Correcting flaws in Mitchell's analysis of EPBC

Binbin Di, Leonie Simpson, Harry Bartlett, Ed Dawson, Kenneth Wong
Science and Engineering Faculty
Queensland University of Technology
2 George St, Brisbane 4000, QLD

binbin.di@hdr.qut.edu.au, lr.simpson@qut.edu.au, h.bartlett@qut.edu.au, e.dawson@qut.edu.au, kk.wong@qut.edu.au

Abstract

Efficient error-Propagating Block Chaining (EPBC) is a block cipher mode intended to simultaneously provide both confidentiality and integrity protection for messages. Mitchell pointed out a weakness in EPBC and claimed that this permits a forgery attack. This paper corrects a flaw in Mitchell's analysis and shows that the attack is no better than brute force of the integrity check vector.

Keywords: block cipher, authenticated encryption, EPBC, forgery attack.

1 Introduction

Efficient error-Propagating Block Chaining (EPBC) (Zuquete and Guedes 1997) is a mode of operation for block ciphers that is intended to provide authenticated encryption (AE). EPBC can be used with any block cipher. The plaintext is divided into blocks as defined by the selected block cipher. A predefined Integrity Check Vector (ICV) is appended to the plaintext message and the message is then encrypted in EPBC mode. When the ciphertext is decrypted, the receiver checks the correctness of ICV. Any change to the ciphertext should propagate erroneous decryptions to all subsequent ciphertext blocks, resulting in the decryption to an incorrect ICV (Zuquete and Guedes 1997), as shown in Figure 1. Messages with an incorrect ICV are rejected by the receiver.

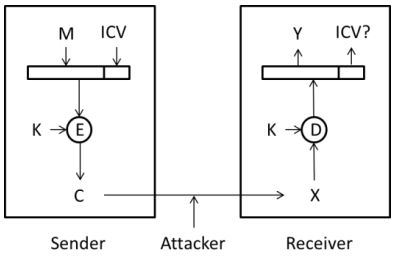

Figure 1: Integrity mechanism (Recacha 1996)

Recently, Mitchell analysed EPBC, pointing out a weakness in the integrity mechanism and proposed a forgery attack based on this weakness (Mitchell 2007). He claimed that knowing sufficient plaintext/ciphertext pairs permitted the inner vectors, used to conceal plaintext patterns, to be disclosed with very high probability. Once these inner vectors were known, a forgery could be constructed. However, we show that his calculation is inaccurate, and the probability of a successful forgery is no better than that of guessing the ICV.

2 Description of EPBC

EPBC is a mode of operation for an n-bit block cipher, for even n, say $n = 2m$. Two secret keys denoted K and K' are used. One key, K, is used for encryption and decryption. Let $e_K(P)$ denote the encryption of the plaintext block P and $d_K(C)$ denote the decryption of the ciphertext block C under the key K. The second secret key, K', and a sequence number S are used to generate a pair of secret n-bit initial vectors denoted by F_0 and G_0, where $F_0 = e_{K'}(S)$ and $G_0 = e_{K'}(F_0)$, which are used for encryption and decryption of the first block.

The EPBC encryption operation is defined as follows:

$$G_i = P_i \oplus F_{i-1},\ 1 \le i \le u,$$

$$F_i = e_K(G_i),\ 1 \le i \le u,$$

$$C_i = F_i \oplus g(G_{i-1}),\ 2 \le i \le u,$$

where $C_1 = F_1 \oplus G_0$ and g is a function applied to the two m-bit halves of the n-bit block. More precisely, suppose X is an n-bit block, where $X = L \| R$, L is the high order m-bit block and R is the low order m-bit block ($\|$ denotes concatenation). Then g is defined as follows:

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

where $\vee$ and $\wedge$ denote the bitwise inclusive or and logical and operations respectively, and $\bar{X}$ denotes the bitwise inverse version of X. This process is shown in Figure 2.

