Integrity-protecting block cipher modes — Untangling a tangled web

Chris J. Mitchell
Information Security Group, Royal Holloway, University of London
www.chrismitchell.net

6th March 2024

Abstract

This paper re-examines the security of three related block cipher modes of operation designed to provide authenticated encryption. These modes, known as PES-PCBC, IOBC and EPBC, were all proposed in the mid-1990s. However, analyses of security of the latter two modes were published more recently. In each case one or more papers describing security issues with the schemes were eventually published, although a flaw in one of these analyses (of EPBC) was subsequently discovered — this means that until now EPBC had no known major issues. This paper establishes that, despite this, all three schemes possess defects which should prevent their use — especially as there are a number of efficient alternative schemes possessing proofs of security.

1 Introduction

This paper is a somewhat tangled story of three different, albeit very closely related, proposals for a block cipher mode of operation providing authenticated encryption. Sadly, all of the schemes been shown to be insecure — often in quite different ways. This is, to the author's knowledge, the first paper to bring together the three strands of research; at the same time errors in previous cryptanalysis are acknowledged and further attacks described.

All three of the schemes we examine are examples of a 'special' mode of operation for block ciphers, designed to offer 'low cost' combined integrity and confidentiality protection by combining encryption with the addition of simple (or fixed) redundancy to the plaintext. The underlying idea is to design the mode so that modifying the ciphertext without invalidating the added redundancy is impossible without knowledge of the encryption key. Such modes are the theme of section 9.6.5 of Menezes, van Oorschot and

Vanstone's landmark book [6]. Two main methods for adding redundancy have been proposed:

- add a fixed block (or blocks) at the end of the plaintext, which may be public or secret (in the latter case the block acts as an auxiliary key);
- append to the plaintext some easily computed and simple (public) function of the plaintext.

In either case we refer to the block added to the end of the plaintext as a *Manipulation Detection Code (MDC)*. Whichever approach is adopted, the method for computing the MDC needs to be simple, or it offers no advantage over the more conventional 'encrypt then MAC' approach.

In all the modes we examine, the MDV (also known as an *Integrity Control Value* (ICV)) is defined to be a fixed, possibly secret, final plaintext block.

- The first of the three schemes we examine is known as PES-PCBC. The name derives from it being a version of PCBC mode designed specifically for use in a Privacy Enhanced Socket (PES) protocol. There are actually a number of modes known as PCBC (Plaintext-Ciphertext Block Chaining); the version on which PES-PCBC was based is that incorporated in Kerberos version 4. This mode was shown to be insecure for the purposes of integrity protection by Kohl, [3]. For a discussion of the weaknesses of other variants of PCBC see [7]. The design goal for PES-PCBC was to enhance the security of PCBC by preventing the known attacks. This scheme and its properties are discussed in Section 3.
- The second scheme, known as IOBC (short for *Input and Output Block Chaining*) was published in 1996 by Rechacha [10]. IOBC is a straightforward variant of PES-PCBC. The paper describing IOBC was originally published in Spanish, and it wasn't until an English language translation was kindly provided by the author in around 2013[1] that any further discussion of the scheme appeared. This scheme and its properties are discussed in Section 4.
- The last of the three schemes, known as EPBC (short for *Efficient Error-Propagating Block Chaining*) was published in 1997 by Zúquete and Gudes [16]. It is very similar to IOBC, and is designed to be used in exactly the same way. The design goal was to address issues in IOBC which restricted its use to relatively short messages. A possible method of cryptanalysis allowing certificational forgeries was published in 2007 [8], although Di et al. [2] showed in 2015 that the attack does

[1] See `https://inputoutputblockchaining.blogspot.com/`

not work as claimed. The scheme and its level of security are discussed in Section 5.

In Section 6 we examine other more general attacks which apply to all, or at least large classes of, schemes sharing the same underlying structure as PES-PCBC, IOBC and EPBC. In doing so we establish that all three modes suffer from attacks which mean that they should not be adopted. We conclude the paper in Sections 7 and 8 by first briefly mentioning two other related modes (and their analyses), and then summarising the main conclusions that can be drawn from the analyses given.

2 Preliminaries

We start by introducing some notation and assumptions. All three modes operate using a block cipher. We write:

- n for the plaintext/ciphertext block length of this cipher;
- $e_K(P)$ for the result of block cipher encrypting the n-bit block P using the secret key K;
- $d_K(C)$ for the result of block cipher decrypting the n-bit block C using the key K; and
- $\oplus$ for bit-wise exclusive-or.

Finally we suppose the plaintext to be protected using the mode of operation is divided into a sequence of n-bit blocks (if necessary, having first been padded): $P_1, P_2, \ldots, P_t$, where P_t is equal to the MDC.

3 PES-PCBC

The description follows Zúquete and Guedes [15], although we use the notation of [16]. The scheme uses two secret n-bit Initialisation Vectors (IVs), denoted by F_0 and G_0. The nature of their intended use is not described in [15]; however it is stated in [16] that the 'initial values of F_{i-1} and G_{i-1} are distinct, secret initialisation vectors', which is what we assume below.

