USENIX Association

# Proceedings of the 9th USENIX Security Symposium

Denver, Colorado, USA
August 14–17, 2000

# A Chosen Ciphertext Attack Against Several E-Mail Encryption Protocols

Jonathan Katz* Bruce Schneier†

## Abstract

Several security protocols (PGP, PEM, MOSS, S/MIME, PKCS#7, CMS, etc.) have been developed to provide confidentiality and authentication of electronic mail. These protocols are widely used and trusted for private communication over the Internet. We point out a potentially serious security hole in these protocols: any encrypted e-mail can be decrypted using a one-message, adaptive chosen-ciphertext attack which exploits the structure of the block cipher chaining modes used. Although such attacks seem to be of primarily theoretical interest, we argue that they are feasible in the networked systems in which these e-mail protocols are used. We suggest several solutions to protect against this class of attack.

## 1 Introduction

Electronic mail (e-mail) has become an essential communication tool. The ease of e-mail communication, when compared to traditional choices such as physical mail, fax, or telephone, makes it the communications medium of choice for many people. As more people and businesses move on-line, and Internet access becomes more commonplace, e-mail is expected to become even more important as a communications tool.

In order for e-mail to fully supplant other alternatives, businesses (and, to a lesser extent, individual users) must be assured of the privacy of their e-mail correspondence. This is of special concern in the case of e-mail, since it is no doubt easier for an adversary to "tap" into an Internet link than to tap into a phone line. Furthermore, e-mail has the potential of offering security beyond that of telephone conversations, as there is currently no good way of "scrambling" a telephone conversation (although one can detect — and with a bit more trouble, prevent — the "tapping" of the phone line in the first place). It is primarily for these reasons that e-mail encryption protocols were developed.

In any system, there are multiple points which an adversary can attack; of course, a system is only as secure as its weakest point of attack. In this paper, we point out a so-called *chosen ciphertext* attack which succeeds against all current implementations of e-mail security protocols. Furthermore, we argue that this attack is entirely feasible in the networked environment in which these e-mail security protocols are used.

## 2 Background

### 2.1 E-mail Encryption Protocols

The attack outlined herein is applicable to many different e-mail encryption protocols [21, 22]. In this paper, we explicitly consider attacks on OpenPGP [7, 10, 24] (the attack is also applicable to previous versions of PGP), S/MIME [20] (building upon CMS [14] and/or PKCS#7 [15]), PEM [18], and MOSS [8]. We refer the reader to the listed references for an in-depth description of these protocols; in this paper, we merely provide a high-level description necessary

*Department of Computer Science, Columbia University; jkatz@cs.columbia.edu.

†Counterpane Internet Security, Inc.; schneier@counterpane.com.

for a proper understanding of the attack. Specifically, the attack exploits the symmetric-key modes of encryption used, and therefore only this detail of the encryption protocols (which is the same for all protocols, at this level of description) is presented.

Consider an e-mail message (or file) $M = M_1, M_2, \ldots$, where the message $M$ is broken into a sequence of blocks of appropriate length for the underlying block cipher used (e.g. 64 bits for DES, 128 bits for AES). The protocols considered encrypt the message as follows:

1. A random "session-key" $K$ is generated.

2. $M$ is encrypted using a symmetric-key encryption algorithm (block cipher) and key $K$, using some mode of encryption (we discuss below the details of the mode used). This gives ciphertext $C_0, C_1, C_2, \ldots$ (note the generation of the additional ciphertext block $C_0$).

3. The session-key $K$ is encrypted using the recipient's public key. This is represented by $\mathcal{E}_{\rm pk}(K)$.

4. The following message is sent to the recipient: $< \mathcal{E}_{\rm pk}(K), C_0, C_1, \ldots >$.

The recipient, reversing the above steps, uses his private key to compute $K$; given $K$, the recipient can then use symmetric-key decryption to determine the original message $M$.

Note that a mode of encryption (in step 2, above) is necessary in order to allow for the encryption of messages which are longer than one block.

## 2.2 Chosen Ciphertext Attack

The attack presented here is known in the cryptographic literature as an adaptive chosen-ciphertext attack. The reader is referred elsewhere for formal definitions [2, 16], but a simple description is provided here. Assume an adversary intercepts ciphertext $C$ and is trying to determine the underlying plaintext $P = \mathcal{D}(C)$ (where $\mathcal{D}(\cdot)$ refers to decryption of the ciphertext). We refer to $C$ as the *challenge ciphertext*. Under an adaptive[1] chosen-ciphertext attack, the adversary may submit ciphertexts $C_1, C_2, \ldots$ of his choice to a *decryption oracle* which then returns the corresponding plaintexts $P_1 = \mathcal{D}(C_1), P_2 = \mathcal{D}(C_2), \ldots$. The adversary is allowed to use the information thus obtained to recover the desired plaintext $P$. Note that for the attack to be non-trivial the adversary is not allowed to submit the challenge ciphertext $C$ to the decryption oracle.

