Cryptanalysis of the EPBC authenticated encryption mode

Chris J. Mitchell

Information Security Group, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
c.mitchell@rhul.ac.uk

Abstract. A large variety of methods for using block ciphers, so called 'modes of operation', have been proposed, including some designed to provide both confidentiality and integrity protection. Such modes, usually known as 'authenticated encryption' modes, are increasingly important given the variety of issues now known with the use of unauthenticated encryption. In this paper we show that a mode known as EPBC (Efficient error-Propagating Block Chaining), proposed in 1997 by Zúquete and Guedes, is insecure. Specifically we show that given a modest amount of known plaintext for a single enciphered message, new enciphered messages can be constructed which will pass tests for authenticity. That is, we demonstrate a message forgery attack.

1 Introduction

Traditionally, the recommended way to use a block cipher to provide both integrity and confidentiality protection for a message has been to encrypt the data and then compute a CBC-MAC on the encrypted data, using two distinct secret keys. This approach is rather unattractive for some applications because it requires each block of data to be processed twice. This observation has given rise to a number of proposals for combining encryption and integrity protection (see, for example, Sect. 9.6 of [1]).

At the same time, in recent years two major problems have been identified which have highlighted the need for better-defined integrity and confidentiality modes. Firstly, issues have been identified with certain combinations of encryption and use of a CBC-MAC — see, for example, Bellare, Kohno and Namprempre [2]. That is, it is vital to define precisely how the two operations are combined, including the order of the computations; otherwise there is a danger of possible compromise of the data. Secondly, even where integrity is not explicitly required by the application, if integrity is not provided then in some cases padding oracle attacks may be used to compromise secret data (see, for example, [3–7]).

This has given rise to a number of proposals for well-defined authenticated-encryption modes, including OCB [8], EAX [9] and CCM [10, 11]. These techniques are also the subject of ongoing international standardisation efforts — the third committee draft of what is intended to become ISO/IEC 19772 on authenticated encryption was published in June 2007 [12] (see also Dent and Mitchell, [13]).

In this paper we examine another authenticated-encryption mode, known as EPBC, which was introduced by Zúquete and Guedes in 1997 [14]. We show that this mode is subject to a message forgery attack using only a modest amount of known plaintext, and hence does not provide adequate integrity protection. When combined with other recent cryptanalyses of authenticated-encrypted modes [15], this emphasises the need to only use modes which have robust evidence for their security, e.g. OCB, EAX or CCM.

2 Integrity Protection

Before discussing any possible attacks, we need to explain how EPBC mode is intended to be used to provide both confidentiality and integrity protection. The idea is very simple. First divide the data to be encrypted into a sequence of n-bit blocks, padding as necessary, where n is the block length for the block cipher in use. Then append an additional n-bit block to the end of the message, where this block can be predicted by the decrypter (e.g. a fixed block); this is referred to as the *integrity control value* by Zúquete and Guedes [14]. When the message is decrypted, a check is made that the final block is the expected value and, if it is, the message is deemed authentic.

Before proceeding observe that this general approach possesses an intrinsic weakness. That is, suppose that a fixed final block (the *terminator block*) is used to detect message manipulations (as above). Then an attacker might be able to persuade the legitimate originator of protected messages to encrypt a message which contains the fixed terminator block somewhere in the middle of the message. The attacker will then be able to delete all ciphertext blocks following the encrypted terminator block, and such a change will not be detectable.

Despite this weakness, using an appropriate encryption mode combined with a method for adding verifiable redundancy to a message is still used for message integrity protection — e.g. in Kerberos (see, for example, [13]). As far as this paper is concerned we note that such an attack could be prevented by ensuring that the legitimate encrypter refuses to encrypt any plaintext message containing the terminator block. We further note that such an attack requires *chosen* plaintext, and the attack we demonstrate later in this paper requires only a limited amount of *known* plaintext.

