Addendum to
NIST Special Publication 800-38A
October, 2010

National Institute of
Standards and Technology

U.S. Department of Commerce

Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode

Morris Dworkin

C O M P U T E R S E C U R I T Y

Addendum to
NIST Special Publication 800-38A

Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode

Morris Dworkin

C O M P U T E R   S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

October 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Patrick Gallagher, Director

Reports on Information Security Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

Addendum
National Institute of Standards and Technology Special Publication 800-38A
11 pages (June 2010)
CODEN: NSPUE2

Table of Contents

List of Figures

1 Introduction

A limitation to Cipher Block Chaining (CBC) mode, as specified in NIST SP 800-38A, Ref. [1], is that the plaintext input must consist of a sequence of blocks. (In the rest of this publication, a block is called a "complete block" to emphasize the contrast with a "partial block" whose bit length is smaller than the block size.) Although Appendix A of Ref. [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 addendum to Ref. [1] specifies three variants of CBC mode that accept any plaintext input whose bit length is greater than or equal to the block size, whether or not the length is a multiple of the block size. Unlike the padding methods discussed in Ref. [1], these variants avoid ciphertext expansion.

These variants are denoted CBC-CS1, CBC-CS2, and CBC-CS3, where "CS" indicates "ciphertext stealing," because when padding bits are needed in these variants, they are taken from the penultimate ciphertext block. The variants differ only in the ordering of the ciphertext bits. CBC-CS1 was specified as a suggestion on the NIST Computer Security Resource Center web site. CBC-CS2 is specified, for example, in Ref. [3]. CBC-CS3 is the variant specified for Kerberos 5 in Ref. [2].

Below are the specifications for encryption and decryption using CBC-CS1, CBC-CS2, and CBC-CS3, building on the specification of the CBC encryption and decryption in Ref. [1]. Diagrams are given for CBC-CS1 encryption and decryption. Each variant inherits the relevant requirements of Ref. [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 upper case 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 len(X).
- For a bit string X and a positive integer r that does not exceed len(X), the string consisting of the leftmost r bits of X is denoted $\mathrm{MSB}_r(X)$, and the string consisting of the rightmost r bits of X is denoted $\mathrm{LSB}_r(X)$.
- For an input block B and key K, the output block of the cipher function ("encryption") is denoted $\mathrm{CIPH}_K(B)$, and the output block of the inverse cipher function ("decryption") is denoted $\mathrm{CIPH}^{-1}_K(B)$.

In principle, the input strings to encryption and decryption for each of the variants may be any string whose length is not less than b bits, but typically an implementation is designed to accept a restricted set of lengths. For example, the set of lengths may be restricted to multiples of 8, so that the input strings may be represented with bytes/octets. The input lengths that an implementation allows are called the *valid* lengths.

2 Specification of CBC-CS1

Algorithm: CBC-CS1-Encrypt

Input: plaintext P, such that len(P) is valid;
initialization vector IV;
key K.

Output: ciphertext C, such that len(C)=len(P).

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, … , P_{n-1}, P_n^* be the unique sequence of bit strings such that:
   a) $P = P_1||P_2||\ldots||P_{n-1}||P_n^*$; and
   b) P_1, P_2, … and P_{n-1} are complete blocks.
   Consequently, len(P_n^*)=d, and $1 \leq d \leq b$, so that P_n^* is either a complete block or a nonempty partial block.

2. Let PAD be the bit string consisting of b-d '0' bits, and let $P_n= P_n^*||PAD$. Note that if P_n^* is a complete block, then PAD is the empty string, and $P_n= P_n^*$.

3. Apply CBC encryption to the plaintext (P_1, P_2, … P_{n-1}, P_n) with initialization vector IV and key K to produce (C_1, C_2, … C_{n-1}, C_n).

4. Let $C_{n-1}^* = \text{MSB}_d(C_{n-1})$.

5. Return $C_1||C_2||\ldots||C_{n-2}||C_{n-1}^*||C_n$.

When P_n^* is a complete block, CBC-CS1-Encrypt is equivalent to CBC encryption.

Diagram: Figure 1 below illustrates the CBC-CS1-Encrypt algorithm for the case that P_n^* is a partial block, i.e., $d<b$. The bolded rectangles contain the inputs and outputs. The dotted rectangles provide alternate representations of two blocks in order to illustrate the role of the "stolen" ciphertext.

