

Protecting the Integrity of Privacy-enhanced Electronic Mail with DES-Based Authentication Codes

Stuart G. Stubblebine[†] Virgil D. Gligor

Electrical Engineering Department
University of Maryland
College Park, Maryland 20742

Abstract

The Privacy-enhanced Electronic Mail supports integrity services with both symmetric and asymmetric keys. An option of the symmetric-key services is that of protecting message integrity with DES-based authentication codes. We discuss a vulnerability of this option to message-integrity attacks. We present a solution for the removal of this vulnerability that allows the retention of the DES-based authentication codes.

1. Introduction

The Privacy-enhanced Electronic Mail (PEM) [1, 2] supports confidentiality, integrity, and authentication of electronic mail in the Internet. These services use end-to-end cryptography between sender and receiver User Agent processes, with both symmetric and asymmetric keys, and do not impose any special processing requirements on the underlying Message Transfer System. An option of the symmetric-key services is that of protecting message integrity with Message Authentication Codes (MACs) which are computed by using the Data Encryption Standard (DES) in Cipher Block Chaining (CBC) mode.

An early version of the DES-MAC option in PEM [3] was shown to be vulnerable to integrity attacks against multiple-receiver messages [4]. When these messages were used, a receiver could create a bogus message and have it accepted by other receivers, or by the sender, without detection. The current version of PEM eliminates this vulnerability (1) by using a different key for the DES-MAC computation from that used for the CBC encryption of the user data, and (2) by recommending that at most a single receiver be named as an addressee of DES-MAC messages [1,2]. However, another message-integrity vulnerability of the DES-MAC checksummed messages remains uncorrected. We present this vulnerability, and propose a solution for its removal that allows the retention of the DES-based authentication codes. (A discussion of whether such authentication codes should be retained by PEM is beyond the scope of this paper. We assume that, since the DES-MAC checksum is sufficiently strong for most applications, is an international standard [5,6], and is supported by commercially available hardware [7], its use is desirable.)

2. Representation of PEM Messages

PEM services support both single- and multiple-receiver messages. The representation of a single-receiver, DES-MAC checksummed message, denoted by message type T1 in Figure 1, consists of an Initialization Vector (IV) for data encryption and decryption, an address field for the sender and for the receiver, a key field, a MAC field, and a user data field. Both the key field and the MAC field are encrypted under an Interchange Key (IK), which is shared between the sender and receiver.

[†] S.G. Stubblebine's current address is: USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292 - 6695. (stubblebine@isi.edu, gligor@eng.umd.edu)