The PES-PCBC encryption of the plaintext $P_1, P_2, \ldots, P_t$ operates as follows:

$$\begin{aligned} G_i &= P_i \oplus F_{i-1}, \quad (1 \leq i \leq t), \\ F_i &= e_K(G_i), \quad (1 \leq i \leq t), \\ C_i &= F_i \oplus G_{i-1}, \quad (1 \leq i \leq t). \end{aligned}$$

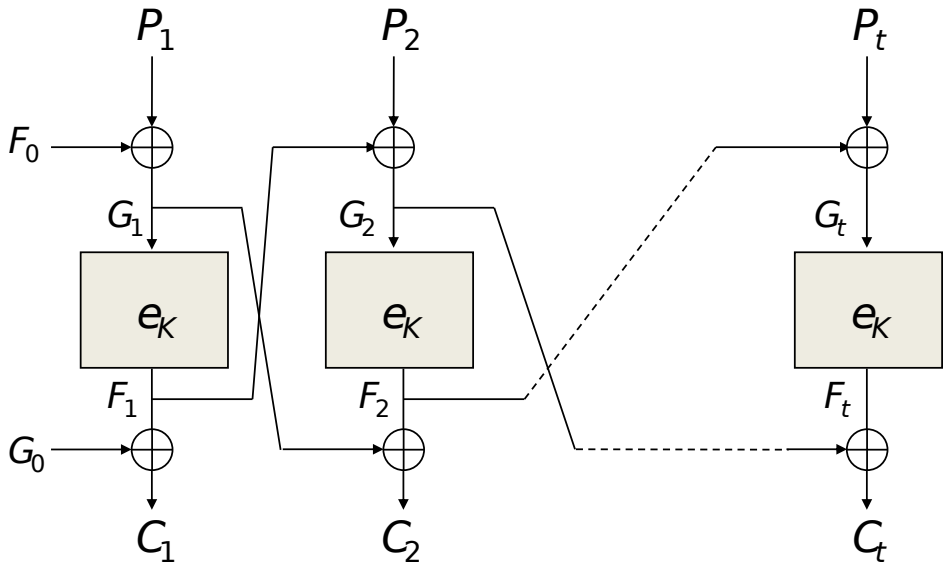

Figure 1: PES-PCBC encryption

The resulting ciphertext is $C_1, C_2, \ldots, C_t$.

The operation of the mode (when used for encryption) is shown in Figure 1. Note that we refer to the values F_i and G_i as 'internal' values, as they are computed during encryption, but they do not constitute part of the ciphertext.

3.1 Cryptanalysis

It is clear that weaknesses in PES-PCBC when applied for integrity protection were discovered soon after its publication in 1996. The 1997 paper by Zúquete and Guedes, [16], briefly outlines a known-plaintext attack allowing simple forgeries. We next describe a slightly simplified variant of this attack, requiring just one known plaintext block instead of two.

First observe that, using the same notation as before, PES-PCBC decryption operates as follows:

$$\begin{aligned} F_i &= C_i \oplus G_{i-1}, \quad (1 \leq i \leq t), \\ G_i &= d_K(F_i), \quad (1 \leq i \leq t), \\ P_i &= G_i \oplus F_{i-1}, \quad (1 \leq i \leq t), , \end{aligned}$$

and the receiver of an encrypted message will accept it as genuine if the final recovered plaintext block P_t equals the expected MDC.

The following result captures the attack.

Theorem 3.1 *Suppose the ciphertext $C_1, C_2, \ldots, C_t$ was constructed using PES-PCBC from the plaintext $P_1, P_2, \ldots, P_t$, and that j satisfies $1 < j < t$. Suppose the $t+2$-block ciphertext $C'_1, C'_2, \ldots, C'_{t+2}$ is constructed as follows:*

$$\begin{aligned} C'_i &= C_i, \quad (1 \leq i \leq j), \\ C'_{j+1} &= P_j, \\ C'_i &= C_{i-2}, \quad (j+2 \leq i \leq t+2). \end{aligned}$$

When decrypted to yield $P'_1, P'_2, \ldots, P'_{t+2}$, the value of the final plaintext block P'_{t+2} will equal P_t for the original (untampered) message, and hence will pass the integrity check.

Proof In the discussion below we refer to the 'internal values' generated during decryption of $C'_1, C'_2, \ldots, C'_{t+2}$ as F'_i and G'_i. First note that, trivially: $F'_i = F_i$, $G'_i = G_i$, and $P'_i = P_i$, $(1 \leq i \leq j)$. Next observe that

$$\begin{aligned} F'_{j+1} &= C'_{j+1} \oplus G'_j \\ &= P_j \oplus G_j \\ &= F_{j-1}. \end{aligned}$$

Hence $G'_{j+1} = d_K(F'_{j+1}) = d_K(F_{j-1}) = G_{j-1}$. Finally, we have $P'_{j+1} = G'_{j+1} \oplus F'_j = G_{j-1} \oplus F_j = C_j$. Since $C'_{j+2} = C_j$, $F'_{j+1} = F_{j-1}$ and $G'_{j+1} = G_{j-1}$, it is immediate that $F'_{j+2} = F_j$, $G'_{j+2} = G_j$, and $P'_{j+2} = P_j$, and the desired result follows. ∎

3.2 Impact

The above attack shows that, given just one PES-PCBC-encrypted message and knowledge of only a single plaintext block for this encrypted message, a 'fake' message can be constructed will be guaranteed to pass the integrity checks. This fact meant that it has been recognised since 1996/97 that PES-PCBC should not be used.

Before proceeding note that the originally proposed context of use for PES-PCBC, as described in [15], involved including an encoded version of the message length in the first plaintext block. In such a case the attack described in Theorem 3.1 will not work since it involves lengthening the message by two blocks. However, a slightly more involved version of the Theorem 3.1 attack (outlined in [16]) avoids changing the message length and hence works even if the message length is encoded in the plaintext — at the cost of requiring knowledge of two plaintext blocks instead of one.

4 IOBC

The IOBC mode was published in 1996 by Recacha [10], the same year as the publication of PES-PCBC. One might reasonably conclude that the design of IOBC, as a modification to PES-PCBC, was motivated by the weaknesses in PES-PCBC, but curiously the Recacha paper does not even mention PES-PCBC. Certainly, the inclusion of the function g in the feedback stops the attack on PES-PCBC working — at least in a naive way.

