Segmented Integer Counter Mode: Specification and Rationale

David A. McGrew
Cisco Systems, Inc.
170 West Tasman Drive, San Jose, CA 95134

October 19, 2000

1 Introduction

We propose a specification for counter mode which is useful for securing packetized media such as network protocols and low-level storage media (e.g., disk sectors). Our specification is simple, yet is flexible enough to adapt to different domains of use.

Counter mode is well described in [8], which summarizes its advantages and reviews its security properties. Essentially, counter mode defines an additive stream cipher, given a block cipher with a fixed key, in which the keystream is the concatenation of the output blocks of the cipher with the input blocks defined by the integers in ascending order. We call this encryption mechanism integer counter mode in order to contrast it with the Linear Feedback Shift Register (LFSR) counter mode described below.

In the following, we provide the specification detail needed to use the integer counter mode method to secure packetized media and an example application, we briefly review security considerations, and provide a rationale for our specification.

2 Integer Counter Mode

This section provides a complete specification for the use of Integer Counter Mode for packetized media, which we call the *segmented integer counter mode*, or SIC mode.

We represent the input to the cipher as an unsigned integer, and we represent all unsigned integers as having the least significant bit on the right. The block width of the cipher (that is, the number of bits in an input) is denoted w. The integer i corresponding to the block cipher input I is defined by $i = \sum_{j=0,w-1} 2^j i_j$, where i_j is the j^{th} least significant bit of i. The encryption function itself is denoted as E, and we do not explicitly denote the dependence of the cipher on a key because the key does not change during the generation of the keystream.

The keystream is logically divided into *segments*, each segment of which can be used to encrypt a single packet or other media sector. All segments are equally sized.

The input of the cipher is logically broken up into three parts: a *block index* b which identifies the blocks within a segment, a *segment index* s which identifies a segment within a keystream, and a *randomizer* r which is generated in an unpredictable manner. These three numbers are concatenated to form the inputs to the encryption function, with the block index as the least significant part, the sequence index as the middle part, and the randomizer as the most significant part. Symbolically, an input can be denoted as $r \mid s \mid b$, where the symbol $\mid$ denotes concatenation. Alternatively, the input can be denoted as $r2^{N_b+N_s} + s2^{N_b} + b$, where N_b and N_s are the number of bits in the block index and segment index, respectively.

The number N_b of bits in a block index determine the length of a segment to be $w2^{N_b}$. The value N_b must be determined before and remain fixed throughout keystream generation, and should be chosen to reflect the needs of the application. Admissible values are from zero to w.

The number N_s of bits in a segment index are determine the total number of segments in a keystream to be 2^{N_s}. The value N_s must be determined before and remain fixed throughout keystream generation, and should be chosen to reflect the needs of the application. Admissible values are from zero to w.

The length of the randomizer is $w - N_b - N_s$ bits. Admissible values are from zero to w. The total length of keystream produced is $w2^{N_s+N_b}$ bits.

A keystream segment is defined by the logical concatenation of the encryption function outputs, where the encryption function inputs have r and s fixed and b in ascending order from zero to $2^{N_b} - 1$. Using the unsigned integer convention described above (in which the 'first' bit of keystream is the rightmost), this is illustrated as

$$E(r \mid s \mid 2^{N_b} - 1) \mid \cdots \mid E(r \mid s \mid 2) \mid E(r \mid s \mid 1) \mid E(r \mid s \mid 0). \quad (1)$$

2.1 An Example

A strong motivation is to enable the use of counter mode for encryption protocols such as the IPsec Encapsulating Security Protocol (ESP) [5]. As an example, we describe a choice of SIC parameters for the Stream Cipher ESP [9]. In this case, the 32-bit ESP Sequence Number is used as the segment index, so $N_s = 32$. The maximum length of each IP packet is 2^{16} bytes (or 2^{19} bits). When $w = 128$ bits as in AES then $N_b = 12$. This leaves $N_r = 84$ bits for the randomizer.

Alternatively, if IP Jumbograms [2] are to be encrypted, then each segment must be 32 bits in length, so that $N_s = 32, N_b = 32$, and $N_r = 64$.

3 Security

The randomizer should be chosen to be as large as possible, in order to reduce the effectiveness of precomputation attacks. Such attacks may provide a slight reduction in the effective key size if the randomizer is zero, or has a small value [10, 9].