3 The Zúquete-Guedes EPBC Mode

First suppose that the data is to be protected using an n-bit block cipher, i.e. a block cipher operating on plaintext and ciphertext blocks of n bits. We further suppose that n is even, and put $n = 2m$ (as is the case for all standardised block ciphers — see, for example, [16]). We write $e_K(P)$ for the result of block cipher encrypting the n-bit block P using the secret key K, and $d_K(C)$ for the result of block cipher decrypting the n-bit block C using the key K. Suppose the plaintext to be protected is divided into a sequence of n-bit blocks (if necessary, having first been padded): $P_1, P_2, \ldots, P_t$.

The scheme uses two secret n-bit Initialisation Vectors (IVs), denoted by F_0 and G_0. The EPBC encryption of the plaintext $P_1, P_2, \ldots, P_t$ is then defined as:

$$G_i = P_i \oplus F_{i-1}, \quad (1 \leq i \leq t), \tag{1}$$
$$F_i = e_K(G_i), \quad (1 \leq i \leq t), \tag{2}$$
$$C_i = F_i \oplus g(G_{i-1}), \quad (2 \leq i \leq t), \tag{3}$$

where $C_1 = F_1 \oplus G_0$, $\oplus$ denotes bit-wise exclusive-or, 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 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.

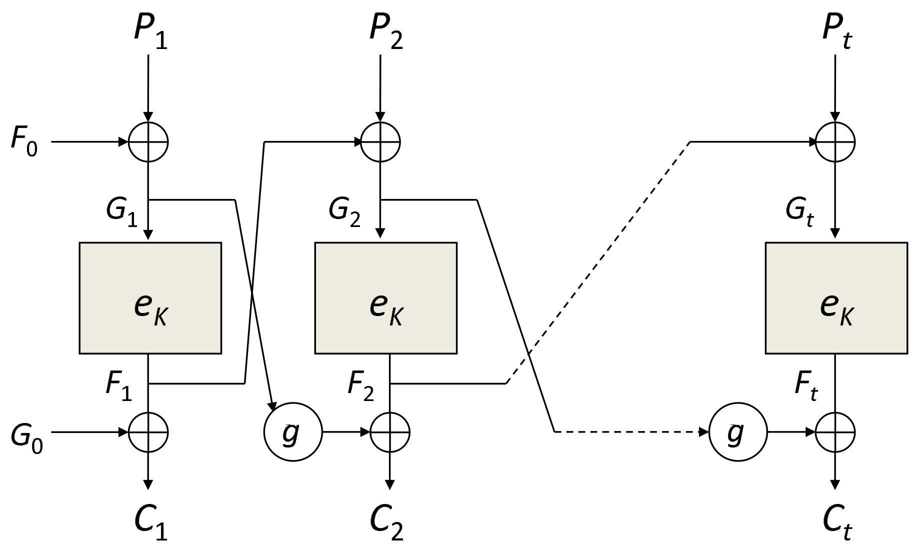

Fig. 1. EPBC encryption

The function g is defined 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).

Finally, note that decryption operates similarly. We have:

$$F_i = C_i \oplus g(G_{i-1}), \quad (2 \le i \le t), \tag{4}$$
$$G_i = d_K(F_i), \quad (1 \le i \le t), \tag{5}$$
$$P_i = G_i \oplus F_{i-1}, \quad (1 \le i \le t). \tag{6}$$

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

4 Some Preliminary Observations

We first establish some simple results on the operation of the EPBC scheme. In particular, we consider the operation of the function g.

Lemma 1. *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 $(1 \le i \le m)$, if $\ell'_i = 0$ then $r'_i = 0$.*

Proof Let $X = L||R$, where $L = (\ell_1, \ell_2, \ldots, \ell_m)$ and $R = (r_1, r_2, \ldots, r_m)$. Suppose $\ell'_i = 0$ for some i. But, by definition, $\ell'_i = \ell_i \vee \overline{r_i}$, and hence $\ell_i = \overline{r_i} = 0$. Hence $r'_i = \ell_i \wedge \overline{r_i} = 0$. □

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 1 as a special case.

