A Note on NSA's Dual Counter Mode of Encryption

Pompiliu Donescu * pompiliu@eng.umd.edu

Virgil D. Gligor ** gligor@eng.umd.edu

David Wagner *** daw@cs.berkeley.edu

September 28, 2001

Abstract. We show that both variants of the Dual Counter Mode of encryption (DCM) submitted for consideration as an AES mode of operation to NIST by M. Boyle and C. Salter of the NSA are insecure with respect to both secrecy and integrity in the face of chosen-plaintext attacks. We argue that DCM cannot be easily changed to satisfy its stated performance goal and be secure. Hence repairing DCM does not appear worthwhile.

1 Introduction

On August 1, 2001, M. Boyle and C. Salter of the NSA submitted two variants of the Dual Counter Mode (DCM) of encryption [1] for consideration as an AES mode of operation to NIST. The DCM goals are: (1) to protect both the secrecy and integrity of IP packets (as this mode is intended to satisfy the security goals of Jutla's IAPM mode [4]), and (2) to avoid the delay required before commencing the decryption of out-of-order IP packets, thereby decreasing the decryption latency of IAPM. DCM is also intended to allow high rates of encryption.

The authors argue that DCM satisfies the first goal because "an error in a cipher block causes all data in the packet to fail the integrity check". DCM appears to satisfy the second goal because it maintains a "shared secret negotiated during the key exchange," which avoids the delay inherent to the decryption of a secret IV before the first out-of-order packet arrival can be decrypted. The authors note correctly that Jutla's IAPM mode does not satisfy their second goal.

In this note, we show that both variants of DCM are insecure with respect to both secrecy and integrity in the face of chosen-plaintext attacks. Further, we argue that DCM cannot be easily changed to satisfy its stated performance goal for the decryption of out-of-order packets *and* be secure. We conclude since other proposed AES modes satisfy the proposed goals for DCM, even if repairing DCM is possible, which we doubt, such an exercise does not appear to be worthwhile.

[1] VDG Inc., 6009 Brookside Drive, Chevy Chase, MD 20815.

[2] Electrical and Computer Engineering Department, University of Maryland, College Park, Maryland 20742.

[3] Computer Science Division, EECS Department, University of California Berkeley, Berkeley, CA. 94720.

2 Secrecy and Integrity Attacks against DCM

2.1 DCM Security Goals

The security of a mode is commonly expressed as a combination of secrecy and integrity goals to be achieved in the face of different types of attacks. A common type of attacks used to break encryption modes intended to satisfy these goals is the *chosen-plaintext attack*. In a chosen-plaintext attack, an adversary can obtain samples of valid encryptions for plaintext messages of his choice even though the secret encryption key remains unknown to him[1].

A typical *secrecy* goal is one that requires that an adversary be unable to distinguish the encryption of a plaintext that he chooses from that of a random string of the same length without the benefit of knowing the random, secret, encryption key. This goal, which is usually called indistinguishibility in a real-or-random sense or simply real-or-random security, provides a powerful notion of secrecy when combined with a chosen-plaintext attack; i.e., it covers many desirable secrecy properties such as assuring an adversary's inability to decrypt a ciphertext message whose plaintext is unknown with non-negligible probability [2]. A typical *integrity* goal is one that requires that an adversary be unable to produce a never-seen-before ciphertext message that decrypts correctly (i.e., a valid forgery) with non-negligible probability. This goal, which is usualy called existential-forgery protection or existential unforgeability, yields the most powerful notion of integrity known to date when combined with a chosen-plaintext attack (viz., discussion of integrity notions at *http://csrc.nist.gov/encryption/modes/proposedmodes*).

Undoubtedly, DCM aims to support the security notions defined above in the face of chosen-plaintext attacks, as its *stated* design goal is to improve upon the performance characteristics of Jutla's IAPM—a mode that satisfies these security notions—not to weaken its security.