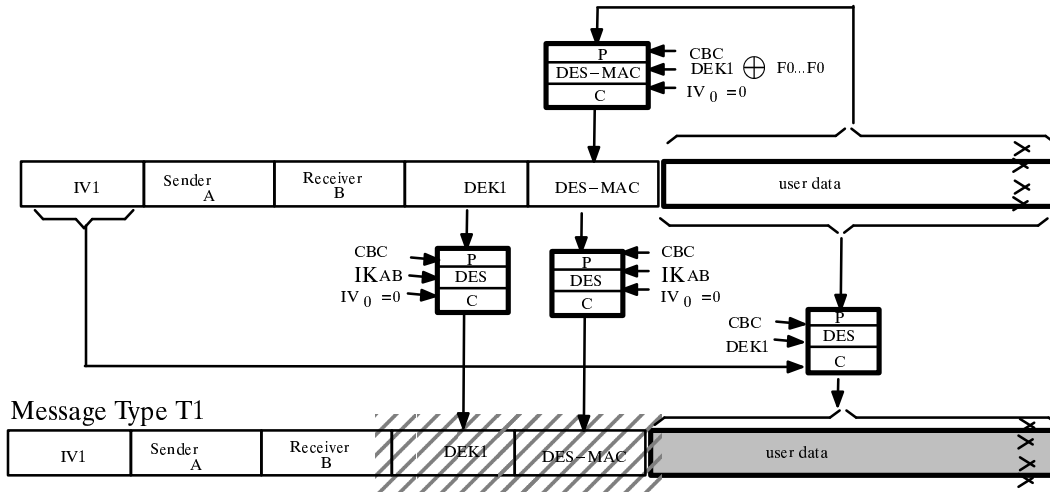


Figure 1. The Representation of PEM Messages using DES-MAC *

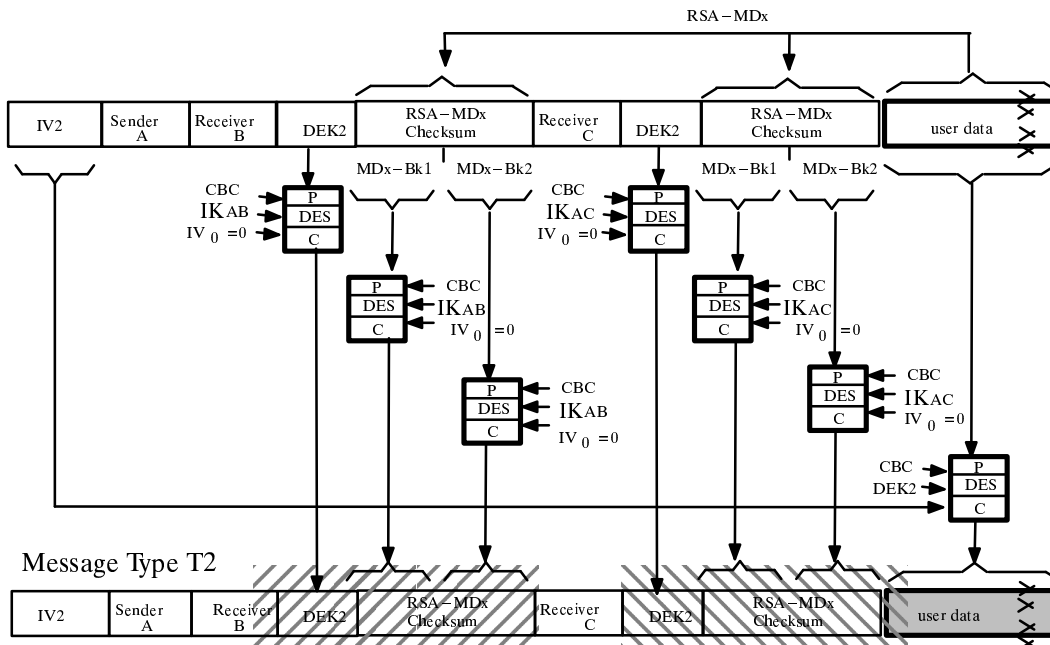


Figure 2. The Representation of PEM Messages using RSA-MDx*

The key field contains a random Data Encryption Key (DEK), which is used for the CBC encryption of the user data with the initialization vector, IV. It is also used for computing the DES-MAC of the user data in the CBC mode with a zero initialization vector, IV₀. When the DEK is used for computing the DES-MAC, the constant F0F0F0F0F0F0F0F0 is added (modulo two) to it. (This separation of the key used for DES-MAC computation from the key used to encrypt the user data is necessary to ensure that a bogus DES-MAC cannot be computed from message ciphertext.) A different DEK is used for each message. To enable detection of header modification, the user-data field must include the message header. (Since the header contains random fields, this requirement has the added benefit of guarding against chosen plaintext attacks.)

(*) The representation of both the single- and multiple-receiver messages omits details that are irrelevant to the subject of this note (viz., [1,2]).

The representation of a two–receiver message, denoted by message type T2 in Figure 2, differs from messages of type T1 in that (1) separate pairs of DEK and MAC fields are included for each receiver, and are encrypted in the interchange key, IK, that is shared between the sender and each receiver (e.g., IK_{AB} and IK_{AC} in Figure 2); and (2) the MAC is the result of computing RSA–MDx over the user data, where RSA–MDx can be RSA–MD2, MD4 or MD5 [8,9]. If communication between pairs of principals includes both messages of type T1 and T2, the same IK is used.

