the last block is completed. The signalling mechanism to notify to the receiver which is the injected padding is out of the scope of IOC specification, although assumed that it will be part of the authenticated plain message.

2.2 Operational Guidelines for IOC Initializing Vectors and S Message Counter

The initialization vectors IV_a and IV_b needed to process each message shall be compliant with the following operational requirements: (3)

1. - IV_a and IV_b shall be shared between the sender and the receiver;
2. - IV_a and IV_b shall be random;
3. - IV_a and IV_b shall be secret;
4. - IV_a and IV_b shall be different between them;
5. - IV_a or IV_b shall be different for all the messages sequence encoded without a fresh IV reset (see below).

Given the above requirements, the following design guidelines are specified as part of IOC mode to manage IV_a and IV_b along a security session[5] (see figure 3): - (4)

a) - For the first message of a security session a pair of "fresh" IV_a and IV_b is generated by some mean that guarantees they are random, different and secret. In particular, the initial message sequence number S can be encrypted with $E_k\{\}$ in ECB mode (i.e. $IV_a = E_k\{S\}$ and $IV_b = E_k\{IV_a\}$, being S the message sequence counter. In any case, this method is optional and any alternative method in place that guarantees the above operational requirements can be used to generate a 'fresh' pair of *IV*s.;

b) - For subsequent messages, IV_a and IV_b can be taken as the last inner vectors O_N and I_N respectively, from the previous message (i.e. $IV_{a,S+1} = O_{N,S}$ and $IV_{b,S+1} = I_{N,S}$ - see figure 3);

c) - Alternatively to method (b), IV_a and IV_b can be reset with "fresh" values at any point for a particular message using the method specified in (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 *IV*s reset may help, for instance, resynchronization in case of message losses in applications tolerant to data-loss.

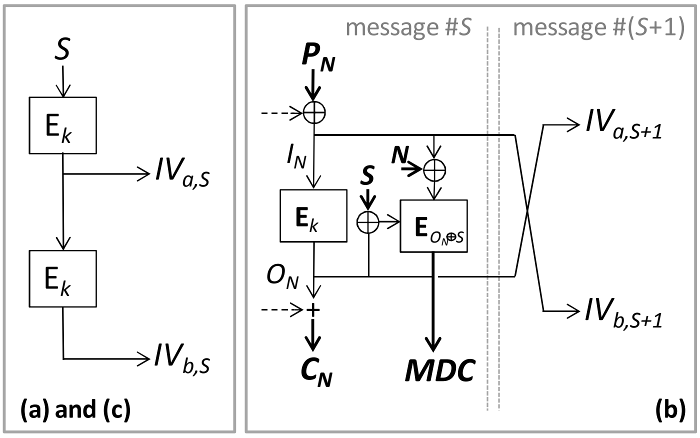


Figure 3: Operational guidelines for the initializing vectors

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

6. - Each time a security session is initiated, a message counter S, shall be reset to a initial value (can be, 0, 1, or whatever value agreed between the sender and the receiver);
7. - S value shall be incremented synchronously for each message both by the sender and the receiver sides, independently it is used, or not, for fresh *IV* renewals;
8. - 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 as specified in section 2.2;
9. - During a security session where a same *k* key is used, *S* counter shall not be repeated for any message.

To finalize with this section, only the following two requirements apply to plaintext message padding:

10. -If necessary, padding bits with whatever value shall be appended to the complete the last message block;
11. -The number of padding bits shall be indicated as part of the authenticated message.

[5] A *security session* is here understood as the chain of plain messages encoded using a same ciphering *session* key *k*.

3. Some Observations on IOC Specification

3.1 Confidentiality of the Inner Vectors and Algebraic Relationships between Them

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 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 for any i value even in the case a potential attacker would know all the plain and cipher text 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), then, the main information from IOC definition available to the attacker are equations (1) rewritten as it follows:

$$\left.\begin{array}{l} C_i = O_i + I_{i-1} \\ P_i = I_i \oplus O_{i-1} \end{array}\right\}, \text{ for } i = 1, \ldots, N \text{ and other assumptions like in (1).} \qquad (5)$$

Complemented by the fact that the encryption algorithm establishes the following unknown deterministic relations among the unknown terms:

$$\left.\begin{array}{l} O_i = E_k\{I_i\}, \text{for } i = 1,\ldots,N \\ MDC = E_{O_N \oplus S}\{I_N \oplus N\} \end{array}\right\}. \qquad (6)$$