2.2 DCM Definition – First Variant

We first illustrate our attacks againts the first variant of DCM. Given a polynomial of degree W for a primitive LFSR, where W is the width of the block cipher E, the authenticated DCM encryption mode encrypts the plaintext P into the ciphertext C as follows [1]:

ENCRYPT($P_1, \ldots, P_j$):
1. Set checksum = 0.
2. For $i = 1, \ldots, j$, do:
3. Set $x_i = f(x_{i-1})$.
4. Set $C_i = E(P_i \oplus x_i) \oplus x_i$.
5. Set checksum = checksum $\oplus P_i$.

[1] Chosen-plaintext attacks are quite practical [5]. In fact, they are some of the oldest known attacks in modern cryptography; viz., the "gardening" attacks of British cryptographers during WWII. [3]

6. Set $x_{j+1} = f(x_j)$.
7. Set $C_{j+1} = E(\text{checksum} \oplus x_{j+1}) \oplus x_0$.

In this definition, the plaintext contains j blocks, $P = \langle P_1, \ldots, P_j \rangle$, and its DCM encryption produces a ciphertext string that contains $j+1$ blocks, $C = \langle C_1, \ldots, C_j, C_{j+1} \rangle$. It is important to note that "x_0 is the shared secret negotiated during the key exchange" between a sender and a receiver and thus remains constant during the use of that key.

Integrity Attacks. To illustrate the integrity flaws of DCM we provide two examples of an chosen-plaintext existential forgery attack. The first example takes advantage of the properties of the LFSR connection polynomial f, whereas the second is independent of f. In fact, the sequence x_i could be generated by a cryptographic means from x_0, re-using the same x_0 for every message encrypted with a key, and DCM would still remain vulnerable to integrity attacks.

Attack 1. In this attack the adversary first obtains x_0 and then constructs a valid forgery based on knowledge of x_0. To obtain x_0 the adversary first obtains $x_0 \oplus x_j$ in a chosen-plaintext attack. For example, for $j = 2$, an adversary can launch a chosen-plaintext attack to obtain chosen plaintext and ciphertext pairs (P, C) and (P', C') encrypted in DCM, where $P = \langle 0 \rangle$ and $P' = \langle 0, 0 \rangle$, where here 0 denotes the all-zeros block. Then the relation $C_2 \oplus C_2' = x_0 \oplus x_2$ reveals $x_0 \oplus x_2$, and moreover we claim that this information can be used to recover x_0.

Claim. For a known (non-singular) LFSR of degree W with known feedback taps, knowledge of $x_0 \oplus x_j$ reveals x_0.

Proof. Let $p_f(t)$ be the connection polynomial of the LFSR (of degree W). View the initial fill x_0 as a polynomial from $GF(2)[t]$, taken modulo $p_f(t)$. Then $x_1 = t \times x_0 \bmod p_f(t)$, and $x_j = t^j \times x_0 \bmod p_f(t)$, where t^j denotes the monomial of degree j. Thus, $x_0 + x_j = (1 + t^j) x_0 \bmod p_f(t)$. Since $1 + t^j$ has an easily-computable inverse modulo $p_f(t)$ with very high probability, we can compute $x_0 = (x_0 + x_j)/(1 + t^j) \bmod p_f(t)$.

Given x_0, an adversary can easily break the integrity of DCM. Here is a simple example that shows how to constructing a valid forgery with probability 1.

Let $P = \langle P_1, P_2 \rangle$ be a chosen plaintext such that $P_1 = P_2$, and obtain the corresponding ciphertext C using the chosen-plaintext assumption. Note that $C_2 = E(P_2 \oplus x_2) \oplus x_2$ and $C_3 = E(x_3) \oplus x_0$, as the checksum is zero. Since the adversary knows x_0, he also knows x_j for all j; for example, he knows $x_0 \oplus x_2$. Hence, he can construct a new ciphertext $C' = \langle C_1, C_2' \rangle \neq C$, and this forgery will be accepted by the receiver if we take

$$C_2' = C_2 \oplus x_0 \oplus x_2.$$

This forgery is valid and passes the integrity check with probability 1: when we decrypt C', the plaintext consists of only P_1, and the decryption of the checksum block yields

$$P_2' = E^{-1}(C_2' \oplus x_0) \oplus x_2 = E^{-1}(C_2 \oplus x_0 \oplus x_2 \oplus x_0) \oplus x_2$$
$$= E^{-1}(C_2 \oplus x_2) \oplus x_2 = P_2,$$

Since our choice of P ensures that $P_2 = P_1$, the checksum check will pass with probability 1.