Lemma 2. *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).*

Proof The result follows from a simple case by case analysis. □

Before proceeding we make a general observation which underlies the attack procedure described below. That is, given a random set A of any particular size (not equal to 1), then the expected value of $|B|$ is always smaller than $|A|$.

5 Attack Stage 1 — Deducing Internal Pairs

The objective of this stage of the attack 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). In the second stage of the attack, described in Sect. 6, we show how to use these internal values to complete a forgery attack on EPBC mode.

Suppose an attacker knows s consecutive pairs of plaintext/ciphertext blocks for some $s > 1$. That is, suppose

$$(P_j, C_j), (P_{j+1}, C_{j+1}), \ldots, (P_{j+s-1}, C_{j+s-1})$$

Table 1. Input/output possibilities for g

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

are known, where we also suppose that $j > 1$.

First observe that we know that $C_j = F_j \oplus g(G_{j-1})$ (since $j > 1$). From Lemma 1, we also know that, if $g(G_{j-1}) = L' || R'$ where $L' = (\ell'_1, \ell'_2, \ldots, \ell'_m)$ and $R' = (r'_1, r'_2, \ldots, r'_m)$, then $(\ell'_i,\, r'_i)$ can never equal (0,1) for any i. Hence knowledge of C_j will immediately yield knowledge about F_j. Specifically, it will yield the information that certain bit pairs cannot occur in F_j, where each bit pair contains a bit from the left half and the corresponding bit from the right half. More precisely, given that there are m such bit pairs in F_j, and for each such pair one of the four possible bit pairs will be ruled out, the number of possibilities for F_j will be reduced from 2^{2m} to 3^m.

Using Lemma 2, we can extend this observation making use of subsequent plaintext/ciphertext block pairs. 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}. This yields information about (potentially) even more forbidden bit pairs in $g(G_{j+1})$. Given that

$$C_{j+2} = F_{j+2} \oplus g(G_{j+1}),$$

and given knowledge of C_{j+2}, this gives even more information about forbidden bit pairs in F_{j+2}, and so on.

That is, it is possible to deduce increasing amounts of information about the sequence of n-bit blocks:

$$F_j, g(G_{j+1}), F_{j+2}, g(G_{j+3}), \ldots.$$

Hence, assuming that we know sufficiently many pairs to perform the calculations, 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}.

In the above discussion we did not use all the available information to make the deductions. In fact we only used knowledge of $C_j, C_{j+2}, C_{j+4}, \ldots$ and $P_{j+1}, P_{j+3}, P_{j+5}, \ldots$. We have also only shown how to derive information about $F_j, F_{j+2}, F_{j+4}, \ldots$ and $G_{j+1}, G_{j+3}, G_{j+5}, \ldots$.

However, we can simply repeat the above argument starting with F_{j+1}, using the remainder of the information available. That is, repeating the process starting one block later will enable the deduction of information about $F_{j+1}, F_{j+3}, F_{j+5}, \ldots$ and $G_{j+2}, G_{j+4}, G_{j+6}, \ldots$. Finally note that the above analysis does not make use of knowledge of P_j, only of C_j.

The above discussion has been rather informal, in that 'sufficiently large' has not been quantified. However, Lemma 2 enables us to make this more precise.

Consider any pair of bit positions in an n-bit block: $(i,\ i+m)$, say, where $1 \leq i \leq m$. Then, from Lemma 1, we know that $g(G_{j-1})$ cannot have (0,1) in these two bit positions. Hence, given knowledge of C_j, we know that the bits in positions $(i,\ i+m)$ in $F_j = C_j \oplus g(G_{j-1})$ can only take three of the four possible values. Precisely which three possibilities will depend on the pair of bits in positions $(i,\ i+m)$ in C_j, which we assume are randomly distributed.