Although, (5) constitutes an analytical model for the relationships between the inner vectors and the plain and cipher text blocks, the simultaneous use of the two types of sum operators, the conventional arithmetic addition plus 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:

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

where $\Delta(A, B)$ is the difference vector that comes produced by the up to (n-1) bit carries than can take place in 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 the values 0 and 1 will not follow a uniform probability since the accumulated propagation of carries making that some values of $\Delta(A, B)$ more probable than others.Using (7), equations (5) can be rewritten now in a more intelligible and manageable form:

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

Or in a more compact writing:

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

where $\Delta_i = \Delta(O_i, I_{i-1}) = (O_i \oplus I_{i-1}) \oplus (O_i + I_{i-1})$ is the bit carry-delta vector 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. It is immediate to show that (8) is an indeterminate system of $2N$ independent linear equations and $3N$+2 unknown terms (I_0= IV_a, I_1, …, I_N, O_0= IV_b, O_1, …, O_N, and Δ_1, Δ_2, …, Δ_N) 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.

[6] In any case, equations (8) together with the non-linear equations (6) and the expression of the carry-delta vectors, contain relevant information that, in principle, could of some use to implement a forgery attack.

A very useful alternative writing of equations (8) for IOC analysis, is its corresponding matrix form (9) presented below. For instance, this matrix expression makes immediately evident the independent nature of the equations and 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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&0&0&0&0&0&0&0&0&0&0&0\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdots&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdots&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdots&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot&\cdot\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&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&\cdots&0&0&0&0&0&0&0&1&0&1&0
\end{bmatrix}
\times
\begin{bmatrix}
I_0\\ O_0\\ \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\\ \ldots\\ \Delta_{N-3}\\ I_{N-3}\\ O_{N-3}\\ \Delta_{N-2}\\ I_{N-2}\\ O_{N-2}\\ \Delta_{N-1}\\ I_{N-1}\\ O_{N-1}\\ \Delta_N\\ I_N\\ O_N
\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-2}\\ P_{N-2}\\ C_{N-1}\\ P_{N-1}\\ C_N\\ P_N
\end{bmatrix}
\qquad (9)$$

3.2 Information Entropy Contained in the Carry-Delta Vectors, Δ_i

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 C'_i, C_{j+1}, 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 equal some non-authentic O_j value. 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

[7] That statement is based in the fact that each Δ_i appears at most once at any column, and therefore there is no linear combination of those equations that makes possible to remove any 'unconvenient' carry-delta vector.

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.

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:
$\Delta = \Delta(A, B) = (A \oplus B) \oplus (A + B)$.

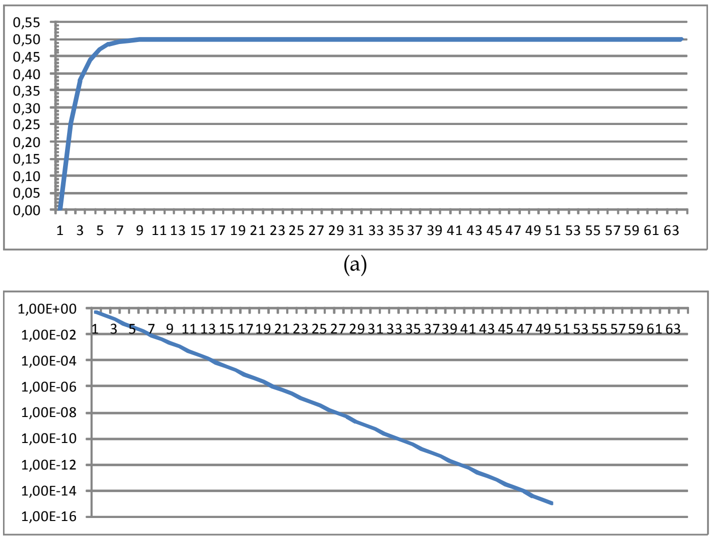
Then, the only difference between the two bits in the i-th position of $(A \oplus B)$ and $(A + B)$ comes from whether a carry coming the bit addition in the (i-1)-th position has to be applied for the sum of the i-th position. Thus, it is immediate that the i-th bit of Δ will be 1 if such carry bit 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 accumulating 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;
- $\mathrm{P}_2 = \frac{1}{4}$, since only if the first two added bits were simultaneously 1 the bit carry is produced;
- $\mathrm{P}_3 = \frac{1}{4}(1 - \mathrm{P}_2) + \frac{3}{4}\mathrm{P}_2$, 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 $\mathrm{P}_i = \frac{1}{4}(1 - \mathrm{P}_{i-1}) + \frac{3}{4}\mathrm{P}_{i-1} = \frac{1}{4} + \frac{\mathrm{P}_{i-1}}{2}$ 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 as the inner vectors do for all of their bits. Thus, let's quantify the entropy contained in Δ_i vectors, or on other terms, the equivalent length of Δ_i in terms of perfectly 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:

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

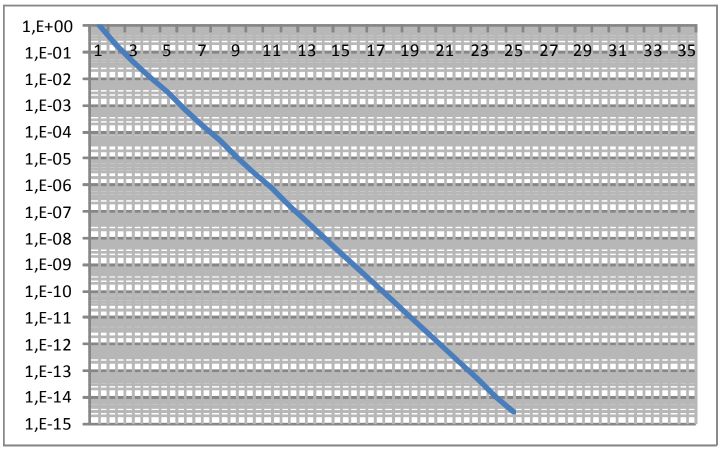


Figure 5: log(1 - H([Δ]*i*)) vs *i*

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.

n **(physical bits)**	**1**	**2**	**3**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
H(Δ) **(eq. 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 for different block sizes

4. IOC Confidentiality Strength Proof

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, she/he will be unable to determine that P_k if 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 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 highlight before closing this short section that the secrecy of the inner vectors I_i and O_i is not relevant only for data confidentially service offered by IOC, but together with their cryptographic–hash nature, they provide to IOC a fundamental property to guarantee the message integrity. Moreover, it is also worth to remark that, as other encryption chaining modes, IOC shows also some other collateral advantages thanks, in this case, to the secrecy of the inner vectors I_i and O_i. For instance, it makes more difficult the cryptanalysis of the core encryption algorithm $\mathrm{E}_k\{\}$ since a potential attacker will not be able to compile any dictionary of plain/ciphered blocks (X, $\mathrm{E}_k\{X\}$) in order to try to determine the encryption key k or to take profit of such dictionary as it could happen, for instance, with ECB encryption mode.

5. IOC Integrity Strength Proof

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. Further to those three classes the following sub-classes can be further identified in order to have a detailed case classification to be used as check-list for an exhaustive cryptanalysis guide:

- 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). Observe that 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 *IV*s 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.

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* ≤ N)
  - $I_i = \bigoplus_{k=0}^{i-2/2} \left(P_{i-2k} \oplus C_{i-(2k+1)} \oplus \Delta_{i-(2k+1)}\right) \oplus IV_b$
  - $O_i = \bigoplus_{k=0}^{i-2/2} \left(C_{i-2k} \oplus \Delta_{i-2k} \oplus P_{i-(2k+1)}\right) \oplus IV_a$
- Odd blocks (1≤ *i* ≤ N)
  - $I_i = P_i \oplus \left(\bigoplus_{k=0}^{i-3/2} \left(C_{i-(2k+1)} \oplus \Delta_{i-(2k+1)} \oplus P_{i-(2k+2)}\right)\right) \oplus IV_a$
  - $O_i = C_i \oplus \left(\bigoplus_{k=0}^{i-3/2} \left(P_{i-(2k+1)} \oplus C_{i-(2k+2)} \oplus \Delta_{i-(2k+2)}\right)\right) \oplus IV_b$

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 *MDC* vector separately computed by the sender and receiver according to $MDC = E_{O_N \oplus S}\{I_N \oplus N\}$. That is, the IOC integrity mechanism validity is equivalent to the condition that the x-or sums of last inner vectors, I_N and O_N, with the *N* and *S* parameters computed by the receiver shall be the same ones computed by the sender if and only if the received cryptogram is exactly the same that was encoded and sent.