This demonstrates that DCM's integrity properties are broken. Note also that the attack remains valid (changing t^j to t^{jW} in the proof of Claim 2.2) even if we shift the register W times between each block.

Attack 2. Next we break the integrity of DCM with a simple truncation executed by splicing ciphertext blocks of two chosen plaintext messages P and Q. Let the first chosen plaintext be $P = \langle P_1, P_2, \ldots, P_{n-1}, P_n \rangle$ such that

$$P_1 \oplus P_2 \oplus \ldots \oplus P_{n-1} = 0,$$

and let the second chosen plaintext be $Q = \langle Q_1, Q_2, \cdots, Q_{n-1} \rangle$ such that all $Q_i = 0$. Encrypt plaintexts P and Q in DCM to obtain ciphertexts C and D. This reveals two useful quantities:

1. We learn the last block of the ciphertext D,

$$D_n = E(0 \oplus x_n) \oplus x_0 = E(x_n) \oplus x_0.$$

2. We learn the first $n-1$ blocks of the ciphertext C, namely, $C_1, C_2, \ldots, C_{n-1}$.

Now, one can use this newly revealed information to construct a forged ciphertext

$$C' = \langle C_1, C_2, \ldots, C_{n-1}, D_n \rangle.$$

Clearly C' is different from C and D, so it is new ciphertext. When decrypted, C' yields the plaintext $P' = \langle P_1, P_2, \ldots, P_{n-1} \rangle$, which is a truncation of P that also differs from Q. Note that the checksum will be valid when decrypting C', so this forgery will remain undetected by the receiver.

One interesting feature of this attack is that it does not use any properties of the LSFR's connection polynomial f. It only uses the fact that the x_i sequences produced by the LFSR do not change between messages encrypted in the same key by DCM, since x_0 does not change.

Secrecy Attack. To illustrate a secrecy flaw of DCM, we show that an adversary can easily break DCM in a "real-or-random" sense in a chosen-plaintext attack [2].

Recall that in a real-or-random attack, a referee chooses a random, secret encryption key and flips a coin to decide whether to return the encryption of the (1) (real) plaintexts submitted to it by an adversary or (2) random strings of the same lengths as those of the (real) plaintexts submitted by the adversary. If the adversary can distinguish which decision was taken by the referee (i.e., determine the referee's coin flip) with a probability that exceeds 1/2 by a non-negligible amount following receipt of the ciphertext messages from the referee, the adversary is said to have broken DCM in a real-or-random sense.

In this setting, one possible attack goes as follows. Choose two plaintexts, $P = \langle 0 \rangle$ and $P' = \langle 0, 0 \rangle$, and learn their ciphertexts C, C' using the chosen-plaintext assumption. If their first blocks match, i.e., $C_1 = C'_1$, then the referee must have chosen the (real) strings P and P' for encryption; otherwise, the referee must have encrypted random strings of the same size as that of P^1 and P^2, and the adversary votes accordingly. This attack succeeds in breaking the secrecy of DCM with probability very close to 1.