Throughout this paper, the notation $ENC(key, IV; P_1, \dots, P_n)$ and $DEC(key, IV; C_1, \dots, C_n)$ represents the DES–CBC encryption of plaintext P_1, \dots, P_n and decryption of ciphertext C_1, \dots, C_n with the key key and initialization vector IV . The notation $DES–MAC(key, IV; P_1, \dots, P_n)$ represents the computation of the Message Authentication Code of P_1, \dots, P_n with the key key and initialization vector IV , using the Data Encryption Standard (DES) in Cipher Block Chaining mode [5,6].

3. Attack Scenario

Suppose that principal P^A sends a type T2 message to principals P^B and P^C . Principal P^C intends to use this message to construct a bogus message of type T1 that would appear to be sent by principal P^A to principal P^B (or by principal P^B to principal P^A), as illustrated in Figure 3.

The construction of the bogus message of type T1 is illustrated in Figure 4. To construct this message, principal P^C uses the encrypted key block, $ENC(IK_{AB}, IV_0; DEK_2)$, of the received type T2 message in the place of the encrypted key block, $ENC(IK_{AB}, IV_0; DEK_1)$, of a legitimate type T1 message sent by P^A to P^B ; both blocks are encrypted under the interchange key shared by principals P^A and P^B , IK_{AB} , which remains unknown to P^C . Similarly, principal P^C uses the first block of the encrypted RSA–MDx checksum, $ENC(IK_{AB}, IV_0; MD_x–Bk1)$, in the place of encrypted DES–MAC checksum, $ENC(IK_{AB}, IV_0; DES–MAC)$, of a legitimate type T1 message sent by P^A to P^B . Since principal P^C knows the plaintext block of the RSA–MDx checksum, P^C chooses the plaintext blocks $P_{1_1} \dots P_{1_i}$ for the bogus user data so that the result of the DES–MAC computation over these data equals the first plaintext block of the RSA–MDx checksum, $MD_x–Bk1$; i.e., $DES–MAC(DEK_2 \oplus F_0 \dots F_0, IV_0; P_{1_1} \dots P_{1_i}) = MD_x–Bk1$.

Principal P^C 's choice of the first $i–1$ blocks, $P_{1_1} \dots P_{1_{i-1}}$, is unrestricted. However, to ensure that $DES–MAC(DEK_2 \oplus F_0 \dots F_0, IV_0; P_{1_1} \dots P_{1_i}) = MD_x–Bk1$, principal P^C must choose the last block P_{1_i} to equal $C_{2_{i-1}} \oplus P_{1_i}'$. To obtain $C_{2_{i-1}}$, P^C encrypts the first $i–1$ blocks, $P_{1_1} \dots P_{1_{i-1}}$, of the bogus user data under the key $DEK_2 \oplus F_0 \dots F_0$ and $IV_0=0$; and to obtain P_{1_i}' , P^C decrypts $MD_x–Bk1$ under key $DEK_2 \oplus F_0 \dots F_0$ and $IV_0=0$. Principal P^C can compute $C_{2_{i-1}}$ and P_{1_i}' because it knows both the key DEK_2 and the value of $MD_x–Bk1$; both are decrypted by P^C from the type T2 message received from principal P^A under the interchange key IK_{AC} and $IV_0=0$. However, as illustrated in Figure 4, the plaintext block P_{1_i} would appear garbled since it is defined as $C_{2_{i-1}} \oplus P_{1_i}'$. A receiver may or may not find this suspicious, depending upon the placement of that block in the message. By using similar choices of plaintext and ciphertext repeatedly, the placement of the garbled block within the bogus message becomes unrestricted.

The attack would be successful even if (1) a different block–cipher algorithm would be selected, not just DES, provided that the CBC mode would be used; (2) an initialization vector $IV_0 \neq 0$ would be used; (3) any known plaintext–ciphertext block pair that is encrypted under the key shared by principals P^A and P^B , IK_{AB} , would be used to construct the DES–MAC for the bogus type T1 message – not just the pair $\langle MD_x–Bk1, ENC(IK_{AB}, IV_0, MD_x–Bk1) \rangle$; and (4) a separate key per receiver would be used to compute a different DES–MAC value per receiver.