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: $Q_i = C_i \oplus Y_{i-1} \oplus \Delta(Q_i, Y_{i-1})$.

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 *MDC**. 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 with a probability of $(1\text{-}2^{-n})$. Nonetheless, it is directly derived from equation (11) that for a contiguous 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 separate plain and cipher blocks chains and carry-delta vectors that impact in the values of I_N and O_N for $\dot{C}_{j\text{-}1}$ their x-or sums are 0, then if in the position of $\dot{C}_i$ is inserted the cryptogram $\dot{C}_j$ 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 *MDC* block, but at least the last inner vectors, and the *N* parameter used by the receiver would much with the used ones to compute that *MDC*. But we could say at least that this forgery attack would elude almost all the layers of IOC integrity mechanism). Fortunately, although the attacker would know all those plain and cipher text blocks, the sums of all the involved carry-delta vectors 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.

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 *MDC* 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 *MDC** computation.

[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).

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

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 *MDC** value will match C_N 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 ($i < j$), ($1 \leq i < N$) and ($1 < j \leq N$) 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-1},O_{i-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 Q_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 uncontrollable error propagation towards the *MDC** 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:
   a.1) Eventual repetitions of the inner vectors used to substitute the subsequent blocks of the cryptogram by others taken from other authentic cryptogram.
   a.2) Exploit of 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 $1 \leq i \leq N$) 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:

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

If such condition could be implemented, then only replicating C_{j+1}, C_{j+2}, … and C_N after $C_i^{'}$ would allow to obtain the same last inner vectors I_N and O_N that were used by the sender to compute the *MDC*. Or, at least, would avoid uncontrolled error propagation and to try to complement this first substitution with a second one to complete a forgery attack. 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, to the only way to make that such condition happens, $|j - i|$ shall be even and then it is very simple to compute a for $C_i^{'}$ that only depend on a I_{i-1} and a O_j :

- If ($j > i$), then

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

- If ($i > j$), then

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

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

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

Since the first term, the x-or sum of the carry-deltas will be cero with a probability smaller than $2^{-(n-5/4)}$, 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 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 required 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 this '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 the attack 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 *MDC** 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 in 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. Conclusions

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.

IOC is a simplification of a previous AE mode called IOBC that was proposed in 1996 by the author and it was supposed to offer integrity strength of $2^{-n/2}$. That is, the probability of success of the best forgery attack was bounded by such probability. In 2013, Mitchell [5], demonstrated that IOBC integrity strength was really bounded by $2^{-n/3}$, if maximum message length was maintained (alternatively if maximum message length is shortened by a factor of 3, then the $2^{-n/2}$ strength is preserved).

IOC is a considerable simplification of IOBC both for implementation and operation aspects, and it introduces at the same time a huge improvement in the offered integrity security strength. As the most remarkable characteristics of IOC, the following ones shall be highlighted:

- IOC allows an AE implementation with a computational complexity cost almost negligible not only in comparison with only-confidentiality ECB encryption but also in comparison with other AE modes;

- IOC operation is also extremely simple: no additional secret material is required apart from the encryption key, *k*, already required for any 'confidentiality-only' operational mode. Moreover, IOC only requires one additional encryption per message in order to compute its corresponding *MDC*, and two more for generating a pair of initializing vectors but only for the first message of a security session;

- In comparison with its parent IOBC mode, most significant IOC evolution is that while IOBC used a permutation operator to introduce some unpredictability on the evolution of the inner vectors, IOC replaces that permutation by a much simpler use of two different linear operators: x-or is used to chain previous output inner vector toward the input of next cipher step but regular modulo-2^n addition is now used to chain the other chaining line. The overall result is an extremely low-cost asymmetric chaining feedback that makes almost totally unpredictable the evolution of the inner vectors and almost impossible to implement a forgery attack.

Before closing this document, perhaps it is worth to raise just a few considerations:

- If IOC is used with a symmetric block ciphering algorithm, then some, or all, of the plaintext message 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 points);

- 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 this combination of these different sums. That unpredictability makes impossible to a potential attacker to synthesize any false IOC cryptogram block with a sound probability of enforcing controlled inner vectors and, thus, uncontrolled error propagation till the last block, the MDC, is unavoidable.

7. References

[1] - F. Recacha. IOBC: Un nuevo modo de encadenamiento para cifrado en bloque. In Proceedings: IV Reunion Espanola 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. February 2013. Submitted to ACISP 2013.