2.3 DCM Definition – Second Variant

A second variant of the DCM mode is proposed by Boyle and Salter that is specifically designed for Internet use, i.e., with IPsec. This variant does not re-use the initial LFSR fill x_0 across packet streams and does not initialize the checksum to zero; instead, the initial fill and checksum definition include the IPsec sequence numbers, SEQ, and Security Parameter Index, SPI, as follows:

ENCRYPT($\mathrm{SEQ}, \mathrm{SPI}, P_1, \ldots, P_j$):
1. Set checksum $= \langle \mathrm{SEQ}, \mathrm{SPI}, \mathrm{padding} \rangle$.
2. Set $y_0 = x_0 \boxplus \langle \mathrm{SEQ}, \mathrm{SPI}, \mathrm{padding} \rangle$.
3. For $i = 1, \ldots, j$, do:
4. Set $y_i = f(y_{i-1})$.
5. Set $C_i = E(P_i \oplus y_i) \oplus y_i$.
6. Set checksum $=$ checksum $\oplus P_i$.
7. Set $y_{j+1} = f(y_j)$.
8. Set $C_{j+1} = E(\mathrm{checksum} \oplus y_{j+1}) \oplus y_0$.

Both SEQ and SPI are 32 bit numbers, and "padding" is the complement of the 64-bit number $\langle \mathrm{SEQ}, \mathrm{SPI} \rangle$ obtained by the concatenation of SEQ and SPI. Note that the addition defining y_0 is executed as a vector of 32-bit adds to minimize the hardware design and overhead.

Secrecy and Integrity Attacks. Now we show that an adversary can also mount chosen-plaintext attacks against this variant of DCM and thereby break its secrecy and integrity with non-negligible probability. Within a single security association (i.e., an SPI), the adversary can choose different SEQ values by encrypting appropriate number of packets within this security association. For our attacks, we use the following basic fact.

Claim. For a (non-singular) LFSR, the state-update function is linear, or in other words, $f(a \oplus b) = f(a) \oplus f(b)$.

Proof. As in the proof of the claim in Section 2.2, let $p_f(t)$ be the connection polynomial of the LFSR, and represent states of the LFSR as polynomials from $GF(2)[t]$, taken modulo $p_f(t)$. We have $f(x) = t \times x \bmod p_f(t)$, and so $f(a+b) = t \times (a + b) = ta + tb = f(a) + f(b) \bmod p_f(t)$, as claimed.

We are now ready to describe the attack. Let the adversary choose two plaintext strings P and Q such that the values of the initial LFSR fills y_0^P and y_0^Q used in the encryption of chosen strings P and Q (1) differ by a known constant c, and (2) have the relationship: $y_0^Q = y_0^P \oplus c$. Let the chosen sequence numbers for Q and P be SEQ^Q and SEQ^P, respectively. We claim that if an adversary can choose the sequence numbers SEQ^Q and SEP^P so that the stated relationship between the initial fills holds, then he can break both secrecy and integrity of the DCM variant for IP. Later we show that this hypothesis holds: an adversary can choose sequence numbers satisfying such a relationship with non-negligible probability and hence break DCM.

Claim. If the attacker can force the use of two sequence numbers SEQ^P and SEQ^Q satisfying $y_0^Q = y_0^P \oplus c$, where c is a known constant, then:

(a) the secrecy of DCM is broken; and
(b) the integrity of DCM is broken.

Proof. (a) The adversary chooses two plaintext strings P and Q each consisting of one block, $P = \langle P_1 \rangle$ and $Q = \langle P_1 \oplus f(c) \rangle$. We let C and D denote the corresponding ciphertexts obtained by encrypting the chosen plaintexts P and Q under DCM. By the definition of this DCM variant and the previous claim, we obtain the following equations:

$$\begin{aligned}
y_1^P &= f(y_0^P) \\
y_1^Q &= f(y_0^Q) = f(y_0^P \oplus c) = f(y_0^P) \oplus f(c) = y_1^P \oplus f(c) \\
C_1 &= E(P_1 \oplus y_1^P) \oplus y_1^P \\
D_1 &= E(Q_1 \oplus y_1^Q) \oplus y_1^Q = E(P_1 \oplus f(c) \oplus y_1^P \oplus f(c)) \oplus y_1^P \oplus f(c) \\
&= E(P_1 \oplus y_1^P) \oplus y_1^P \oplus f(c) = C_1 \oplus f(c).
\end{aligned}$$

The adversary can then break DCM in a real-or-random sense by simply choosing the plaintexts P, Q as above, obtaining putative ciphertexts C, D, and then checking whether $D_1 = C_1 \oplus f(c)$. If this condition holds, then these ciphertexts represent encryptions of the real strings sent by the adversary; otherwise, we received encryptions of random strings, and the adversary votes accordingly. This shows that this version of DCM is broken in a real-or-random sense if the hypothesis of the claim holds with non-negligible probability.