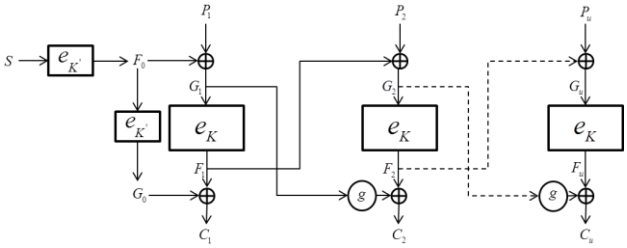

Figure 2: EPBC encryption (Mitchell 2007)

The decryption operation is simply a reverse process of the encryption as follows:

$$F_i = C_i \oplus g(G_{i-1}),\ 2 \le i \le u,$$

$$G_i = d_K(F_i), 1 \le i \le u ,$$
$$P_i = G_i \oplus F_{i-1}, 1 \le i \le u .$$

where $F_1 = C_1 \oplus G_0$.

Verifying the integrity is done simply by checking the last l bits of recovered plaintext (where l is the length of the ICV). If this matches the expected value of the ICV, the message is regarded as authentic.

3 Review of Mitchell's analysis

Mitchell's forgery attack (Mitchell 2007) on EPBC aims to forge a ciphertext in such a way that the forgery is not detected by the integrity mechanism. This is an existential forgery (Preneel 1998). In order to achieve this, the attacker has to construct a message such that the last ciphertext block will decrypt to the correct ICV value. The inner vectors, F_i and G_i, in EPBC ensure the integrity protection by propagating inaccurate decryptions from any tampered ciphertext blocks through to the end (Zuquete and Guedes 1997). Zuquete and Guedes (1997) note that the forged ciphertext blocks must be constructed to adjust values of the inner vectors during the decryption process, to permit correct decryption of the ICV.

The function g in EPBC is critical in protecting the contents of the inner vectors from discovery. Mitchell's analysis is composed of two stages: investigating a vulnerability of the function g which can be used to reveal the inner vectors and then using this knowledge to construct a message which will not be detected as a forgery by the integrity mechanism of EPBC.

3.1 Mitchell's analysis of function g

This stage aims to use knowledge of a series of plaintext and ciphertext pairs to disclose the inner vectors, G_i. Knowledge of G_i permits a forgery attack on EPBC mode. The process of constructing a forged ciphertext is outlined in Sect. 4.

Properties of the function g are used to reveal the contents of the inner vectors G_i. Suppose X is an n-bit block, where $X = L \| R$, $L = (x_1, x_2, ..., x_m)$ and $R = (x_{m+1}, x_{m+2}, ..., x_{m+m})$. Also, suppose $g(X) = L' \| R'$ where $L' = (x_1', x_2', ..., x_m')$ and $R' = (x_{m+1}', x_{m+2}', ..., x_{m+m}')$. Because g applies bitwise operations to the two m-bit halves (L and R) of each block, it can be treated as m parallel operations on pairs of bits (x_j, x_{j+m}), where x_j is the j-th bit of the block and x_{j+m} is the $(j+m)$-th bit of the block, for $j = 1, ..., m$. Table 1 (modified from (Mitchell, 2007)) shows the set B of possible output pairs (x_j', x_{j+m}') that can be obtained after applying g to each possible set A of input pairs (x_j, x_{j+m}). Sets in column A are grouped by the number of alternatives in each set. We will explain later why we have separated group 2 into subsets $2a$ and $2b$.

Assume that a set of staggered plaintext/ciphertext blocks $(C_i, P_{i+1}), (C_{i+2}, P_{i+3}), ...$ are known by the attacker. Assume also (x_j, x_{j+m}) is a bit pair in inner vector G_{i-1}, where j $(1 \le j \le m)$ is a randomly chosen bit position in an n-bit block. There are four possible values for this bit pair (listed as group 4 in Table 1). Mitchell (2007) notes that the set of output bit pairs from the function g can never include the specific bit pair (0, 1). Thus, the pair in position (x_j, x_{j+m}) of $g(G_{i-1})$ can only take one of the values listed (for group 4) in column B of Table 1. Because of this, we can also narrow possible bit pairs in position (x_j, x_{j+m}) in $G_{i+1} = P_{i+1} \oplus C_i \oplus g(G_{i-1})$ from four to three, where the bits in position (x_j, x_{j+m}) of $C_i \oplus P_{i+1}$ determine which set of three pairs is relevant in each individual case. Similarly, when G_{i+1} runs through the function g, either three (50% chance) or two (50% chance) alternatives result for the bit pairs in position (x_j, x_{j+m}) in $g(G_{i+1})$.