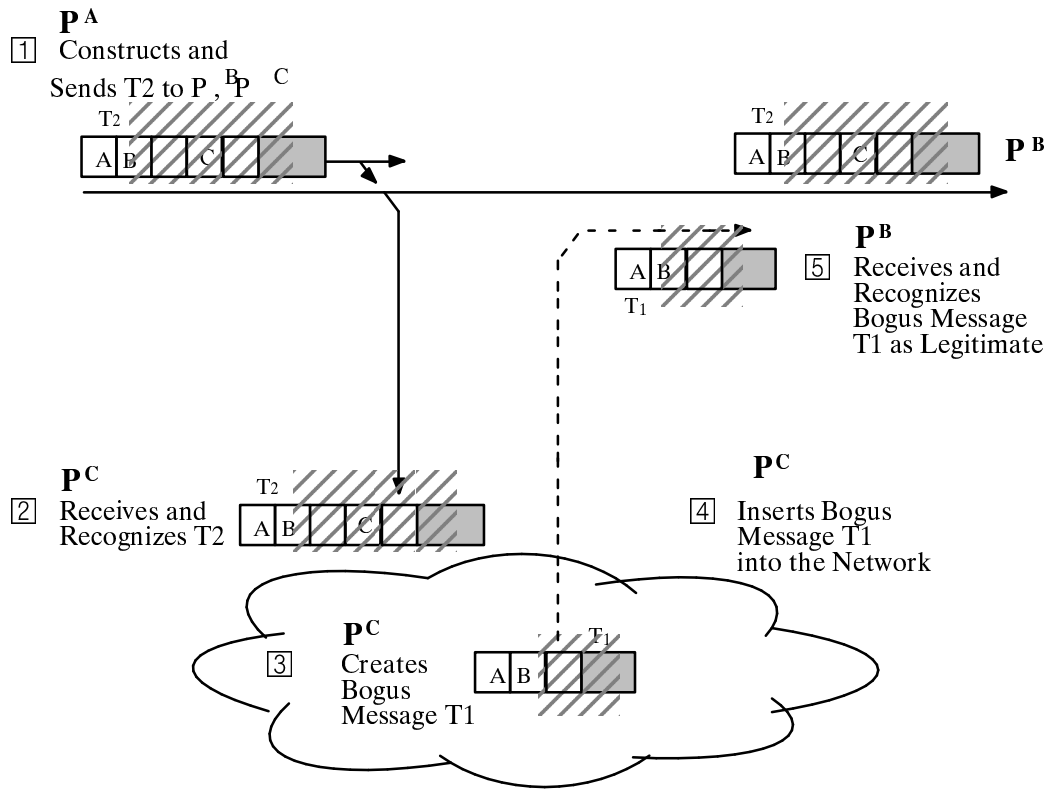


Figure 3. The Attack Scenario for the DES-MAC Option of PEM

4. Removing the Vulnerability of the DES-MAC Option in PEM

We suggest a two part solution for removing the PEM vulnerabilities posed by the use of DES-MAC checksums: the first part is proposed for single-receiver, DES-MAC checksummed messages, whereas the second part is proposed to allow use of the DES-MAC checksums for multiple-receiver messages.

In suggesting a solution, we make three desirable assumptions. First, we assume that the use of the same interchange key, IK , for both single- and multiple-receiver messages should be continued. This removes the task of acquiring an interchange key for each type of message. Second, we assume that a single data encryption key, DEK , is retained for each multiple-receiver checksummed message. This removes the task of re-encrypting a message for each message receiver. Third, we assume that the same checksum value (e.g., DES-MAC or RSA-MDx) is retained for each receiver of a multiple-receiver checksummed message. This eliminates the recomputation of a different DES-MAC for every receiver.

The three assumptions made above suggest that at least two known plaintext-ciphertext pairs would be available, namely $\langle DEK, ENC(IK, IV_0, DEK) \rangle$ and $\langle \text{checksum}, ENC(IK, IV_0; \text{checksum}) \rangle$ for every multi-receiver message. Thus, any solution must either eliminate these known pairs or must ensure that, despite the presence of known pairs such as those above, an attacker could not construct a bogus message that is recognizable within a probability threshold [10].