(b) The adversary applies the following attack. The adversary chooses a plaintext $P = \langle P_1, P_2 \rangle$ (with sequence number SEQ^P) and obtains a ciphertext $C = \langle C_1, C_2, C_3 \rangle$. Then, the adversary chooses a second plaintext $Q = \langle Q_1, Q_2 \rangle$ (with sequence number SEQ^Q chosen so that $y_0^Q = y_0^P \oplus c$) such that $Q_1 \oplus Q_2 = P_1 \oplus P_2 \oplus f(c) \oplus f^2(c)$, and he obtains a ciphertext $D = \langle D_1, D_2, D_3 \rangle$. (We write f^2 for the composition of f with itself, i.e., $f^2 = f \circ f$.) Then the adversary constructs his forgery using the sequence number SEQ^Q and the ciphertext

$$C' = \langle C_1 \oplus f(c), C_2 \oplus f^2(c), D_3 \rangle.$$

Since the forgery uses SEQ^Q, it has $y_0 = y_0^Q = y_0^P \oplus c$, $y_1 = y_1^Q = y_1^P \oplus f(c)$, and so on. Let $P' = \langle P_1', P_2' \rangle$ denote the decrypted plaintext, and let P_3' represent the decryption of the checksum block C_3'. We see that

$$\begin{aligned}
P_1' &= E^{-1}(C_1 \oplus f(c) \oplus y_1^Q) \oplus y_1^Q = E^{-1}(C_1 \oplus f(c) \oplus y_1^P \oplus f(c)) \oplus y_1^P \oplus f(c) \\
&= E^{-1}(C_1 \oplus y_1^P) \oplus y_1^P \oplus f(c) = P_1 \oplus f(c) \\
P_2' &= E^{-1}(C_2 \oplus f^2(c) \oplus y_2^Q) \oplus y_2^Q = E^{-1}(C_1 \oplus f^2(c) \oplus y_2^P \oplus f^2(c)) \oplus y_2^P \oplus f^2(c) \\
&= E^{-1}(C_2 \oplus y_2^P) \oplus y_2^P \oplus f^2(c) = P_2 \oplus f^2(c) \\
P_3' &= E^{-1}(D_3 \oplus y_3^Q) \oplus y_3^Q = \mathrm{checksum}^Q = \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle \oplus Q_1 \oplus Q_2.
\end{aligned}$$

Using the plaintext blocks P_1', P_2', P_3' computed above, the adversary verifies that the integrity check passes, as follows:

$$\begin{aligned}
P_3' &= \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle \oplus Q_1 \oplus Q_2 \\
&= \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle \oplus P_1 \oplus f(c) \oplus P_2 \oplus f^2(c) \\
&= \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle \oplus P_1' \oplus P_2'
\end{aligned}$$

In the above relation, the adversary uses the condition on the chosen plaintext blocks $Q_1 \oplus Q_2 = P_1 \oplus P_2 \oplus f(c) \oplus f^2(c)$. This completes the proof of the claim.

Next, we show that an adversary can choose the sequence numbers SEQ^P and SEQ^Q such that the relationship $y_0^Q = y_0^P \oplus c$ holds with high probability. Knowledge of IPsec packet formats enables an adversary to force certain sequence numbers for selected packets sent over a security association (i.e., using the same SPI) by the adversary making an appropriate choice of the amount of data sent.