At first glance, this type of attack seems purely theoretical (when does an adversary have access to free decryption?), but consideration of the attack is of practical significance [12, 6, 4, 23, 13]. One can readily think of examples in which an adversary submitting a ciphertext to a user might obtain partial information about the decrypted plaintext. For instance, an adversary might be interacting with a computer which, when given some ciphertext, performs a specified action if and only if the ciphertext is *valid* (i.e., whose decryption corresponds to *some* plaintext); this allows an adversary to distinguish valid ciphertexts from invalid ones. Such an attack on the RSA Encryption Standard PKCS#1 has been demonstrated [4], leading to a feasible attack on certain implementations of the SSL V.3.0 protocol. A similar attack, called a "reaction attack," has been used to break several coding-theory based public-key cryptosystems [13]. In yet another example [12], the adversary communicates with a party on the network who responds to ciphertext messages only if the decryption of the message corresponds to valid English text. This, too, gives the adversary information about the decrypted plaintext which may prove useful in cryptanalysis.

# 3 Details of the Attack

We stress that we do not expose any weaknesses in the public-key (typically RSA or ElGamal) or symmetric-key (IDEA, CAST, or 3-DES) algorithms

[1] *Adaptivity* in this context means that the adversary is allowed to submit ciphertexts to the decryption oracle even after viewing the challenge ciphertext. In a non-adaptive attack, the adversary may only submit ciphertexts *before* viewing the challenge ciphertext.

used in the various encryption protocols. In fact, the attack is independent of the encryption algorithms used, and we therefore omit mention of specific algorithms when describing the attack. Rather, our attack focuses on the chaining mode used when encrypting messages longer than one block.

We begin with a description of the attack on the chaining mode used by OpenPGP [7]. OpenPGP uses a slight variation of Cipher Feedback (CFB) mode for symmetric encryption. Before encryption, the message $M'$ is prepended by a 10-octet string. The first 8 octets are random, and the 9th and 10th octets are copies of the 7th and 8th octets, respectively. The resulting text $M = M_1, M_2, \ldots, M_k$ is parsed as a sequence of $k$ blocks, each 64 bits long (for convenience, we assume a block cipher with 64-bit block size; the attack is similar for block ciphers with other block sizes). The OpenPGP variant of CFB-64 chaining mode is as follows ($\mathcal{E}_K(\cdot)$ represents application of the block cipher using session-key $K$):

```
**Encryption:** $c_0 = 0^{64}$
                for $i = 1$ to $k$:
                    $c_i = M_i \oplus \mathcal{E}_K(c_{i-1})$
                **Output:** $C_0, C_1, \ldots, C_k$

**Decryption:** for $i = 1$ to $k$:
                    $M_i = C_i \oplus \mathcal{E}_K(C_{i-1})$
                if ($C_0 = 0^{64}$) and (9th and 10th
                    octets match 7th and 8th octets)
                    **Output:** $M_1, M_2, \ldots, M_k$
                else
                    **Error**
```

Given ciphertext $<\mathcal{E}_{\text{pk}}(K), 0^{64}, C_1, \ldots, C_k>$, to obtain the value of block $M_i$ $(i > 2)$ one does the following:

1. Choose a (random) 64-bit number $r$.

2. Submit the ciphertext:
   $< \mathcal{E}_{\text{pk}}(K), 0^{64}, C_1, C_2, C_{i-1}, r >$.

3. Receive back the decryption $M' = M'_1, \ldots, M'_4$, where $M'_4 = r \oplus \mathcal{E}_K(C_{i-1})$.

4. Compute $M_i = M'_4 \oplus r \oplus C_i$.

(Note that this also allows determination of block $M_2$.) If there is concern that the decrypted message will appear too similar to the original message, a random string can be inserted between $C_2$ and $C_{i-1}$; also, note that the last 6 octets of $C_2$ can be randomly chosen.