Group	Input pairs A	Output pairs B
4	(0, 0) (0, 1) (1, 0) (1, 1)	(0, 0) (1, 0) (1, 1)
3	(0, 1) (1, 0) (1, 1) (0, 0) (1, 0) (1, 1) (0, 0) (0, 1) (1, 1) (0, 0) (0, 1) (1, 0)	(0, 0) (1, 0) (1, 1) (1, 0) (1, 1) (0, 0) (1, 0) (0, 0) (1, 0) (1, 1)
2b	(1, 0) (1, 1) (0, 1) (1, 1) (0, 0) (1, 0) (0, 0) (0, 1)	(1, 0) (1, 1) (0, 0) (1, 0) (1, 0) (1, 1) (0, 0) (1, 0)
2a	(0, 1) (1, 0) (0, 0) (1, 1)	(0, 0) (1, 1) (1, 0)
1	(1, 1) (1, 0) (0, 1) (0, 0)	(1, 0) (1, 1) (0, 0) (1, 0)

Table 1: Input/output possibilities for the function g (Mitchell, 2007, modified)

If the bit pair in position (x_j, x_{j+m}) of $g(G_{i+1})$ has two alternatives, so will the bit pair in this position in $G_{i+3} = P_{i+3} \oplus C_{i+2} \oplus g(G_{i+1})$. Mitchell (2007) argues that the output bit pairs in $g(G_{i+3})$ will then either have two alternatives (5/6 chance) or one alternative (1/6 chance). Finally, if there is one alternative in the input, the output pairs of the function g have only one alternative. According to this argument, the possible alternatives for each bit pair in G_{i+2v-1} will eventually be reduced to a single (known) alternative if sufficiently many staggered plaintext/ciphertext pairs $(C_i, P_{i+1}), ..., (C_{i+2v-2}, P_{i+2v-1})$ are known.

Based on Table 1, Mitchell (2007) proposed a matrix (shown in Figure 3) for the probability of transitions between the different groups in Table 1. The entries in row i and column j in the matrix denote the probability that there are j possible output bit pairs from the function

g, given that there were i possible input bit pairs. For example, for a set of three input bit pairs (3rd row) the output will be either three bit pairs (3rd column) with 50% chance or two bit pairs (2nd column) with 50% chance.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/6 & 5/6 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 3: Mitchell's transition probability matrix (Mitchell 2007)

If we iterate the above matrix v times, the bottom left entry of the resulting power matrix give us the probability of obtaining a single pair of bits as the possible output of g after v iterations (Mitchell 2007). Note that v also indicates the number of staggered plaintext/ciphertext pairs required to iterate this analysis v times.

Using this approach, Mitchell calculated the probability of knowing a single pair of bits and the corresponding probability of revealing a whole 128-bit block for various values of v. We present this information in Table 2; here p denotes the probability of a unique possibility for a single bit pair and q denotes the probability of a unique possibility for a 128-bit block. For instance, with knowledge of 30 staggered pairs of plaintext/ciphertext blocks, there is a 99% chance that a bit pair will be known.

Mitchell used this theory to launch an attack. We discuss the methodology in section 4.

v	10	20	30	40	50	60
p	0.71027	0.95305	0.99241	0.99878	0.99980	0.99997
q	Very small	0.04607	0.61409	0.92485	0.98728	0.99808

Table 2: Probability of a unique possibility for a bit pair and a 128-bit block (Mitchell 2007)

3.2 The flaw in Mitchell's analysis

Determining the inner vectors (F_i and G_i) is critical to the success of Mitchell's forgery attack. We reviewed his process for obtaining the inner vectors and found that this process cannot uniquely determine the inner vectors of EPBC. Two alternatives remain for every pair of inner vectors. For example, this gives 2^{64} alternatives for a 128-bit block cipher.