4.1 IOBC operation

We start by describing the operation of the IOBC mode of operation. We base the description on Recacha's 1996 paper [10], although we use the same notation as in the description of PES-PCBC. We suppose that the cipher block length n is a multiple of four (as is the case for almost all commonly used schemes), and put $n = 2m$ where m is even. The scheme uses two secret n-bit IVs, denoted by F_0 and G_0. The nature of the intended restrictions on their use is not altogether clear; one suggestion in the original Recacha paper [10] is that they should be generated as follows.

Suppose K' is an auxiliary key used solely for generating the IVs. Suppose also that S is a sequence number, managed so that different values are used for every message. Then $F_0 = e_{K'}(S)$ and $G_0 = e_{K'}(F_0)$. For the purposes of this paper we assume that F_0 and G_0 are always generated this way, and thus the scheme can be thought of as employing a pair of block cipher keys and a non-secret, non-repeating, sequence number (which must be carefully managed to prevent accidental re-use of sequence number values). Note that special measures will need to be taken if the same key is to be used to encrypt communications in both directions between a pair of parties. Avoiding sequence number re-use in such a case could be achieved by requiring one party to start the sequence number they use for encryption at a large value, perhaps halfway through the range.

The IOBC encryption of the plaintext $P_1, P_2, \ldots, P_t$ operates as follows:

$$\begin{aligned} G_i &= P_i \oplus F_{i-1}, \quad (1 \le i \le t), \\ F_i &= e_K(G_i), \quad (1 \le i \le t), \\ C_i &= F_i \oplus g(G_{i-1}), \quad (2 \le i \le t), \end{aligned}$$

where $C_1 = F_1 \oplus G_0$ and g is a function that maps an n-bit block to an n-bit block, defined below. The operation of the mode (when used for encryption) is shown in Figure 2.

The function g is defined as follows. Suppose X is an n-bit block, where $n = 2m$. Suppose also that $X = L||R$ where L is the leftmost $m - 1$ bits of

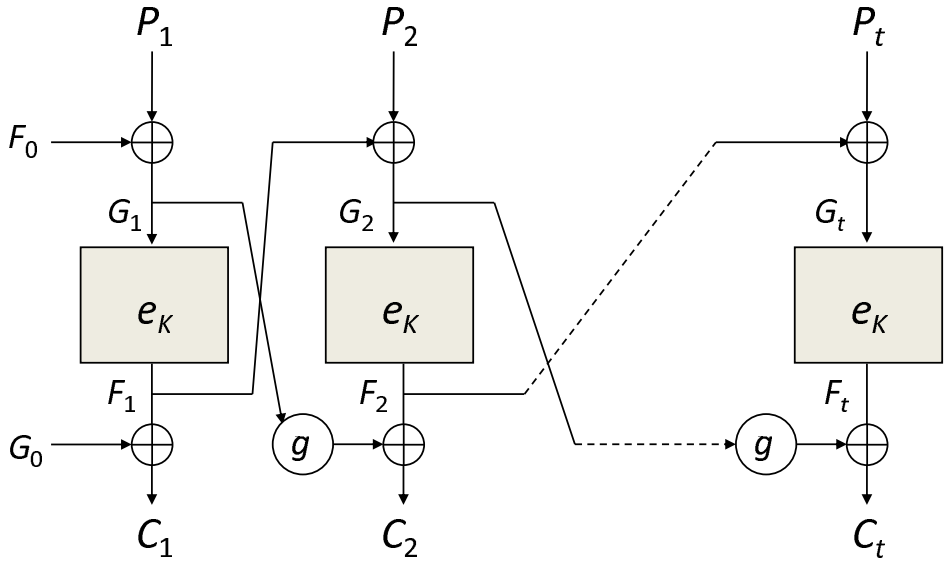

Figure 2: IOBC encryption

X and R is the rightmost $m+1$ bits of X (and, as throughout, $||$ denotes concatenation). Then

$$g(X) = (>_1 (L))||(>_1 (R))$$

where $>_i$ denotes a rightwards (cyclic) shift by i bit positions.

Decryption operates similarly. We have:

$$\begin{aligned} F_i &= C_i \oplus g(G_{i-1}), \quad (2 \le i \le t), \\ G_i &= d_K(F_i), \quad (1 \le i \le t), \\ P_i &= G_i \oplus F_{i-1}, \quad (1 \le i \le t). \end{aligned}$$

and $F_1 = C_1 \oplus G_0$, where d denotes block cipher decryption.

It should be clear that PES-PCBC is the same as IOBC with the exception that in PES-PCBC the function g is the identity function.

4.2 Remarks on use

As described above, we assume throughout that the IVs F_0 and G_0 are derived by ECB-mode-encrypting a sequence number using a secondary key. Thus the ciphertext blocks will be a function of this serial number (as well as the pair of keys used). We thus write $[S], C_1, C_2, \ldots, C_t$ for a sequence of ciphertext blocks, meaning that $C_1, C_2, \ldots, C_t$ were encrypted using the

sequence number S. This is logical, since the sequence number will need to be sent or stored with the ciphertext to enable correct decryption.

IOBC should only be used with relatively short messages. As specified by Recacha [10] (and for reasons which become clear below), a message to be encrypted using IOBC shall contain at most $n^2/2-1$ plaintext blocks, where n is the plaintext block length. Thus for $n = 64$ and $n = 128$, the two most commonly used block lengths, a message shall contain at most 2047 and 8191 blocks, respectively.

As with all modes we discuss here, data integrity is achieved by adding an MDC to the end of the plaintext.