In particular, the string of the b-d rightmost bits of C_{n-1}, denoted C_{n-1}^{**}, becomes the padding for the input block to the final invocation of the block cipher within the execution of CBC mode. The ciphertext that is returned in Step 5 above omits C_{n-1}^{**}, because it can be recovered from C_n during decryption.

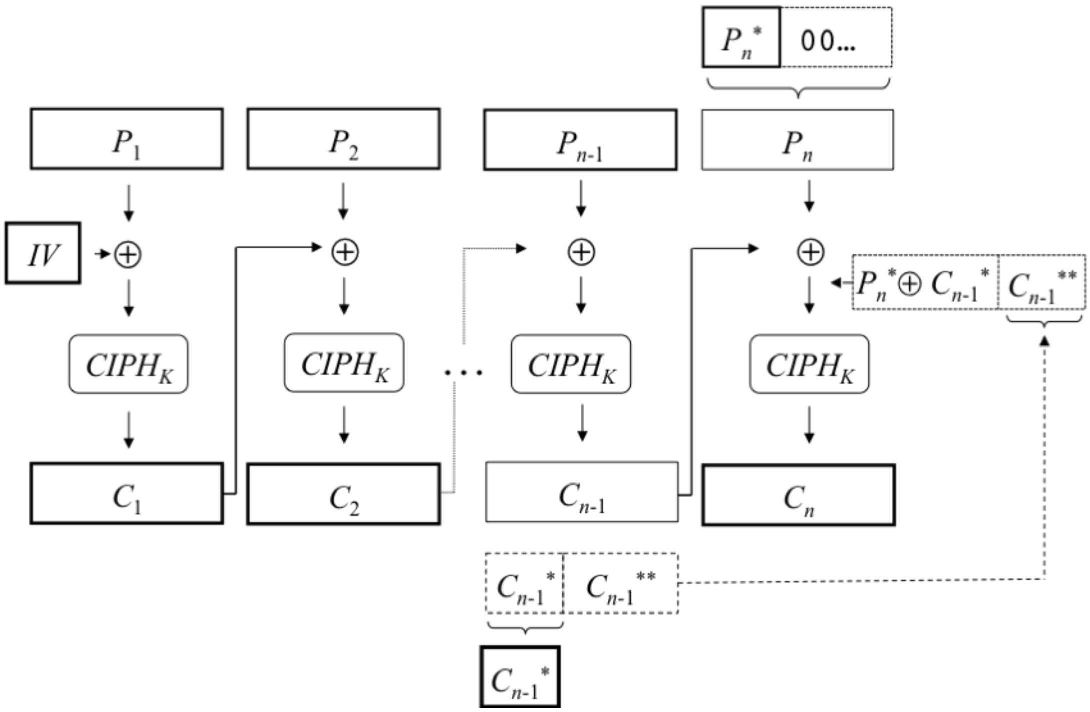

Figure 1: CBC-CS1-Encrypt

Algorithm: CBC-CS1-Decrypt

Input: ciphertext C, such that len(C) is valid;
initialization vector IV;
key K.

Output: plaintext P, such that len(P) = len(C).

Steps:

1. Let n be the smallest integer such that $n \cdot b \geq \text{len}(C)$, let $d = \text{len}(C) - (n-1) \cdot b$, and let C_1, C_2, … , C_{n-2}, C_{n-1}^*, C_n be the unique sequence of bit strings such that
   a) $C = C_1 \| C_2 \| \ldots \| C_{n-2} \| C_{n-1}^* \| C_n$;
   b) C_n is a complete block; and
   c) if $n>2$, then $C_1, \ldots, C_{n-2}$, are complete blocks.

   Consequently, $\text{len}(C_{n-1}^*) = d$, and $1 \leq d \leq b$, so that C_{n-1}^* is either a complete block or a nonempty partial block.

2. Let $Z^* = \text{MSB}_d(\text{CIPH}^{-1}_K(C_n))$, and $Z^{**} = \text{LSB}_{b-d}(\text{CIPH}^{-1}_K(C_n))$.

3. Let $C_{n-1} = C_{n-1}^* \| Z^{**}$. Note that if C_{n-1}^* is a complete block, then Z^{**} is the empty string, and $C_{n-1} = C_{n-1}^*$.
4. Apply CBC decryption to (C_1, C_2, …, C_{n-1}) with initialization vector IV and key K to produce (P_1, P_2, … P_{n-1}).