The thick line in Table 1 divides the input/output possibilities for g into two separate groups. We consider specifically the cases where there are two possible input pairs, and divide this into groups: $2a$ and $2b$. Note that it is not possible to obtain an input set in group $2a$ as the output from applying g to any of the sets in group $2b$. Recall that $G_{i+1} = P_{i+1} \oplus C_i \oplus g(G_{i-1})$; thus, for any given bit position (x_j, x_{j+m}) in G_{i-1}, the relevant pair in $P_{i+1} \oplus C_i$ must be XORed with each of the possible output pairs in $g(G_{i-1})$. Now every set in group $2b$ consists of one pair in which the two bits are identical and one pair in which the two bits are different. Regardless of which set we choose from group $2b$ and which bit pair is XORed to both of these, the resulting set of bit pairs will have the same property and must therefore belong to group $2b$ as well. Therefore the possibilities which are above the splitting line are as far as we can achieve. Thus, in this attack, for every pair of bits, instead of four alternatives we can have two (from group $2b$). The remaining groups in Table 1 are not accessible.

From Table 1, we develop a new theoretical transition probability matrix, as shown in Figure 4. The entry in the i row and the j column of this matrix denotes the probability that the number of input pairs i will generate the number of output pairs j. The labels $2a$ and $2b$ in the matrix relate to the two rows below the thick line and the four rows above the thick line in Table 1.

$$\begin{array}{c} \\ 1 \\ 2a \\ 2b \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2a & 2b & 3 & 4 \end{array} \\ \left(\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right) \end{array}$$

Figure 4: Theoretical transition probability matrix

Although it is not possible to uniquely determine the inner vectors F_i and G_i, we can reduce the number of possible values for each inner vector. For example, for a 128-bit block cipher, the number of possible values can be reduced from 4^{64} alternatives to 2^{64} alternatives. The chance of guessing the whole final inner vector correctly is now 2^{-64}. Although this is low it is dramatically better than the probability of guessing the entire block (2^{-128}).

By iterating the corrected matrix (Figure 4) v times, the $(4, 2b)$ entry of the resulting power matrix gives the probability p' of obtaining two alternatives as the possible output of g after v iterations. This probability can then be used to determine the corresponding probability q' that an n-bit block has only $2^{n/2}$ alternatives. Table 3 lists the values of p' and q' for chosen values of v and n =128. For example, after 10 iterations, the probability p' that there are two alternatives remaining for a bit pair is 99%; for a complete block of 128 bits, the probability q' that there are only two alternatives for each pair is 88%.

v	2	3	4	5	10	15
p'	0.5	0.75	0.875	0.9375	0.99805	0.99994
q'	Very small	0.00000001	0.000194	0.016075	0.882389	0.996101

Table 3: Probability of two alternatives for a bit pair and for every pair in a 128-bit block

4 Mitchell's Forgery Attack

Based on his analysis, Mitchell (2007) explains how a forged ciphertext message can be derived by controlled deletion of blocks in a legitimate ciphertext message. Assume the attacker has obtained the values for two of the inner vectors G_i. Blocks can be deleted anywhere between the first ciphertext block and the second last

ciphertext block. The ciphertext block after the deleted blocks must be modified to permit recovery of the correct inner vectors for the decryption process of the following ciphertext block. This ensures that the decrypted ICV value remains unchanged (Zuquete and Guedes 1997). For details of the construction, refer to Mitchell's paper.

4.1 Attack application

We demonstrate this attack for a seven block ciphertext $C_1, C_2, ..., C_7$. Assume that the inner vectors G_3 and G_5 are known and that the final plaintext block, P_7, is the ICV. Following Mitchell's process, we constructed a forged ciphertext by deleting ciphertext blocks C_4 and C_5, modifying C_6 and leaving C_7 unchanged. After decryption, the forged ciphertext $C_1, C_2, C_3, C_4^*, C_5^*$ generates the correct value for the ICV. This forgery attack has been demonstrated for a specific example by coding it in *C* programming language, using AES with a block length of 128 bits.