4.3 Cryptanalysis

We start by giving a simple result that is implicit in Recacha [10]. It is interesting to note that this result applies regardless of the choice of the choice of function g, i.e. to any mode operating as in Figure 2.

Lemma 4.1 (Mitchell, [9]) *Suppose* $[S], C_1, C_2, \ldots, C_t$ *and* $[S'], C'_1, C'_2, \ldots, C'_{t'}$ *are IOBC encrypted versions of the plaintext sequences* $P_1, P_2, \ldots, P_t$ *and* $P'_1, P'_2, \ldots, P'_{t'}$*, respectively. If the ciphertext:*

$$[S'], C^*_1, C^*_2, \ldots, C^*_{t-v+u} =$$
$$[S'], C'_1, C'_2, \ldots, C'_{u-1}, C_v \oplus g(G'_{u-1}) \oplus g(G_{v-1}), C_{v+1}, \ldots, C_t$$

is submitted for IOBC decryption (where $1 < u < t'$ *and* $1 < v < t$*, and* G_{v-1} *and* G'_{u-1} *are values computed during the encryption of the respective sequences of blocks), then the resulting sequence of plaintext blocks* $P^*_1, P^*_2, \ldots, P^*_{t-v+u}$ *will be equal to*

$$P'_1, P'_2, \ldots, P'_{u-1}, P_v \oplus F'_{u-1} \oplus F_{v-1}, P_{v+1}, P_{v+2}, \ldots, P_t.$$

Lemma 4.1 suggests how it may be possible to forge an IOBC-encrypted message so that the final block will contain the correct MDC. However, the problem remains of discovering $g(G'_{u-1}) \oplus g(G_{v-1})$, as used in constructing the forged ciphertext in the statement of the lemma. Recacha [10] discusses this very point, and explains that making this difficult motivates the inclusion of the function g in the design of IOBC — that is, if g was not included (as is the case for PES-PCBC), then simple forgeries could be achieved.

We also have the following, also implicit in Recacha's 1996 paper.

Lemma 4.2 (Mitchell, [9]) *Suppose* $[S], C_1, C_2, \ldots, C_t$ *is the encryption of* $P_1, P_2, \ldots, P_t$ *using IOBC, and that* F_i *and* G_i *are as defined in Section 4.1. Then:*

(i) $C_{j+1} \oplus P_{j+2} = g(G_j) \oplus G_{j+2}$, $1 \leq j \leq t-2$;

(ii) $\bigoplus_{i=1}^{k} g^{k-i}(C_{j+2i-1} \oplus P_{j+2i}) = g^k(G_j) \oplus G_{j+2k}$, $1 \leq j \leq t-2$, $1 \leq k \leq (t-j)/2$.

It is not hard to see that if $g^k(G_j) = G_j$ for some k, then Lemma 4.2(ii) could be combined with Lemma 4.1 to yield a forgery attack (given a ciphertext message with corresponding known plaintext). This point is made by Recacha [10], who explains that the bit permutation g has been chosen so that the smallest integer $i > 1$ such that g^i is the identity permutation is $(n/2-1)(n/2+1) = n^2/4 - 1$ (since $m = n/2$ is even). The restriction that the maximum length of messages encrypted using IOBC is $n^2/2 - 1$, as defined in Section 4.2, prevents this problem arising in practice.

However, in some cases g^k is 'close' to the identity permutation for somewhat smaller values of k. The following result highlights this for two practically important values of n. Observe that analogous results can be achieved for any n.

Lemma 4.3 (Mitchell, [9]) *Suppose X is a randomly selected n-bit block.*

(i) If $n = 64$ then $\Pr(X = g^{341}(X)) = 2^{-22}$; and

(ii) if $n = 128$ then $\Pr(X = g^{1365}(X)) = 2^{-42}$.

The above result can now be used in a straightforward way to enable message forgeries. As described in detail in [9], if $n = 64$, given an IOBC ciphertext containing at least 685 blocks and some of the corresponding plaintext, then a forged ciphertext can be created which will pass integrity checks with probability 2^{-22}. 685 is clearly much less than the defined message length limit of 2047 blocks. A precisely analogous attack works for $n = 128$, although the success probability is only 2^{-42}.

4.4 Impact

The attack outlined immediately above could be prevented by further curtailing the maximum length for messages, but this would in turn further limit the applicability of the scheme. Moreover, the lack of a formal proof of security means that other attacks are possible. Indeed, a simple chosen plaintext forgery attack was outlined in [9], although it requires approaching $2^{n/2}$ ciphertexts for chosen plaintexts (this attack is discussed further in Section 6.1 below). These points strongly argue against adoption of this scheme.

5 EPBC

5.1 EPBC operation

The EPBC scheme was proposed by Zúquete and Guedes [16] the year after the publication of IOBC. The primary design goal was to remove the message length restriction inherent in the design of IOBC; it also enables a small efficiency improvement. It was further claimed by its authors to be more secure than IOBC.

The scheme operates in a very similar way to IOBC, exactly as shown in Figure 2, and (like IOBC) requires that n is even. The only significant difference is in the choice of the function g. The function g for EPBC is not bijective, unlike in IOBC, and operates as follows. Suppose X is an n-bit block, where $X = L||R$ and L and R are m-bit blocks (and, as throughout, $||$ denotes concatenation). Then

$$g(X) = (L \vee \overline{R})||(L \wedge \overline{R})$$

where $\vee$ denotes the bit-wise inclusive or operation, $\wedge$ denotes the bit-wise logical and operation, and $\overline{X}$ denotes the logical negation of X (i.e. changing every zero to a one and vice versa).

