

# Proposal To Extend CBC Mode By “Ciphertext Stealing”

May 6, 2007

A limitation to Cipher Block Chaining (CBC) mode, as specified in [1], is that the plaintext input must consist of a sequence of complete blocks. Although Appendix A of [1] describes how padding methods can be used to meet this requirement, in such cases the length of the resulting ciphertext expands over the length of the unpadded plaintext by the number of padding bits.

This note proposes an extension of CBC mode that accepts any plaintext input whose bit length is greater than or equal to the block size but not necessarily a multiple of the block size. Unlike the padding methods discussed in [1], the extended mode does not expand the size of the ciphertext. The extended mode is called Cipher Block Chaining-Ciphertext Stealing (CBC-CS), because when padding bits are necessary, they are taken from the penultimate ciphertext block.

Below are the specifications and diagrams for CBC-CS encryption and decryption, building on the specification of CBC mode encryption and decryption in [1]. CBC-CS inherits the relevant requirements of [1], e.g., on the underlying block cipher, the key, and the initialization vector.

The following notational conventions apply to the specifications below:

- Bit strings are denoted with capital letters; integers with lower case letters.
- The block size of the underlying block cipher is denoted  $b$ .
- For a bit string  $X$ , the bit length of  $X$  is denoted  $\text{len}(X)$ ;
- For a bit string  $X$  and a positive integer  $r$  that does not exceed  $\text{len}(X)$ , the string consisting of the leftmost  $r$  bits of  $X$  is denoted  $\text{MSB}_r(X)$ , and the string consisting of the rightmost  $r$  bits of  $X$  is denoted  $\text{LSB}_r(X)$ .
- For an input block  $B$ , the output block of the cipher function (“encryption”) is denoted  $\text{CIPH}(B)$ , and the output block of the inverse cipher function (“decryption”) is denoted  $\text{CIPH}^{-1}(B)$ .

## Algorithm: CBC-CS-Encrypt

*Input:* plaintext  $P$ , such that  $\text{len}(P) \geq b$ ; initialization vector  $IV$ .

*Output:* ciphertext  $C$ .