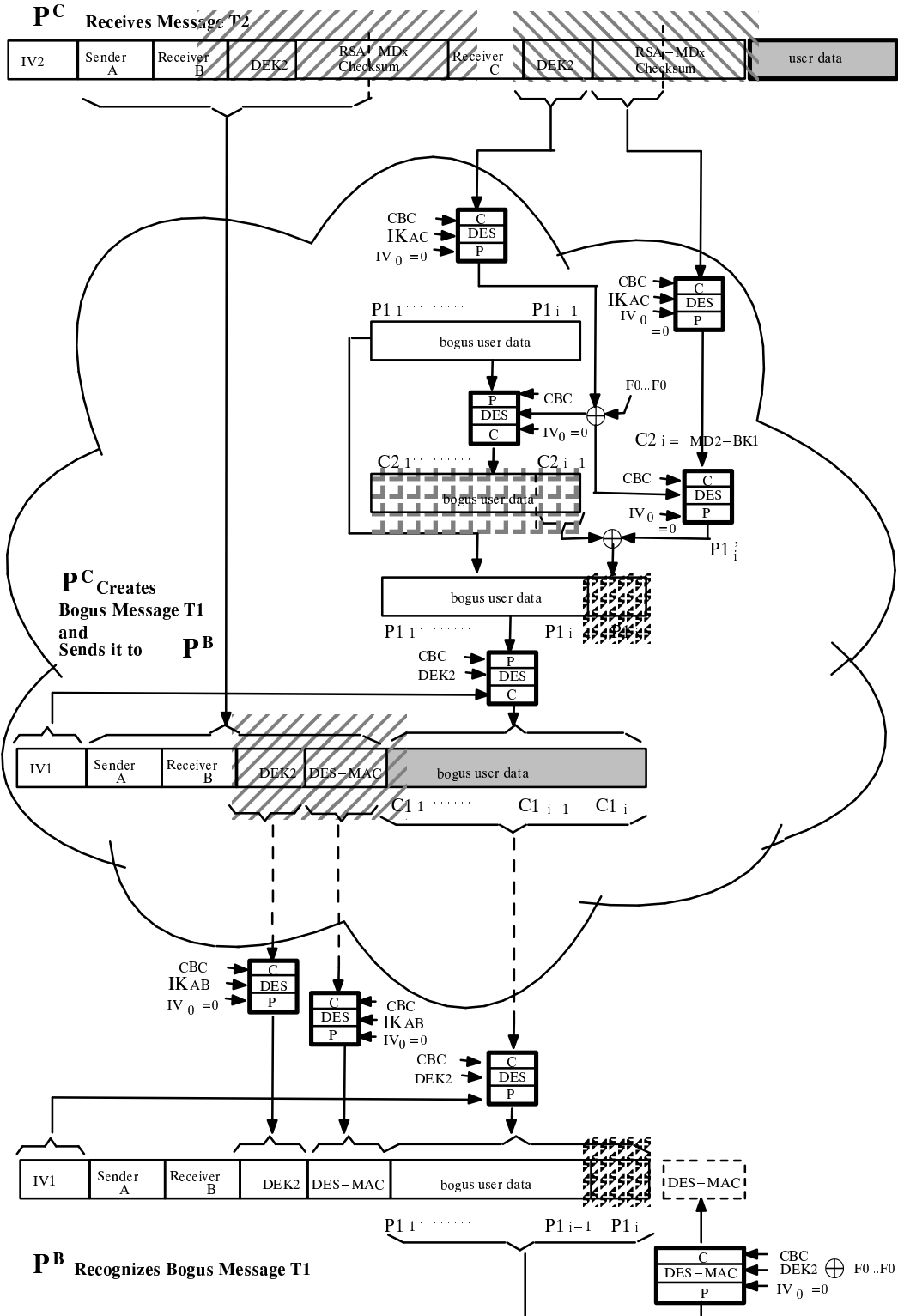


Figure 4. Creation and Recognition of a Bogus Message

The proposed solution for single-receiver messages has the effect of eliminating known pairs. This solution requires that a variant of the interchange key, $g(ik)$, be used to encrypt the DEK and