Much like with PES-PCBC, the IVs F_0, G_0 are required to be distinct, secret initialisation values.

5.2 A flawed cryptanalysis

We first give some simple results on g.

Lemma 5.1 (Mitchell, [8]) *Suppose $g(X) = L'||R'$, where X is an n-bit block and we let $L' = (\ell'_1, \ell'_2, \ldots, \ell'_m)$ and $R' = (r'_1, r'_2, \ldots, r'_m)$ be m-bit blocks. Then, for every i $(1 \leq i \leq m)$, if $\ell'_i = 0$ then $r'_i = 0$.*

The above Lemma implies that output bit pairs (ℓ'_i, r'_i) can never be equal to (0,1). In fact, we can obtain the following more general result which gives Lemma 5.1 as a special case.

Lemma 5.2 (Mitchell, [8]) *Suppose that, as above, $X = L||R$ where $L = (\ell_1, \ell_2, \ldots, \ell_m)$ and $R = (r_1, r_2, \ldots, r_m)$. Suppose also that $g(X) = L'||R'$ where $L' = (\ell'_1, \ell'_2, \ldots, \ell'_m)$ and $R' = (r'_1, r'_2, \ldots, r'_m)$. Then if $(\ell_i, r_i) \in A$ then $(\ell'_i, r'_i) \in B$, where all possibilities for A and B are given in Table 1. Note that, for simplicity, in this table we write xy instad of (x, y).*

Table 1: Input/output possibilities for g

A (set of input pairs)	B (set of output pairs)
$\{00, 01, 10, 11\}$	$\{00, 10, 11\}$
$\{01, 10, 11\}$	$\{00, 10, 11\}$
$\{00, 10, 11\}$	$\{10, 11\}$
$\{00, 01, 11\}$	$\{00, 10\}$
$\{00, 01, 10\}$	$\{00, 10, 11\}$
$\{10, 11\}$	$\{10, 11\}$
$\{01, 11\}$	$\{00, 10\}$
$\{01, 10\}$	$\{00, 11\}$
$\{00, 11\}$	$\{10\}$
$\{00, 10\}$	$\{10, 11\}$
$\{00, 01\}$	$\{00, 10\}$
$\{11\}$	$\{10\}$
$\{10\}$	$\{11\}$
$\{01\}$	$\{00\}$
$\{00\}$	$\{10\}$

We next summarise the key part of the known-plaintext attack described in [8]. The primary objective is to use knowledge of known plaintext/ciphertext pairs (P_i, C_i) to learn the values of corresponding 'internal pairs' (F_i, G_i). These can then be used in a fairly straightforward way (as detailed in [8]) to construct a forged ciphertext which will pass the integrity checks.

It is claimed in [8] that, assuming that we have sufficiently many known plaintext and ciphertext pairs, for sufficiently large w there will only be one possibility for F_{j+2w}. Using knowledge of P_{j+2w+1}, this immediately gives certain knowledge of G_{j+2w+1}. I.e., for all sufficiently large values of w, complete knowledge can be obtained of F_{j+2w} and G_{j+2w+1}.

However, more recently, Di et al. [2] pointed out that the above analysis has a major flaw. The issue arises in the argument that, since $G_{j+1} = P_{j+1} \oplus F_j$, information about forbidden bit pairs in F_j, combined with knowledge of P_{j+1}, gives information about forbidden bit pairs in G_{j+1}. Di et al. [2] point out that if, there are two possibilities for a bit pair in F_j then there will always still be two possibilities for the corresponding bit pair in $g(G_{j+1})$ — as opposed to the analysis in [8] which suggests that the number of possibilities will be reduced to one with probability 1/6. That is, the number of possibilities for a bit pair in $g(G_{j+1})$ will never go below two, preventing the attack strategy working.

5.3 Impact

Di et al. [2] were not able to suggest any further attacks apart from a brute force approach. This suggests that EPBC may, after all, be secure. However, in the next section we show otherwise.

6 Other attacks

We now consider other possible attacks. Given that all three modes we have considered share the same underlying structure, as shown in Figure 2, we focus on attacks that apply to large classes of possibilities for the function g.

6.1 A chosen plaintext forgery attack

We start by giving an attack which will work for any function g, using a method outlined in [9] — and presented here in greater detail. This certificational chosen-plaintext-based forgery attack limits the security of any scheme using the design shown in Figure 2 (including IOBC and EPBC), regardless of length limits for plaintexts and the choice of g.

Lemma 6.1 *Suppose that $C'_1, C'_2, \ldots, C'_{t'}$ and $C_1, C_2, \ldots, C_t$ are encrypted versions of the plaintext sequences $P'_1, P'_2, \ldots, P'_{t'}$ and $P_1, P_2, \ldots, P_t$, respectively, using the same key K (although the IVs may be different). Suppose also that $P'_j = P_i$ and $C'_j = C_i$ for some $j < t'$ and $i < t$. As previously we refer to the 'internal values' computed during encryption of these two messages as F'_i, F_i, G'_i and G_i.*

Then (under reasonable assumptions about the random behaviour of the block cipher) with probability approximately 0.5 it will hold that $F'_{j-1} = F_{i-1}$, $G'_{j-1} = G_{i-1}$, $F'_j = F_i$ and $G'_j = G_i$.

Proof Let the event E_Δ be that $\Delta = F'_{j-1} \oplus F_{i-1}$. Then, under reasonable assumptions about randomness, $Pr(E_\Delta)$ is 2^{-n} for any given Δ.

In the case E_0, we immediately have $G'_{j-1} = G_{i-1}$. Also, since $P'_j = P_i$, it follows immediately that $F'_j = F_i$, $G'_j = G_i$ and $C'_j = C_i$ with probability 1.