5. Let $P_n^* = C_{n-1}^* \oplus Z^*$.

6. Return $P_1||P_2||\ldots||P_{n-1}||P_n^*$.

When C_{n-1}^* is a complete block, CBC-CS1-Decrypt is equivalent to CBC decryption.

Diagram:

Figure 2 below illustrates the CBC-CS1-Decrypt algorithm for the case that C_{n-1}^* is a partial block, i.e., $d<b$. As in the previous diagram, the bolded rectangles contain the inputs and outputs.

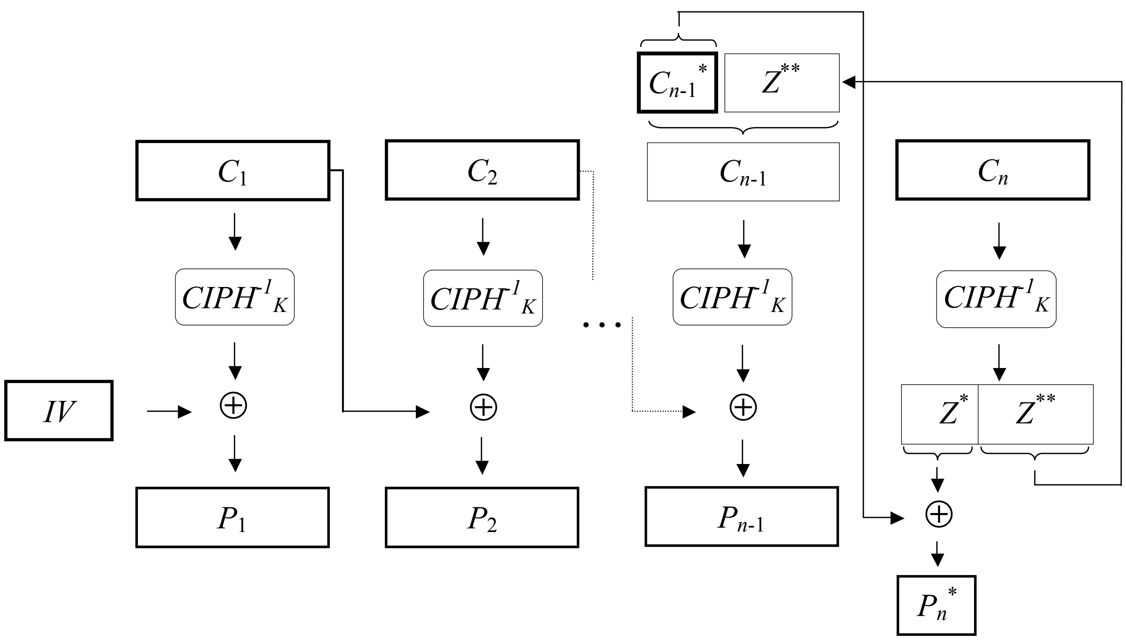

Figure 2: CBC-CS1-Decrypt

3 Specification of CBC-CS2

Algorithm: CBC-CS2-Encrypt

Input: plaintext P, such that len(P) is valid;
initialization vector IV;
key K.

Output: ciphertext C, such that len(C) = len(P).

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, … , P_{n-1}, P_n^* be the unique sequence of bit strings such that:
   a) $P=P_1||P_2||\ldots||P_{n-1}||P_n^*$; and
   b) P_1, P_2, … and P_{n-1} are complete blocks.

   Consequently, $\text{len}(P_n^*)=d$, and $1 \leq d \leq b$, so that P_n^* is either a complete block or a nonempty partial block.

2. Apply CBC-CS1-Encrypt to the plaintext P with initialization vector IV and key K to produce $C_1||C_2||\ldots||C_{n-2}||C_{n-1}^*||C_n$.

3. If $d=b$, return $C_1||C_2||\ldots||C_{n-2}\,||C_{n-1}^*||C_n$; if $d<b$, return $C_1||C_2||\ldots||C_{n-2}||C_n||C_{n-1}^*$.