Other chosen ciphertext attacks are also possible. For example, submitting:

$$< \mathcal{E}_{\text{pk}}(K), 0^{64}, C_1, C_2^{16} r_1, C_3, r_2, \ldots, C_k, r_{k-1} >,$$

where $C_2^{16}$ represents the first 16 bits of $C_2$, $r_1$ is a random 48-bit string, and $r_2, \ldots, r_{k-1}$ are random 64-bit strings, allows the adversary to compute the entire contents of the original message. The feasibility of any particular attack depends on the specifics of the "decryption oracle access" assumed available (which depends upon the behavior of the recipient of the original message; see Section 4).

This type of attack is not limited to protocols using CFB mode. CBC mode, used by PEM [18] and CMS [14], is also vulnerable to a chosen ciphertext attack, as are all other "popular" modes of encryption [22] (including ECB, OFB, PCBC, and counter mode). Note further that, due to the reliance of CBC and CFB modes on XOR operations, individual bits of selected plaintext blocks can be "set" at will by an adversary mounting an adaptive chosen ciphertext attack. In particular, this allows the adversary to circumvent any redundancies required by protocols using these modes.

# 4 Feasibility of the Attack

A chosen ciphertext attack in the context of e-mail encryption is certainly feasible. Imagine a situation in which *User* has configured his e-mail handler to automatically decrypt any incoming e-mails encrypted using PGP. An adversary *Adv* intercepts a PGP-encrypted message $C$ sent to *User*, and wants to determine the contents of this message. Adversary *Adv* constructs $C'$ according to the above algorithm, and sends this message to *User*. Then, *User*'s e-mail handler automatically decrypts $C'$, and *User* reads the corresponding message $M'$. To *User*, the message appears garbled; he therefore replies to *Adv* with, for example, “What were you trying to send me?”, but also *quotes the “garbled” message $M'$*. *Adv* receives the plaintext $M'$ which he wanted, and can use this to determine the original message. If *User* does not send *Adv* the garbled message, *Adv* can request it (for example: “I don't know what happened. Send me the file you decrypted and I'll try to figure it out.”). This is not an unreasonable feat of social engineering. Note that this attack works even if all e-mail sent by *User* is encrypted[2]; when *User* responds to *Adv*'s garbled message, he encrypts his response using *Adv*'s public key (and thus *Adv* has access to this response, anyway).

In the setting described above it is important that $M'$ not look too similar to $M$, because otherwise the original recipient may become suspicious. For example, the adversary will likely gain nothing if the submitted ciphertext decrypts to a plaintext which is 90% identical to the original message (which is technically allowed under a chosen ciphertext attack); in this case, the original recipient will certainly realize that something strange is going on.

Sometimes, messages are compressed before being encrypted [7]. This makes the attack more difficult for an adversary (due to the redundancy a valid compressed message must contain), but by no means precludes such an attack. Recall that the modes of operation used for encryption are “local” (in the sense that a plaintext block is affected by only very few nearby ciphertext blocks) and therefore the redundancy of the plaintext can be maintained via suitable alteration of the ciphertext. Furthermore, since the compression function is reversible and publicly known, the chosen ciphertext attack given above can be modified for this setting without difficulty.

# 5 Recommendations

We can immediately suggest some possible ways to prevent the attacks outlined herein. The simplest solution is for users not to reply to “garbage” e-mails by quoting the garbage message in their reply. This seems quite limiting — what if the message legitimately got corrupted in transit? It also relies too heavily on the behavior of users of the system.

Another solution is to demand that all encrypted messages be signed, and to not respond to unsigned messages with quotes from those messages. This is also limiting — it is common for people to send encrypted-but-unsigned e-mail — and the extra computational overhead should be avoided. It also shares many of the problems of the previous solution.

Yet another solution is to have the e-mail decryption software store all session keys which have been used so far in messages sent to the user. (Actually, the program should store a one-way hash of each session key, to avoid the problem of someone breaking into the user's files and obtaining a list of previously-used session keys.) Note that the chosen ciphertext attack described in Section 3 sends a previously-used session key as part of the ciphertext (in fact, it uses the same public-key encryption of the session key). The probability of this happening by chance is extremely low. A warning could be generated whenever a session key is repeated, to alert the user to be careful when responding to that particular message. Again, this solution is not completely satisfying. If encrypted e-mail becomes ubiquitous, storage of all session keys used becomes cumbersome.

A first step, and one which at least eliminates the “simple” attacks outlined in this paper, is for protocols to use a mode of encryption which is itself secure under adaptive chosen ciphertext attack [19, 5, 17]. Unfortunately, the various modes currently known to be secure in this setting each have their own drawbacks: expansion of the ciphertext length [17], or requirement of additional primitives [19] and multiple keys [19, 5] (which must then be encrypted using slow public-key methods). We hope that this paper will encourage additional research toward *efficient* one-key modes of encryption secure under chosen ciphertext attacks[3].