Now consider the event E_Δ for $\Delta \neq 0$, i.e. $F'_{j-1} \neq F_{i-1}$. Since $P'_j = P_i$ this immediately implies that $G'_j \neq G_i$, and hence $F'_j \neq F_i$. Now, since $C'_j = g(G'_{j-1}) \oplus F'_j$ and $C_i = g(G_{i-1}) \oplus F_i$, we have

$$C'_j = C_i \text{ if and only if } g(G'_{j-1}) \oplus g(G_{i-1}) = F'_j \oplus F_i.$$

Under reasonable assumptions about the random behaviour of the encryption function, this will occur with probability 2^{-n}. Hence, as Δ ranges over

its 2^n possible values, the expected number of times that $C'_j = C_j$ will hold is approximately 2, one of which will occur when $F'_j = F_i$. The result follows. ∎

We can now give the following simple result that uses the same notation as Lemma 6.1. Note that it uses Lemma 4.1, which we already observed holds regardless of the choice of g.

Lemma 6.2 *Suppose that $C'_1, C'_2, \ldots, C'_{t'}$ and $C_1, C_2, \ldots, C_t$ are as in the statement of Lemma 6.1, and suppose also that $P'_j = P_i$ and $C'_j = C_i$ for some $j < t'$ and $i < t$. Then, with probability approximately 0.5, the constructed ciphertext message*

$$C'_1, C'_2, \ldots, C'_{j-1}, C_i, C_{i+1}, \ldots, C_t$$

will decrypt to $P'_1, P'_2, \ldots, P'_{j-1}, P_j, P_{j+1}, \ldots, P_t$.

Proof The result follows immediately from Lemma 4.1, putting $u = j$, $v = i$ and observing that:

$$C_v \oplus g(G'_{u-1}) \oplus g(G_{v-1}) = C_i \oplus g(G'_{j-1}) \oplus g(G_{i-1})$$

which equals C_i with probability approximately 0.5, since, by Lemma 6.1 we know that $G'_{j-1} = G_{i-1}$ with probability approximately 0.5. ∎

That is, we can construct a forged message that will pass integrity checks with probability 0.5 if we can find a pair of messages $C'_1, C'_2, \ldots, C'_{t'}$ and $C_1, C_2, \ldots, C_t$ for which $P'_j = P_i$ and $C'_j = C_i$ for some $j < t'$ and $i < t$. If the attacker can arrange for $2^{n/2}$ messages to be encrypted, all containing the same plaintext block (at a known position in each case), then by the usual 'birthday paradox' probabilistic arguments, such a pair is likely to arise. In fact, as observed in [9], the number of required message encryptions can be reduced to somewhat less than $2^{n/2}$ by including many occurrences of the fixed plaintext block in each chosen message.

Of course, this is not likely to be a realistic attack in practice; the importance of the above discussion is that it limits the level of security provided by any scheme using the general construction of Figure 2, regardless of the choice of g. In the remainder of this section we consider two attack strategies that work for two different general classes of possible functions g.

6.2 A new attack approach with implications for EPBC

We start by giving a simple generalisation of Theorem 3.1.

Theorem 6.3 *Suppose the ciphertext $C_1, C_2, \ldots, C_t$ was constructed using a scheme of the type shown in Figure 2, and that P_j is a plaintext block for some j satisfying $1 < j < t$. Suppose the $t+2$-block ciphertext $C'_1, C'_2, \ldots, C'_{t+2}$ is constructed as follows:*

$$\begin{aligned} C'_i &= C_i, \quad (1 \le i \le j), \\ C'_{j+1} &= P_j \oplus G_j \oplus g(G_j), \\ C'_i &= C_{i-2}, \quad (j+2 \le i \le t+2). \end{aligned}$$

When decrypted to yield $P'_1, P'_2, \ldots, P'_{t+2}$, the value of the final plaintext block P'_{t+2} will equal P_t for the original (untampered) message, and hence will pass the integrity check.

Remark 6.4 *In the case where g is the identity function, as is the case for PES-PCBC, then the above result reduces to Theorem 3.1.*

Proof We need only examine the decryption of C'_{j+1}; the rest of the proof is exactly as in the proof of Theorem 3.1. Now:

$$\begin{aligned} F'_{j+1} &= C'_{j+1} \oplus G'_j \\ &= (P_j \oplus G_j \oplus g(G_j)) \oplus G_j \\ &= P_j \oplus g(G_j) \\ &= F_{j-1}. \end{aligned}$$

Hence $G'_{j+1} = d_K(F'_{j+1}) = d_K(F_{j-1}) = G_{j-1}$. Finally, we have

$$P'_{j+1} = G'_{j+1} \oplus F'_j = G_{j-1} \oplus F_j = C_j \oplus G_{j-1} \oplus g(G_{j-1}),$$

although the precise value of P'_{j+1} is unimportant. The result follows trivially. ∎

Of course, the degree to which Theorem 6.3 is likely to enable a forgery attack depends very much on the properties of the function g. However, we have the following simple result for the function g used in EPBC.

Lemma 6.5 *Suppose g is as defined for EPBC, and (using the notation of Lemma 5.1) suppose also that $g(L||R) = L'||R'$. Then:*

(i) For all inputs $L||R$, we have $L \oplus L' = R \oplus R'$;

(ii) If $L||R$ is chosen at random, then each bit of $L \oplus L'$ is equal to 1 with probability 0.25.