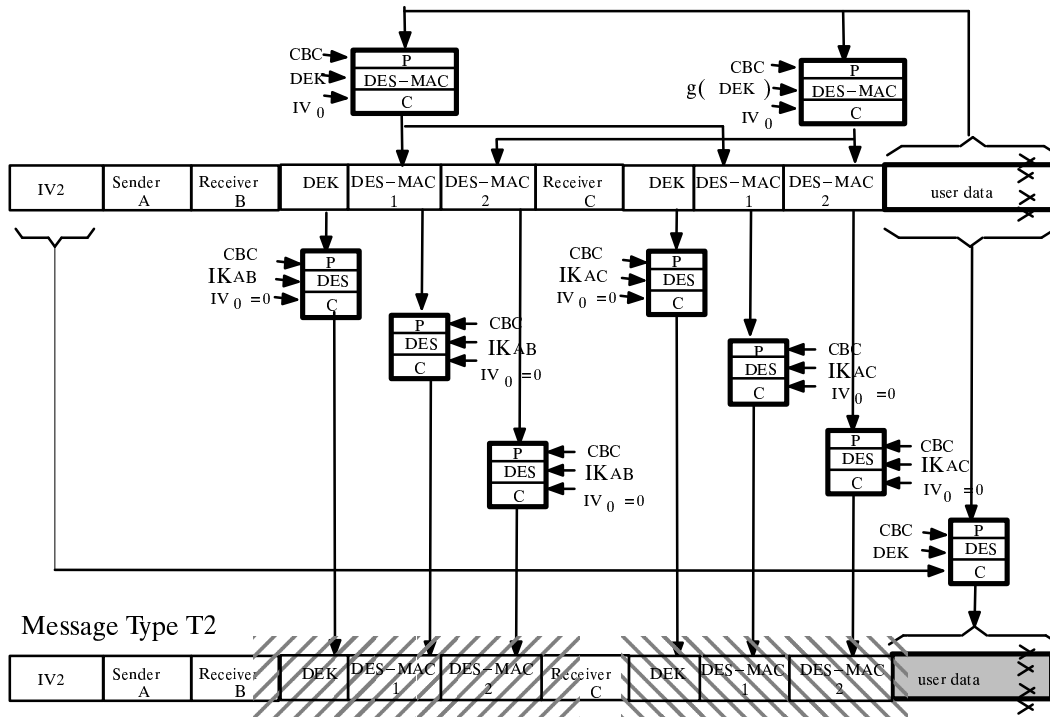


Figure 5. The Representation of Multi-receiver PEM Messages Using Double DES-MAC

DES-MAC value for type T1 messages. The function $g(\text{key})$ should be one-to-one, differ from the identity function, avoid weak or semi-weak keys, maintain key parity, change half of the key bits on the average, and neither weaken the cryptosystem nor unduly increase the probability of determining the secret key. The function $g(\text{key}) = \text{key} \oplus F0...F0$ seems to be a reasonable choice for this purpose.

As illustrated in Figure 5, the proposed solution that scales well for multiple-receiver messages reduces the impact that the presence of known pairs under IK (from Message Type T2 from both that shown in Figure 5 and also from Figure 2) has on the probability that an attacker can construct a bogus message. This solution takes the approach that the DES-MAC function is applied to the user data twice, first with key DEK and then with the key variant $g(\text{DEK})$, to obtain a *double DES-MAC*. The double DES-MAC and the DEK are then encrypted under the interchange key of each receiver in the same way the DEK and the checksum are encrypted in RSA-MDx messages.

This solution scales up well in the sense that its performance is independent of the number of message receivers. Also if IV_0 is set to be equal to the random IV_2 , then the separate encryption of the first DES-MAC component (i.e., DES-MAC 1) is avoided since it is identical to the last encrypted block of the user data.