However, even this would not be sufficient to guarantee full security against an arbitrary adaptive chosen ciphertext attack. An adversary could mount a

[2] After all, the assumption is that this user is security conscious in the first place!

[3] A recently suggested mode, integrity-aware CBC (iaCBC), was subsequently broken. No provably-secure fix is currently known [11].

chosen ciphertext attack against the public-key algorithm itself, or exploit some interaction between the public-key algorithm and the block cipher being used (although no such attacks are known at present). Recent work on chosen-ciphertext secure hybrid encryption schemes [1, 9] (where *hybrid* implies integration of public-key and symmetric-key [i.e., block cipher] encryption) can be used achieve this goal. Unfortunately, currently proposed solutions are only heuristically secure, in that they are proven secure in a model in which a hash function is treated as a random function. More work in this direction is needed.

Another possibility (if one is willing to consider multiple-key approaches) is to generate two session keys: one for encryption and one for authentication. One then encrypts both keys using a public-key algorithm, encrypts the message using the first key (as before), and additionally appends a message authentication code (using the second key) of the ciphertext. Various approaches along these lines have been considered, and an exact security analysis of these approaches appears elsewhere [3].

## References

[1] M. Abdalla, M. Bellare, and P. Rogaway, "DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem," Manuscript, September 1998. Available at `http://www.cs.ucdavis.edu/~rogaway.`

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes," *Advances in Cryptology — CRYPTO '98 Proceedings*, Springer-Verlag, 1998, pp. 26–45.

[3] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations Among Notions and Analysis of the Generic Composition Paradigm," Manuscript, May 2000. Available at `http://eprint.iacr.org.`

[4] D. Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on RSA Encryption Standard PKCS#1," *Advances in Cryptology — CRYPTO '98 Proceedings*, Springer-Verlag, 1998, pp. 1–12.

[5] D. Bleichenbacher and A. Desai, "A Construction of a Super-Pseudorandom Cipher," Manuscript, February 1999.

[6] M. Blum, P. Feldman, and S. Micali, "Proving Security Against Chosen Ciphertext Attacks," *Advances in Cryptology — CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 256–268.

[7] J. Callas, L. Donnerhacke, M. Finney, and R. Thayer, "OpenPGP Message Format," RFC 2440, Nov 1998.

[8] S. Crocker, N. Freed, J. Galvin, and S. Murphy, "MIME Object Security Services," RFC 1848, Oct 1995.

[9] E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," *Advances in Cryptology — CRYPTO '99 Proceedings*, Springer-Verlag, 1999, pp. 537–554.

[10] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly & Associates, 1995.

[11] V. Gligor, Personal Communication, March 2000.

[12] S. Goldwasser, S. Micali, and P. Tong, "Why and How to Establish a Private Code on a Public Network," *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science* 1982, pp. 134–144.

[13] C. Hall, I. Goldberg, and B. Schneier, "Reaction Attacks Against Several Public-Key Cryptosystems," *Proceedings of Information and Communication Security*, Springer-Verlag, 1999, pp. 2–12.

[14] R. Housley, "Cryptographic Message Syntax," RFC 2630, Jun 1999.

[15] B. Kaliski, "PKCS #7: Cryptographic Message Syntax, Version 1.5," RFC 2315, Mar 1998.

[16] J. Katz and M. Yung, "Complete Characterization of Security Notions for Probabilistic Private-Key Encryption," *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing* 2000.

[17] J. Katz and M. Yung, "Unforgeable Encryption and Chosen-Ciphertext Secure Modes of Operation," *Fast Software Encryption — FSE '00 Proceedings*, Springer-Verlag, 2000.

[18] J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures," RFC 1421, Feb 1993.

[19] M. Naor and O. Reingold, "On the Construction of Pseudo-Random Permutations: Luby-Rackoff Revisited," *Journal of Cryptology* 12(1): 29-66 (1999).

[20] B. Ramsdell, "S/MIME Version 3 Message Specification," RFC 2633, June 1999.

[21] B. Schneier, *E-Mail Security*, John Wiley & Sons, 1995.

[22] B. Schneier, *Applied Cryptography, 2nd Edition*, John Wiley & Sons, 1996.

[23] V. Shoup, "Why Chosen Ciphertext Security Matters," IBM Research Report RZ 3076, November, 1998.

[24] P. Zimmerman, *The Official PGP User's Guide*, MIT Press, 1995.