Proof Suppose ℓ is a bit in L, and r is the corresponding bit in R. Suppose also that (ℓ',r') are the bits in the same positions in $L'||R'$. Then

$$\ell' = (\ell \vee r) \oplus \ell = \neg\ell \wedge r.$$

Also

$$r' = (\ell \wedge r) \oplus r = \neg\ell \wedge r.$$

Claim (i) follows immediately, and claim (ii) follows from observing that $\neg\ell \wedge r = 1$ if and only if $r = 1$ and $\ell = 0$. ∎

From Theorem 6.3, we can construct a possible forgery by guessing the value of $G_j \oplus g(G_j)$. If g is as defined for EPBC, then, from Lemma 6.5(i), we simply need to guess the first $n/2$ bits of $G_j \oplus g(G_j)$. Moreover, if we restrict our guesses for this 'first half' to $n/2$-bit strings containing at most $n/8$ ones (where we assume that n is a multiple of 8), then from Lemma 6.5(ii) we will have a better than evens chance of making a correct guess. The number of such strings is simply:

$$\sum_{i=0}^{n/8} \binom{n/2}{i}.$$

There are many ways of estimating this sum, but the following well known result is helpful.

Lemma 6.6 *Suppose* $m \geq 1$ *and* $0 \leq k < m/2$. *Then*

$$\sum_{i=0}^{k} \binom{m}{i} < \binom{m}{k} \frac{m-k+1}{m-2k+1}.$$

If $n = 64$ or $n = 128$ then the above sum is $1.50 \times 10^7 \simeq 2^{23.8}$ or $7.13 \times 10^{14} \simeq 2^{49.3}$, respectively. That is, for $n = 64$, after $2^{23.8}$ trials, there is a good chance one fake message will pass the integrity check, and for $n = 128$, $2^{49.3}$ trials will suffice.

Of course, these are large numbers, but they are significantly less than the certificational attack with complexity $2^{n/2}$ described in Section 6.1.

6.3 Issues with the use of Initialisation Vectors (IVs)

We next show how to construct a forgery in any scheme of the type shown in Figure 2 if the IVs (i.e. F_0 and G_0) are not different for every encrypted message and g is linear.

Before discussing the attack we briefly recap what the authors of the three schemes considered here say about the choice of IVs.

- In the paper introducing PES-PCBC, [15], there is no mention of how F_0 and G_0 are chosen — indeed, the need for them to be selected does not even appear to be mentioned. However, in the subsequent paper introducing EPBC [16], which also points out an attack on PES-PCBC, it is stated that the 'initial values of F_{i-1} and G_{i-1} are distinct, secret initialisation vectors'.

- In the paper introducing EPBC, exactly the same statement, i.e. that the 'initial values of F_{i-1} and G_{i-1} are distinct, secret initialisation vectors' is made twice, with no further guidance.

- In the IOBC paper [10], the issue is discussed in slightly greater detail. It is stated that it 'is a design requirement for IOBC that the initialising vectors ... shall be changed for each encrypted message'.

Of course, changing the values of F_0 and G_0 for each encrypted message, as required for IOBC, is clearly good practice. Indeed, if the same values are used to encrypt two messages which contain the same initial plaintext block, then the resulting ciphertexts will share the same ciphertext block. That is, the mode would leak information about plaintext, which is clearly a highly undesirable property for any mode of operation intended to provide confidentiality. Nonetheless, the designers of EPBC and PES-PCBC did not impose any requirement for the values to be changed.

We can state the following result, which is essentially a special case of Lemma 4.1, and applies to all modes adhering to the design of Figure 2 and for which g is linear (as is the case for IOBC and, trivially, PES-PCBC). In such a case, as observed in [9], the distributivity property $g(X \oplus Y) = g(X) \oplus g(Y)$ for any X and Y holds.

Lemma 6.7 *Suppose $C_1, C_2, \ldots, C_t$ and $C'_1, C'_2, \ldots, C'_{t'}$ are encrypted versions of the plaintext sequences $P_1, P_2, \ldots, P_t$ and $P'_1, P'_2, \ldots, P'_{t'}$, respectively, where the method of encryption is as shown in Figure 2. Suppose also that the two messages are encrypted using identical IVs, i.e. $F_0 = F'_0$ and $G_0 = G'_0$.*

If the ciphertext

$$C^*_1, C^*_2, \ldots, C^*_t = C'_1, C'_2, C_3 \oplus g(C_1 \oplus C'_1 \oplus P_2 \oplus P'_2), C_4, \ldots, C_t$$

is submitted for IOBC decryption (where $1 < u$ and $1 < v < t$, and G_{v-1} and G'_{u-1} are values computed during the encryption of the respective sequences of blocks), then the resulting sequence of plaintext blocks $P^*_1, P^*_2, \ldots, P^*_t$ will be equal to

$$P'_1, P'_2, P_3 \oplus F'_2 \oplus F_2, P_4, P_5, \ldots, P_t.$$

Proof First observe that, by definition, we know that

$$G_2 \oplus g(G_0) = P_2 \oplus C_1, \quad \text{and} \quad G'_2 \oplus g(G'_0) = P'_2 \oplus C'_1.$$

Hence, since we assume that $G_0 = G'_0$, we immediately have:

$$G_2 \oplus G'_2 = C_1 \oplus C'_1 \oplus P_2 \oplus P'_2,$$

and thus (using the distributive property of g):

$$g(G_2) \oplus g(G'_2) = g(C_1 \oplus C'_1 \oplus P_2 \oplus P'_2).$$

The result follows from Lemma 4.1, setting $u = v = 3$. ∎

Thus if two ciphertexts are encrypted using the same IV, and a single plaintext block is known for each message, then a forgery can be constructed. Observe that this attack can be extended using Lemma 4.2ii.

It may well be the case that other forgery strategies can be devised building on the distributive property when g is linear, but we do not explore this further here.