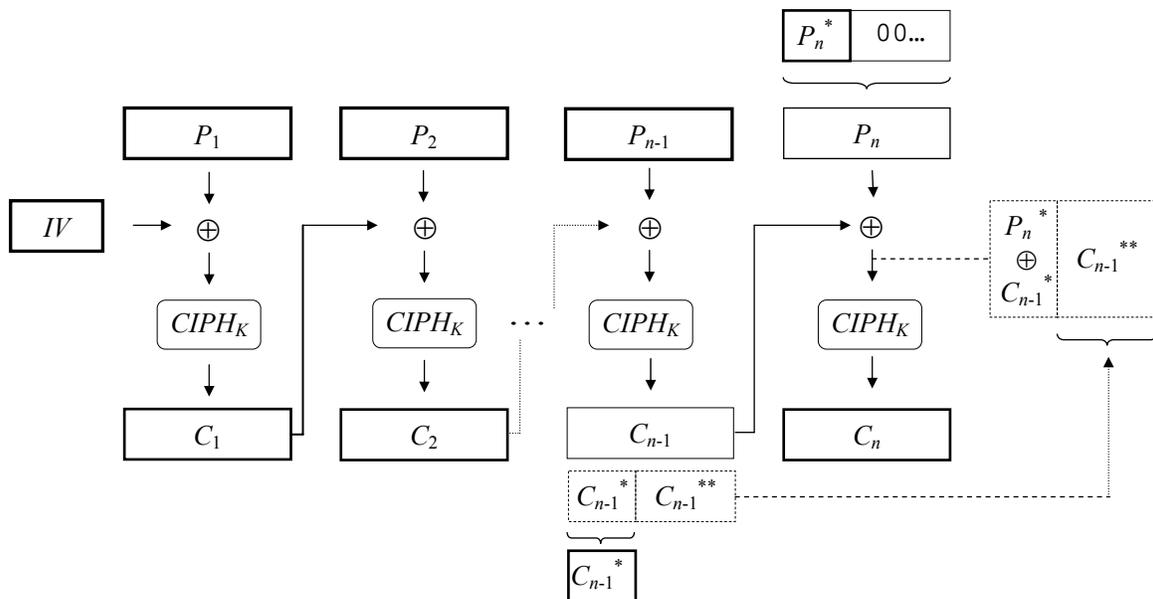
*Steps:*

1. Let  $n$  be the smallest integer such that  $n \cdot b \geq \text{len}(P)$ , let  $d = \text{len}(P) - (n-1) \cdot b$ , and let  $P_1, P_2, \dots, P_{n-1}, P_n^*$  be the unique sequence of bit strings such that:
  - a)  $P = P_1 \parallel P_2 \parallel \dots \parallel P_{n-1} \parallel P_n^*$ ; and

- b)  $P_1, P_2, \dots$  and  $P_{n-1}$  are complete blocks.  
 Consequently,  $d = \text{len}(P_n^*)$  and  $1 \leq d \leq b$ , so that  $P_n^*$  is either a complete block or a nonempty partial block.
2. If  $d = b$ , then
    - a) apply CBC mode encryption with initialization vector  $IV$  to the plaintext  $(P_1, P_2, \dots, P_{n-1}, P_n^*)$  to produce  $(C_1, C_2, \dots, C_{n-1}, C_n)$ ;
    - b) return  $C_1 \parallel C_2 \parallel \dots \parallel C_{n-1} \parallel C_n$ .
 If  $d < b$ , go to Step 3.
  3. Let  $PAD$  be the bit string consisting of  $b-d$  '0' bits, and let  $P_n = P_n^* \parallel PAD$ .
  4. Apply CBC mode to the plaintext  $(P_1, P_2, \dots, P_{n-1}, P_n)$  with initialization vector  $IV$  to produce  $(C_1, C_2, \dots, C_{n-1}, C_n)$ .
  5. Let  $C_{n-1}^* = \text{MSB}_d(C_{n-1})$ .
  6. Return  $C_1 \parallel C_2 \parallel \dots \parallel C_{n-1}^* \parallel C_n$ .

*Diagram:*

The following diagram illustrates CBC-CS encryption for the case that  $d < b$ :



Note that, in effect, the padding of the partial plaintext block  $P_n^*$  with '0' bits causes the rightmost  $b-d$  rightmost bits of  $C_{n-1}$  to be "stolen" as padding for the input to the final invocation of the block cipher. This string of bits, denoted  $C_{n-1}^{**}$ , is omitted from the ciphertext because it can be recovered from  $C_n$  during decryption.

Algorithm: CBC-CS-Decrypt

*Input:* ciphertext  $C$ , such that  $\text{len}(C) \geq b$ ; initialization vector  $IV$ .