As a result we know that the bits in positions $(i,\ i+m)$ in G_{j+1} can only take three of the four possible values. From an examination of Table 1, the number of possibilities for the bits in positions $(i,\ i+m)$ in $g(G_{j+1})$ will either be two or three, depending on the three possibilities for the bit pair in G_{j+1}. Specifically, there is a 50% chance that there will only be two possibilities for the bit pair in G_{j+1}, and a 50% that there will be three possibilities for the bit pair in G_{j+1}.

Extending this analysis using standard probabilistic arguments for stochastic processes, it follows that the probability p that there will only be a single possibility for the bit pair after v iterations of the above process is equal to the bottom left entry in the vth power of the four by four matrix given in Figure 2. The entry in the i row and the jth column of this matrix represents the probability that a set A of size i will map to a set B of size j (as derived from Table 1). Some values of this matrix entry (i.e. of p) for various values of v are given in Table 2.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/6 & 5/6 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 2. Transition probability matrix

Table 2. Probability of a unique possibility for a bit pair

v	prob. p	v	prob. p	v	prob. p
10	0.71027	20	0.95305	30	0.99241
40	0.99878	50	0.99980	60	0.99997

That is, after 30 iterations, i.e. given knowledge of 60 consecutive plaintext/ciphertext pairs, the probability p that a bit pair will be known with certainty is 0.99241. We are actually interested in the probability q that an entire n-bit block will be known with certainty. It is straightforward to verify that

$$q = p^{n/2}.$$

In Table 3, this probability is tabulated for the same values of v as given in Table 2, assuming the use of a 128-bit block cipher.

Table 3. Probability of a unique possibility for a 128-bit block

v	prob. q	v	prob. q	v	prob. q
10	very small	20	0.04607	30	0.61409
40	0.92485	50	0.98728	60	0.99808

Hence, for such a cipher, if 60 consecutive pairs of plaintext/ciphertext blocks are known (bearing in mind that each iteration involves alternate blocks), then there is a 60% chance that the final internal variables F_i and G_i will be completely known. This probability increases to nearly 99% if 100 consecutive block pairs are available.

6 Attack Stage 2 — Completing the Forgery

We now suppose that the attack procedure described in the previous section has been completed, i.e. matching pairs of consecutive plaintext/ciphertext blocks have been used to learn internal values G_i, for some i. We suppose also that the EPBC mode is being used to provide both confidentiality and integrity using a fixed n-bit integrity control value V. That is, when a message is decrypted, the final plaintext block must equal V if the message is to be accepted as genuine. To demonstrate a forgery attack, we therefore need to show how to construct a message for which this will be true.

We suppose that the following resources are available to an attacker.

- An encrypted message $C_1, C_2, \ldots, C_t$ for which the attacker knows the internal value G_s, for some $s \leq t$.

– The final two blocks $(C'_{u-1},\ C'_u)$ of an encrypted message for which the attacker also knows the internal value G'_{u-2}. Note that we are assuming that $P'_u = V$, where P'_u is the final plaintext block corresponding to this enciphered message.

Note that we assume also that the same secret key K has been used to compute both the ciphertexts involved (in fact, the same message could be used to yield both of the sets of values required).

We now define the 'forged' ciphertext message $C^*_1, C^*_2, \ldots, C^*_{s+2}$ as follows:

$$\begin{aligned}
C^*_i &= C_i, \quad (1 \le i \le s),\\
C^*_{s+1} &= C'_{u-1} \oplus g(G'_{u-2}) \oplus g(G_s), \quad \text{and}\\
C^*_{s+2} &= C'_u.
\end{aligned}$$

It remains to show that, when the above forged message is decrypted, the final recovered plaintext block will equal V. Suppose that, when decrypting $C^*_1, C^*_2, \ldots, C^*_{s+2}$, the internal values are F^*_i and G^*_i $(1 \le i \le s+2)$.