When P_n^* is a complete block, then CBC-CS2-Encrypt, like CBC-CS1-Encrypt, is equivalent to CBC encryption. When P_n^* is a partial block, then CBC-CS2-Encrypt and CBC-CS1-Encrypt differ only in the ordering of C_{n-1}^* and C_n.

Algorithm: CBC-CS2-Decrypt

Input: ciphertext C, such that len(C) is valid;
initialization vector IV;
key K.

Output: plaintext P, such that $\text{len}(P)=\text{len}(C)$.

Steps:

1. Let n be the smallest integer such that $n \cdot b \geq \text{len}(C)$, let $d=\text{len}(C)-(n-1)\cdot b$, and let C_1, C_2, … , C_{n-2}, C_{n-1}, C_n^* be the unique sequence of bit strings such that
   a) $C=C_1||C_2||\ldots||C_{n-2}\,||C_{n-1}||C_n^*$;
   b) C_{n-1} is a complete block; and
   c) If $n>2$, then $C_1,\ldots,C_{n-2}$, are complete blocks.

   Consequently, $\text{len}(C_n^*)=d$, and $1 \leq d \leq b$, so that C_n^* is either a complete block or a nonempty partial block.

2. If $d=b$, i.e., C_n^* is a complete block, then apply CBC-CS1-Decrypt to C with initialization vector IV and key K, and return the result, P. STOP.

3. If $d<b$, i.e., C_n^* is a partial block, then let $C' = C_1||C_2||\ldots||C_{n-2}\,||C_n^*||C_{n-1}$.

4. Apply CBC-CS1-Decrypt to C', and return the result, P.

Note that in Step 3, C_n^* precedes C_{n-1} in order to undo the corresponding swap within the ciphertext in Step 3 of CBC-CS2-Encrypt.

4 Specification of CBC-CS3

Algorithm: CBC-CS3-Encrypt

Input: plaintext P, such that len(P) is valid;
initialization vector IV;
key K.

Output: ciphertext C, such that len(C)=len(P).

Steps:

1. Apply CBC-CS1-Encrypt to the plaintext P with initialization vector IV and key K to produce $C_1||C_2||\ldots||C_{n-2}||C_{n-1}^*||C_n$.

2. Return $C_1||C_2||\ldots||C_{n-2}||C_n||C_{n-1}^*$.

Note that in Step 2, C_{n-1}^* and C_n are unconditionally swapped, i.e., even when C_{n-1}^* is a complete block; therefore, CBC-CS3 is not strictly an extension of CBC mode. In the other case, i.e., when C_{n-1}^* is a nonempty partial block, CBC-CS3-Encrypt is equivalent to CBC-CS2-Encrypt.

Algorithm: CBC-CS3-Decrypt

Input: ciphertext C, such that len(C) is valid;
initialization vector IV;
key K.

Output: plaintext P, such that len(P)=len(C).

Steps:

1. Let n be the smallest integer such that $n \cdot b \geq$len(C), let d=len(C)-(n-1)$\cdot b$, and let C_1, C_2, … , C_{n-2}, C_{n-1}, C_n^* be the unique sequence of bit strings such that
   a) $C=C_1||C_2||\ldots||C_{n-2}\,||C_{n-1}||C_n^*$;
   b) C_{n-1} is a complete block; and
   c) If n>2, then $C_{1,}\ldots,C_{n-2}$, are complete blocks.

   Consequently, len(C_n^*)=d, and $1 \leq d \leq b$, so that C_n^* is either a complete block or a nonempty partial block.

2. Apply the CBC-CS1-Decrypt algorithm with initialization vector IV and key K to $C_1||C_2||\ldots\,||C_{n-2}\,||C_n^*||C_{n-1}$, and return the result, P.

Note that in Step 2, C_n^* precedes C_{n-1} in order to undo the corresponding swap that was performed within Step 2 of CBC-CS3-Encrypt.

References

[1] Dworkin, M., *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.

[2] Raeburn, K., *Request for Comments 3962*: *Advanced Encryption Standard (AES) Encryption for Kerberos 5*, Internet Engineering Task Force, February 2005.

[3] Schneier, B., *Applied Cryptography, Second Edition: protocols, algorithms, and source code in* C. New York: Wiley, 1996.