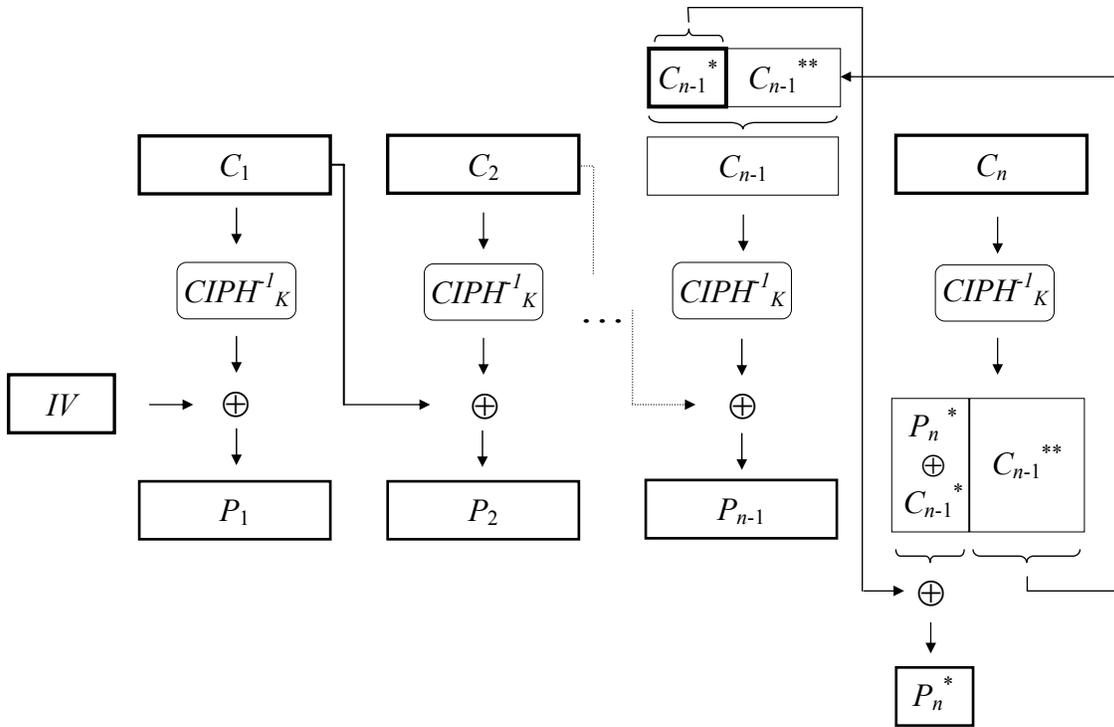
*Output:* plaintext  $P$ .

*Steps:*

1. Let  $n$  be the smallest integer such that  $n \cdot b \geq \text{len}(C)$ , let  $d = \text{len}(C) - (n-1)b$ , and let  $C_1, C_2, \dots, C_{n-2}, C_{n-1}^*, C_n$  be the unique sequence of bit strings such that
  - a)  $C = C_1 \parallel C_2 \parallel \dots \parallel C_{n-2} \parallel C_{n-1}^* \parallel C_n$ ;
  - b)  $C_n$  is a complete block; and
  - c) If  $n > 2$ , then  $C_1, \dots, C_{n-2}$ , are complete blocks.Consequently,  $d = \text{len}(C_{n-1}^*)$  and  $1 \leq d \leq b$ , so that  $C_{n-1}^*$  is either a complete block or a nonempty partial block.
2. If  $d = b$ , then
  - a) apply CBC mode decryption to  $C_1, C_2, \dots, C_{n-2}, C_{n-1}^*, C_n$  with initialization vector  $IV$  to produce  $P_1, P_2, \dots, P_{n-1}, P_n$ ;
  - b) return  $P_1 \parallel P_2 \parallel \dots \parallel P_{n-1} \parallel P_n$ .If  $d < b$ , then go to Step 3.
3. Let  $C_{n-1} = C_{n-1}^* \parallel \text{LSB}_{b-d}(\text{CIPH}^{-1}(C_n))$ .
4. Apply CBC mode decryption to  $(C_1, C_2, \dots, C_{n-1}, C_n)$  with initialization vector  $IV$  to produce  $(P_1, P_2, \dots, P_{n-1}, P_n)$ .
5. Let  $P_n^* = \text{MSB}_d(P_n)$ .
6. Return  $P_1 \parallel P_2 \parallel \dots \parallel P_{n-1} \parallel P_n^*$ .

*Diagram:*

The following diagram illustrates CBC-CS decryption for the case that  $d < b$ . As in the previous diagram, the “stolen” ciphertext is denoted  $C_{n-1}^{**}$ .



- [1] NIST Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, December 2001, Natl. Inst. Stand. Technol. [Web page], <http://www.csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf>.