4.2 Success rate of revised attack

Mitchell claimed that with the knowledge of over 100 consecutive plaintext/ciphertext blocks (v>50), the inner vectors G_i would be revealed with very high probability (Mitchell 2007). However, as we showed in Sect. 3.2, the number of possible values for each inner vector G_i can only be reduced to 2^{64} alternatives. The deletion attack described above requires two G_i values to be known. Therefore the probability of a successful forgery following this method is 2^{-128}. This contradicts Mitchell's claim that the forgery is guaranteed to succeed. Therefore Mitchell's attack is no better than making random changes to the ciphertext (insertion, deletion or substitution) and hoping that the final block decrypts to give the correct ICV.

If the length of the ICV is l bits, then the probability of successful brute force attack on the ICV is 2^{-l}. If l <128 bits then this approach has higher success probability than Mitchell's forgery attack.

4.3 Comparison with key recovery attacks

Recall that EPBC uses two keys, K' and K. Suppose we use a cipher with a block length and a key size both of 128 bits. Key K' is used to firstly encrypt a sequence number S to obtain F_0, and then encrypt F_0 to obtain G_0. Key K is used to encrypt G_i to obtain F_i for each message block. We compare Mitchell's attack against exhaustive search on either or both keys.

Suppose *S* is known to the public and that a number of pairs of plaintext/ciphertext blocks are known to the attacker. Then it can be shown that exhaustive search on both keys requires 2^{256} guesses. Checking each of these guesses will require at least one decryption, so the complexity will be around 2^{256}. Knowing both K' and K allows the attacker to decrypt all ciphertext messages and impersonate either sender or receiver to communicate with the other one.

Now consider the key K'. If this key and at least the first two plaintext/ciphertext pairs are known to the attacker, the relevant inner vectors can be revealed and a forgery attack conducted following Mitchell's process. The probability of guessing this key correctly is 2^{-128}. The correctness of the guess is verified by the receiver accepting the forged message.

Finally, consider the key K. If the attacker knows three consecutive plaintext/ciphertext pairs, it can be shown that this key and the inner vectors for these blocks can be revealed with a complexity of roughly 2^{231} encryption/ decryption operations. The knowledge of key K and these inner vectors guarantees the success of a forgery attack.

5 Conclusion

We reviewed Mitchell's forgery attack on EPBC and found a flaw in his estimation of the probabilities of correctly obtaining the inner vectors. Knowledge of these inner vectors allows a forgery to be constructed. We show that, regardless of the number of known plaintext/ ciphertext blocks, the possible values for each inner vector can only be reduced to two alternatives per bit pair, rather than being uniquely determined as claimed by Mitchell. When the block length of the underlying cipher is 128 bits, the number of alternatives is reduced from 2^{128} to 2^{64}. The success rate of Mitchell's forgery attack is therefore 2^{-128}. This is no better than a brute force attack on the ICV, and worse if the length of the ICV is less than 128 bits. If the block cipher has a 128-bit key this is also comparable to exhaustive search on the key K'. For all of these attacks, the attacker does not know whether the modified ciphertext will be accepted before sending it.

Alternatively, the attacker can construct a forged ciphertext that is guaranteed to be accepted if either the second key K or both keys are known. However the calculation complexity of finding these keys is prohibitive (2^{231} for finding K and 2^{256} for finding both keys).

Our results indicate that EPBC is in fact secure against Mitchell's forgery attack. Additionally, we recommend that the ICV should be no shorter than the block length, to reduce the success rate of brute force attacks on the ICV.

6 References

Mitchell, C. (2007): Cryptanalysis of the EPBC Authenticated Encryption Mode. Cryptography and Coding, Springer Berlin Heidelberg, pp. 118-128.

Preneel, B. (1998): Cryptanalysis of message authentication codes. Information Security, Springer Berlin Heidelberg, pp. 55-65

Recacha, F. (1996): IOBC: Un nuevo modo de encadenamiento para cifrado en bloque. Proceedings: IV Reunion Espanola de Criptologia, Valladolid, pp. 85–92.

Zuquete, A and Guedes, P. (1997): Efficient error-propagating block chaining. Cryptography and coding, Springer Heidelberg, pp. 323-334.