4 Rationale

Many good reasons for counter mode are identified in [8]. It suffices for us to add that counter mode is especially appropriate for network protocols.

Our decision to have the block index be the least significant part of the cipher input is intended to ensure that incrementing that index can be done by a single operation in software.

It is worth noting that SIC can be used without any segments by simply setting the parameter N_s to zero.

4.1 LFSR Counter Mode

An alternative to integer counter mode is the linear feedback shift register (LFSR) counter mode, in which an LFSR is used as the counter [11, 3]. Both constructions have the crucial property that no two inputs to the cipher will be the same for a given key.

We chose the integer approach for SIC over LFSR counter modes because the former is simpler, contains a greater degree of flexibility, and is slightly more efficient in software. SIC can accommodate any size segment with a trivial change in parameters, while LFSR based counter modes would require the definition of feedback polynomials for each possible size. Additionally, the mechanism that an LFSR must use to 'seek' to a segment boundary is different than its regular advance mechanism. This mechanism would need to be present for all feedback polynomials in order for LFSR counter mode to provide equivalent functionality to SIC mode.

We do not regard LFSR counter mode as having a security advantage over integer counter mode. If the block cipher is indistinguishable from random, then the security properties of both approaches are equivalent. If the block cipher is distinguishable from randomness, then the relative security of integer counter mode and LFSR counter mode will depend on the particular weakness of that cipher. Integer counters may be more vulnerable to differential cryptanalysis, if the cipher is differentially weak [12]. However, LFSR counter mode may have a similar vulnerability. Different definitions and extensions of differentials have been used against different ciphers (the exclusive-or definition of Biham and Shamir [1], the group-inverse definition of Lai and Massey [6], the linear-differential cryptanalysis of Langford and Hellman [7], truncated and higher-order differentials [4]), and it is reasonable to expect that an appropriate definition of a differential could target LFSR counter mode.

The ATM Security Specification includes a counter mode that is a hybrid of the integer and LFSR modes [3]. In that specification, the input to the block cipher is partitioned, with a 21-bit LFSR forming one part and sequencing information in other parts. We prefer SIC to this approach for the flexibility reasons cited above.

Acknowledgments

Thanks to Scott Fluhrer for useful discussions and comments.

References

[1] Biham, E., and Shamir, S., “Differential Cryptanalysis of DES-like Cryptosystems”, Proceedings of CRYPTO ’90: Advances in Cryptology, Springer-Verlag, 1990.

[2] Borman, D., Deering, S., Hinden, R., “IPv6 Jumbograms”, IETF Request for Comments RFC 2675.

[3] ATM Security Specification Version 1.0, Section 6.4, “The Counter Mode of Operation”, The ATM Forum Technical Committee, February 1999.

[4] Knudsen, L., “Truncated and Higher Order Differentials”, Proceedings of the Second International Fast Software Encryption Workshop, Springer-Verlag, 1995.

[5] Kent, S., and R. Atkinson, ”IP Encapsulating Security Payload (ESP)”, RFC 2406, November 1998.

[6] Lai, X., Massey, J., Murphys, S., “Markov Ciphers and Differential Cryptanalysis”, Proceedings of Eurocrypt ’91, Springer-Verlag, 1991.

[7] Langford, S., and Hellman, M. “Differential-Linear Cryptanalysis”, Proceedings of CRYPTO ’94: Advances in Cryptology, Springer-Verlag, 1994.

[8] Lipmaa, H., Rogaway, P., and Wagner, D., “Comments to NIST concerning AES Modes of Operation: CTR-Mode Encryption,” Manuscript, October, 2000.

[9] McGrew, D., Fluhrer, S., “The Stream Cipher Encapsulating Security Payload,” draft-mcgrew-ipsec-scesp-01.txt, Internet Draft, July, 2000.

[10] [MF00] McGrew, D., and Fluhrer, S., “Attacks on Encryption of Redundant Plaintext and Implications on Internet Security”, Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag, 2000.

[11] Schroeppel, R., “Comments for AES cipher selection”, Appendix C, electronic mail to NIST. Included in “Email Comments on Block Cipher Modes of Operation.” May 5, 2000.

[12] Schroeppel, R., “Encryption Modes”, electronic mail to NIST AES Modes of Operation Forum, `http://aes.nist.gov/default.htm` October 18, 200.