7 Other related modes

We conclude this discussion of modes by briefly mentioning two further modes 'from the same stable'.

There was a gap of some 16 years before the first of these two additional schemes was made public — IOC (short for *Input and Output Chaining*) was made public by Recacha in 2013 [11]. IOC is clearly closely related to IOBC and EPBC, but was designed to avoid the known issues with these schemes. IOC was made public at a time when there was a renewed interest in the area, at least partly prompted by a NIST initiative on lightweight cryptography[2]. A slightly revised version was made public in early 2p14, [14]. Cryptanalyses of both versions of the scheme first appeared in 2014 [1].

The second additional scheme, known as ++AE, is a further evolution of the previous schemes, again designed with the intention of addressing the known issues. Like IOC, this scheme exists in two slightly different versions, v1.0 [12] and v1.1 [13], both promulgated in 2014. Both versions were cryptanalysed in a pair of papers published in 2016 and 2018 [4, 5].

[2] As discussed at `https://csrc.nist.gov/Projects/Lightweight-Cryptography`, NIST began investigating cryptography for constrained environments in 2013, and one of the goals was to find lightweight methods for authenticated encryption.

8 Summary and conclusions

In this paper we have re-examined the security of three closely related block cipher modes of operation designed to provide authenticated encryption, namely PES-PCBC, IOBC and EPBC. Whilst cryptanalysis of all three schemes has previously been published, the attack on EPBC has subsequently been shown to be incorrect and hence until now no effective attack was known against this mode.

In this paper we have both elaborated on and enhanced existing cryptanalysis, and we have also demonstrated new attacks which show that none of the three schemes can be considered secure. The main findings of the paper are as follows.

- There exists a forgery attack on any scheme of the type shown in Figure 2 which requires of the order of $2^{n/2}$ chosen plaintexts — see Section 6.1.
- It was already known (see [16]) that simple forgeries against PES-PCBC could be devised requiring a single ciphertext message and two known plaintext blocks for this ciphertext — a variant of this attack was described (see Section 3.1 requiring only a single known plaintext block. Another simple forgery attack against PES-PCBC was described in Section 6.3, which is realisable if IVs are ever re-used.
- Di et al. [2] showed that the only known attack on EPBC mode did not work. However, in Section 6.2 we described a new attack strategy which yields a successful forgery with high probability with significantly fewer than $2^{n/2}$ trials (e.g. $2^{23.8}$ trials for $n = 64$).
- Forgery attacks on IOBC were already known (see [9]). A further simple forgery attack was described in Section 6.3, which is realisable if IVs are ever re-used (although it is important to note that IV reuse is specifically prohibited in [10]).

Existing cryptanalysis, when combined with the new attacks described in this paper, suggests very strongly than none of the three modes considered in this paper are sufficiently robust against forgery attacks to be used in practice.

Dedication

This paper is dedicated to the memory of Ed Dawson.

References

[1] P. Bottinelli, R. Reyhanitabar, and S. Vaudenay. Breaking the IOC authenticated encryption mode. In D. Pointcheval and D. Vergnaud, editors, *Progress in Cryptology — AFRICACRYPT 2014 — 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28–30, 2014. Proceedings*, volume 8469 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 2014.

[2] B. Di, L. Simpson, H. Bartlett, E. Dawson, and K. K.-H. Wong. Correcting flaws in Mitchell's analysis of EPBC. In I. Welch and X. Yi, editors, *13th Australasian Information Security Conference, AISC 2015, Sydney, Australia, January 2015*, volume 161 of *CRPIT*, pages 57–60. Australian Computer Society, 2015.

[3] J. T. Kohl. The use of encryption in Kerberos for network authentication. In G. Brassard, editor, *Advances in Cryptology — CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 35–43. Springer-Verlag, Berlin, 1990.

[4] H. Q. Al Mahri, L. Simpson, H. Bartlett, E. Dawson, and K. K.-H. Wong. A fundamental flaw in the ++AE authenticated encryption mode. *J. Math. Cryptol.*, 12(1):37–42, 2018.

[5] H. Q. Al Mahri, L. R. Simpson, H. Bartlett, E. Dawson, and K. K.-H. Wong. Forgery attacks on ++AE authenticated encryption mode. In *ACSW '16: Proceedings of the Australasian Computer Science Week Multiconference, Canberra, Australia, February 2–5, 2016*, pages 1–9. ACM, 2016.

[6] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.

[7] C. J. Mitchell. Cryptanalysis of two variants of PCBC mode when used for message integrity. In C. Boyd and J. M. Gonzalez Nieto, editors, *Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4–6 2005, Proceedings*, number 3574 in Lecture Notes in Computer Science, pages 560–571. Springer-Verlag, Berlin, 2005.

[8] 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.

[9] C. J. Mitchell. Analysing the IOBC authenticated encryption mode. In C. Boyd and L. Simpson, editors, *Information Security and Privacy — 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1–3, 2013. Proceedings*, volume 7959 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013.

[10] 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.

[11] F. Recacha. IOC: The most lightweight authenticated encryption mode? Available at `https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/ioc/ioc-spec.pdf`, March 2013.

[12] F. Recacha. ++AE v1.0. Available at `https://competitions.cr.yp.to/round1/aev10.pdf`, March 2014.

[13] F. Recacha. ++AE v1.1. Available at `https://competitions.cr.yp.to/round1/aev11.pdf`, April 2014.

[14] F. Recacha. Input output chaining (IOC) AE mode revisited. Available at `https://inputoutputblockchaining.blogspot.com/`, January 2014.

[15] A. Zuquete and P. Guedes. Transparent authentication and confidentiality for stream sockets. *IEEE Micro*, 16(3):34–41, May/June 1996.

[16] 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.