Let $\mathrm{SEQ}^P = 100\cdots0$ and $\mathrm{SEQ}^Q = 110\cdots0$. (There are many other choices of sequence numbers that allow an adversary to obtain the desired relationship between the values of y_0^Q and y_0^P and constant c with non-negligible probability. The choice made here is optimized for ease of presentation.) Then, by the definition of padding, we can express the constants $\langle \mathrm{SEQ}^P, \mathrm{SPI}, \mathrm{padding}^P \rangle$ and $\langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle$ as the concatenation of four 32-bit blocks as follows:

$$\begin{aligned}
\langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle &= \langle 110\cdots0, \mathrm{SPI}, 001\cdots1, \overline{\mathrm{SPI}} \rangle. \\
\langle \mathrm{SEQ}^P, \mathrm{SPI}, \mathrm{padding}^P \rangle &= \langle 100\cdots0, \mathrm{SPI}, 011\cdots1, \overline{\mathrm{SPI}} \rangle
\end{aligned}$$

We define c as the vector difference of these two constants, noting that

$$\begin{aligned}
c &= \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle - \langle \mathrm{SEQ}^P, \mathrm{SPI}, \mathrm{padding}^P \rangle \\
&= \langle 010\cdots0, 0\cdots0, 110\cdots0, 0\cdots0 \rangle.
\end{aligned}$$

Hence, one can write

$$\begin{aligned}
y_0^Q &= x_0 \boxplus \langle \mathrm{SEQ}^Q, \mathrm{SPI}, \mathrm{padding}^Q \rangle \\
&= x_0 \boxplus \langle \mathrm{SEQ}^P, \mathrm{SPI}, \mathrm{padding}^P \rangle \boxplus c = y_0^P \boxplus c,
\end{aligned}$$

where the constant c is known (see above), and furthermore has only three bits non-zero. If the corresponding three bits in y_0^P are zero, then we obtain

$$y_0^Q = y_0^P \oplus c.$$

The probability that these three bits of y_0^P are zero is 1/8 because y_0^P is random and uniformly distributed (which is true since y_0^P is the addition of a constant to x_0, which DCM defines to be random and uniformly distributed.) Thus, it is clear that the adversary can find two sequence numbers SEQ^P and SEQ^Q such that $y_0^Q = y_0^P \oplus c$ with very high probability, where the constant c is known. The last statement completes the presentation of the secrecy and integrity attacks on the second variant of the DCM mode.

3 Discussion

The security problems of the first DCM variant are caused by the re-use of x_0, and all x_i sequences derived from it, for all messages encrypted with the same key. The simple modification of x_0 with each message, as presented in the second DCM variant, is very efficient but does not solve DCM's security problems. Since re-use of x_0 for multiple messages or its simple modification on a per message basis (i.e., using IPsec's sequence numbers and security parameter index) is what allows DCM to achieve its performance goals, there seems to be no easy way to eliminate this defect without generating an new x_0 for each message and thereby defeating the performance justification for DCM.

The submission of DCM to NIST's AES modes of operation effort marks the first time when NSA has publicly proposed an encryption mode. We welcome further public proposals by the NSA and encourage the DCM's authors to continue their efforts of evaluating the AES proposals. As they are undoubtedly aware, many past attempts at designing new modes of encryption and authentication have failed, yet these failures contributed significantly to advancing the state of the art.

References

1. M. Boyle, C. Salter, "Dual Counter Mode," July 4, 2001, available at *http://csrc.nist.gov/encryption/modes/proposedmodes*, August 1, 2001.
2. S. Goldwasser and M. Bellare, "Lecture Notes in Cryptography," August 1999, M.I.T Laboratory for Computer Science, available at *http://theory.lcs.mit.edu/ shafi* and at *http://www-cse.ucsd.edu/users/mihir.*
3. F.H. Hinsley and A. Stripp, *Codebreakers: the inside story of Bletchley Park*, Oxford University Press, 1993.
4. C.S. Jutla, "Encryption Modes with Almost Free Message Integrity," IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, manuscript, August 1, 2000. *http://eprint.iacr.org/2000/039.*
5. S. G. Stubblebine and V. D. Gligor, "On message integrity in cryptographic protocols", Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, 85-104, 1992.