We first note that it follows immediately from the definitions that $F^*_i = F_i$ and $G^*_i = G_i$ $(1 \le i \le s)$, where F_i and G_i are the internal values generated during the encryption process that yielded the ciphertext message $C_1, C_2, \ldots, C_t$. We now consider the decryption of C^*_{s+1}.

We have

$$\begin{aligned}
F^*_{s+1} &= C^*_{s+1} \oplus g(G^*_s) \quad \text{(from Sect. 3)}\\
&= C'_{u-1} \oplus g(G'_{u-2}) \oplus g(G_s) \oplus g(G^*_s) \quad \text{(by defn. of } C^*_{s+1})\\
&= C'_{u-1} \oplus g(G'_{u-2}) \quad \text{(since } G^*_s = G_s)\\
&= F'_{u-1}.
\end{aligned}$$

Hence $G^*_{s+1} = G'_{u-1}$.

We now consider the decryption of C^*_{s+2}. We have

$$\begin{aligned}
F^*_{s+2} &= C^*_{s+2} \oplus g(G^*_{s+1}) \quad \text{(from Sect. 3)}\\
&= C'_u \oplus g(G'_{u-1}) \quad \text{(by defn. of } C^*_{s+2})\\
&= F'_u.
\end{aligned}$$

Hence $G^*_{s+2} = G'_u$. Finally, we have

$$\begin{aligned}
P^*_{s+2} &= G^*_{s+2} \oplus F^*_{s+1} \quad \text{(from Sect. 3)}\\
&= G'_u \oplus F'_{u-1} \quad \text{(from above)}\\
&= P'_u\\
&= V,
\end{aligned}$$

as required.

7 Further Observations

7.1 Attack Performance

It is of interest to try to understand the overall complexity of the attack described above, i.e. to understand what information is required to complete an attack, and what computations need to be performed. For a 128-bit block cipher, we have shown in Sect. 5 that knowledge of 80 consecutive pairs of known plaintext/ciphertext blocks will be very likely to yield certain knowledge of the 'final' internal variable G_i.

Thus, if the attacker has access to two messages encrypted using the same key, and the attacker knows the final 80 plaintext blocks for both messages (as well as the ciphertext), then the attacker will almost certainly have sufficient information to perform the procedure described in Sect. 6. The computations involved in both stages of the attack are trivial — indeed, they involve only a small number of computations of the (very simple) function g, and some exclusive-or operations on pairs of blocks.

Thus the attack will have a relatively high success probability given only a modest amount of known plaintext, and the computations involved are completely trivial.

7.2 The Choice for the Function g

As should be clear from the discussions above, completing the attack as described in Sect. 6 requires knowledge of internal variables G_i. It is possible to learn these values as a result of the procedure described in Sect. 5; this procedure only works because the function g is non-bijective, and has a very simple structure. This raises two questions.

Firstly, why is the function g chosen to be non-bijective? Secondly, if there is a good reason to use a non-bijective function, then why not use one with less obvious structure?

The answer to the first question can be found in the original paper of Zúquete and Guedes [14]. As stated in [14], EPBC is very similar to a previously devised mode called IOBC, proposed by Recacha in 1996[1]. Indeed, the only difference is that the function g used in IOBC is bijective, involving some fixed bit permutations. However, as very briefly outlined in [14], IOBC is subject to known-plaintext attacks if the message being encrypted contains more than $n^2/4$ blocks, where n is the block cipher block length, i.e. around 4000 blocks for AES. The exact attack approach is not clear from [14], which states that a detailed description can be found in the 1996 paper of Recacha. Because of the existence of this attack, EPBC was designed to use a non-bijective function g, which (apparently) rules out the known-plaintext attacks applying to IOBC.