The alternate double DES-MAC solution, using the above basic format with the exception that DES-MAC 2 is obtained by computing $\text{DES-MAC}(g(\text{DEK}), IV_0; C_1, \dots, C_n)$ (i.e., the second DES-MAC is computed over the ciphertext of the user data), is also adequate. In contrast, the alternative where $\text{DES-MAC 1} = \text{DES-MAC}(\text{DEK}, IV_0=0; P_1, \dots, P_n)$ and $\text{DES-MAC 2} = \text{DES-MAC}(\text{DEK}, IV_0=0; P_n, \dots, P_1)$ (i.e., DES-MAC 2 is computed over the plaintext in the reverse direction), which is called the *bi-directional MAC* in an early version of PEM [11], is somewhat inadequate. This is the case because bogus messages consisting of plaintext blocks (i.e., user data) arranged in palindrome format would be recognized as legitimate by User Agents. Of course, users will probably find it suspicious that half of the message would be garbled.

6. Conclusion

We provide yet another example of the need for using systematic message–integrity analysis and design methods in two ways. Successful message–integrity attacks are still possible against protocol options that are only informally analyzed. We proposed a solution for the removal of a symmetric–key vulnerability of the DES–MAC option in PEM that allows the retention of the DES–based authentication codes for both single– and multi–receiver messages. Since the integrity protection provided by any message type is largely dependent upon other message types in the protocol [12], the security of these solutions must be re–evaluated should existing message types change or other message types be added.

Acknowledgement

We thank John Linn and Dan Nessett for their comments on an earlier version of this paper. The work reported herein was supported by IBM Corporation under contracts YC313314 and MHVC2160. We are grateful to Tom Tamburo, Wen–Der Jiang, Marty Simmons, Curt Symes, Tom Russell, and Ping Lin for their continued support and encouragement.

References

- [1] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part I – – Message Encipherment and Authentication Procedures, Part II – – Certificate–Based Key Management, Part III – – Algorithms, Modes, and Identifiers,” Internet Working Group, RFC 1113 – 1115, August, 1989.
- [2] M. Bishop, “Privacy–enhanced Electronic Mail,” *Internetworking: Research and Experience*, Vol. 2, pp. 199–233, (1991)
- [3] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part I – – Message Encipherment and Authentication Procedures,” Internet Working Group, RFC–989, February, 1987.
- [4] C. Mitchell and M. Walker, “Solutions to the Multidestination Secure Electronic Mail Problem,” *Computers & Security*, Vol. 7(5), 1988, pp. 483–488.
- [5] Federal Information Processing Standards Publication 113, Computer Data Authentication, May 1985 (also, see ISO DP 8730).
- [6] American National Standard X9.9–1986, *American National Standard for Financial Institution Message Authentication (Wholesale)*, American Bankers Association, Washington (1986).
- [7] D.G. Abraham, G.M. Dolan, G.P. Double, and J.V. Stevens, “Transaction Security System,” *IBM Systems Journal*, Vol. 30, no. 2, 1991, pp. 206 – 229.
- [8] R. Rivest, “The MD4 Message Digest Algorithm,” Technical Memorandum 434, Laboratory for Computer Science, M.I.T., October, 1990.
- [9] R. Rivest, “The MD5 Message Digest Algorithm,” Internet Working Group, RFC 1321, April 1992.
- [10] S. G. Stubblebine and V. D. Gligor, “On Message Integrity in Cryptographic Protocols,” IEEE Symp. on Research on Security and Privacy, Oakland, Calif., May 1992, pp. 85 – 104 (also technical report TR – 2843, University of Maryland, College Park, Maryland 20742, February 1992.)
- [11] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part I – – Message Encipherment and Authentication Procedures,” Internet Working Group, RFC–1040, January, 1988.
- [12] S. G. Stubblebine and V. D. Gligor, “Message Integrity Design,” Draft.

(Publication information for this paper is as follows:

S. G. Stubblebine and V. D. Gligor, “Protecting the Integrity of Privacy–enhanced Electronic Mail with DES–Based Authentication Codes”, Proceedings of the PSRG Workshop on Network and Distributed Systems Security, San Diego, California, February 11–12, 1993.)