The answer to the second question is less clear. Obviously, g could be implemented using one or more block cipher encryptions, which, if done correctly,

[1] Unfortunately, the 1996 Recacha paper cited in [14] does not appear to be readily available.

would certainly remove the simple structures exploited in Sect. 5. However, such an approach would significantly increase the complexity of the mode of operation, and one of the main design goals was to devise a scheme with minimal complexity. Designing a function g which is both sufficiently complex to prevent attack, but is nevertheless very fast to perform would, perhaps, be an interesting research question; however, given that highly efficient provably secure authenticated encryption modes are now known to exist (as discussed in Sect. 1), this is probably not likely to be a particularly fruitful line of enquiry.

8 Summary and Conclusions

In this paper we have demonstrated a forgery attack against EPBC mode when used to provide message integrity. This attack required only known plaintext (and no chosen plaintext). We can therefore conclude that this mode is unacceptably weak, and should therefore not be used. (Whilst it is probably an effective mode for encryption only, much simpler modes are known for this purpose).

If both confidentiality and integrity protection are required, then encryption and a MAC should be combined in an appropriate way, or a dedicated 'authenticated encryption' mode should be used — see, for example, ISO/IEC 19772 [12].

Acknowledgements

The author would like to thank Po Yau for his generous help in performing numerical computations, and anonymous referees for valuable comments which have improved the paper.

References

1. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
2. Bellare, M., Kohno, T., Namprempre, C.: Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-MAC paradigm. ACM Transactions on Information and System Security **7** (2004) 206–241
3. Black, J., Urtubia, H.: Side-channel attacks on symmetric encryption schemes: The case for authenticated encryption. In: Proceedings of the 11th USENIX Security Symposium, San Francisco, CA, USA, August 5-9, 2002, USENIX (2002) 327–338
4. Canvel, B., Hiltgen, A., Vaudenay, S., Vuagnoux, M.: Password interception in a SSL/TLS channel. In Boneh, D., ed.: Advances in Cryptology — CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Volume 2729 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2003) 583–599

5. Paterson, K.G., Yau, A.: Padding oracle attacks on the ISO CBC mode padding standard. In Okamoto, T., ed.: Topics in Cryptology — CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. Volume 2964 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2004) 305–323
6. Vaudenay, S.: Security flaws induced by CBC padding — Applications to SSL, IPSEC, WTLS . . . . In Knudsen, L., ed.: Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 – May 2, 2002, Proceedings. Volume 2332 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2002) 534–545
7. Yau, A.K.L., Paterson, K.G., Mitchell, C.J.: Padding oracle attacks on CBC-mode encryption with secret and random IVs. In Gilbert, H., Handschuh, H., eds.: Fast Software Encryption, 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Papers. Number 3557 in Lecture Notes in Computer Science, Springer-Verlag, Berlin (2005) 299–319
8. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security **6** (2003) 365–403
9. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In Roy, B., Meier, W., eds.: Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers. Volume 3017 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2004) 389–407
10. National Institute of Standards and Technology (NIST): NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality. (2004)
11. Whiting, D., Housley, R., Ferguson, N.: RFC 3610, Counter with CBC-MAC (CCM). Internet Engineering Task Force. (2003)
12. International Organization for Standardization Genève, Switzerland: ISO/IEC 3rd CD 19772, Information technology — Security techniques — Authenticated encryption mechanisms. (2007)
13. Dent, A.W., Mitchell, C.J.: User's Guide to Cryptography and Standards. Artech House (2005)
14. Zuquete, A., Guedes, P.: Efficient error-propagating block chaining. In Darnell, M., ed.: Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17–19, 1997, Proceedings. Number 1355 in Lecture Notes in Computer Science, Springer-Verlag, Berlin (1997) 323–334
15. Mitchell, C.J.: Cryptanalysis of two variants of PCBC mode when used for message integrity. In Boyd, C., Gonzalez Nieto, J.M., eds.: Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4–6 2005, Proceedings. Number 3574 in Lecture Notes in Computer Science, Springer-Verlag, Berlin (2005) 560–571
16. International Organization for Standardization Genève, Switzerland: ISO/IEC 18033–3, Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers. (2005)