Security Aspects of a UNIX PEM Implementation

James M. Galvin <galvin@tis.com>
David M. Balenson <balenson@tis.com>

Trusted Information Systems, Incorporated*
3060 Washington Road
Glenwood, MD 21738

Abstract

Trusted Information Systems has designed and implemented a UNIX[1]-based version of Privacy Enhanced Mail (PEM). The PEM protocols enhance the services provided to users of Internet electronic mail by including the following security services: message integrity, message origin authentication, non-repudiation of origin, and (optional) message confidentiality. These security services provide cryptographic protection to messages transported between end systems by the message transfer system. Of paramount importance is the proper design and implementation of the PEM software, and the proper management and use of the end system hosting the PEM software. The TIS/PEM system is designed and implemented with a number of points of control which can be combined to provide varying levels of protection within a host system.

1 Introduction

Privacy Enhanced Mail (PEM) enhances the services provided to users of Internet electronic mail by including the following security services: message integrity, message origin authentication, non-repudiation of origin, and (optional) message confidentiality. When deciding which services to offer, an essential principle was to concentrate on the set of services that would provide significant and tangible benefits to a broad user community, maximizing the added value with a modest level of implementation effort.

An implementation of PEM needs to be concerned with an additional issue: how to assure the user that the services supported are indeed provided according to the specification and local requirements. The local requirements would be specified in a local Security Policy. An implementation should, as much as possible, accommodate a broad range of security

*This work was partially supported by the U. S. Government Defense Advanced Research Projects Agency under contract number F30602-89-C-0125 to Trusted Information Systems, Inc.

[1]UNIX is a registered trademark of Unix System Labs.

policies. Trusted Information Systems designed and implemented a UNIX-based, openly-available version of PEM (TIS/PEM) with several configurable points of control. The points of control are independent mechanisms that can be combined to support a broad range of security policies.

In this paper we first review the basics of the PEM specifications and describe the organization of the TIS/PEM system. We then explore the various threats to a PEM system residing on a UNIX host and describe the specific points of control, and in particular, the underlying UNIX protection mechanisms, employed by the TIS/PEM system.

2 Privacy Enhanced Mail

Privacy Enhanced Mail has evolved considerably since it was first published [11, 12]. In August 1989, a suite of three documents were published to define the PEM protocol [13], its ancillary infrastructure [9], and the cryptographic algorithms required [14]. Earlier this year, these documents where retired to "Historic Status", and the PEM working group of the Internet Engineering Task Force completed new revisions [10, 8, 2] to the original three documents, and added a fourth document [6] describing how to use PEM to perform the initialization necessary to join the Internet PEM infrastructure. These four documents are expected to be published as Proposed Internet Standards.

The PEM message processing specification defines security enhancements to Internet electronic mail, i.e., text mail defined by RFC 822 [4] and transported principally via SMTP [15]. There are 4 security services provided by PEM:

1. Message integrity — the property that the message has not been altered or destroyed in an unauthorized manner.
2. Message origin authentication — the corroboration that the originator of the message is who it purports to be.
3. Message confidentiality — the property that the message is not made available or disclosed to unauthorized individuals, entities, or processes.
4. Non-repudiation of origin — the property that the originator can not deny having sent the message.

The security enhancements are designed to be integrated with electronic mail user agents. Figure 1 depicts the standard message handling system (MHS) model. A user agent (UA) is responsible for interacting with users and preparing a message to be transported to its recipients. The transport responsibility is relegated to the message transfer system (MTS) by the originator's user agent upon submission of the message to its local message transfer agent (MTA). The message is relayed via the MTS to the recipient's local MTA, which delivers the message to the recipient's UA.

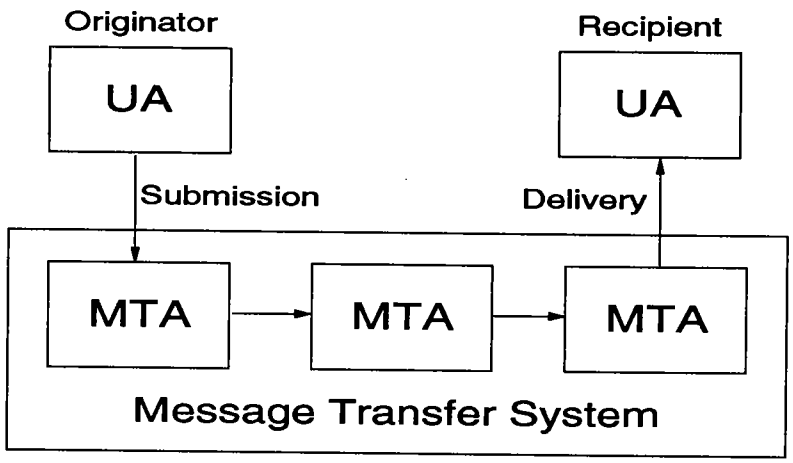

Figure 1: Basic Message Handling System Model

In the context of electronic mail, PEM may be integrated either:

- above a user agent — PEM processing proceeds prior to the user interacting with the UA.
- within a user agent — PEM processing proceeds as explicitly directed by a user of the UA.
- after a user agent — PEM processing proceeds after the user has interacted with the UA but prior to submission of the message to the MTS.

The PEM protocol is specifically designed to be non-invasive with respect to the MTS. This facilitates both its selective deployment at end systems, and its selective use by users. It also relegates the responsibility for the assurance that a PEM implementation provides overall protection to the end system. In particular, for implementations designed to function in multi-user environments, the identification of the user and the protection of the mechanisms used by that user are paramount.

Cryptography is the principle mechanism employed to support the security services. In particular:

1. A message integrity check (MIC) value is computed using a message digest algorithm, for example MD2 [7] or MD5 [16]. This value is transported with the message and recomputed by the recipient to verify the integrity of the message.

2. A signature is computed using an asymmetric signature algorithm, for example RSA [17]. The MIC value is signed by the originator in order to both protect it on its way to the recipient and to support identification of the origin of the message.

3. The message is encrypted with a symmetric encryption algorithm, for example DES [5, 1]. The key used for the encryption is generated for each new message, encrypted once for each intended recipient using an asymmetric encryption algorithm (for example [17]), and is transported, along with any other parameters required by the symmetric algorithm, with the message to enable the authorized recipient(s) to recover the original message.

4. When an asymmetric signature algorithm is used, use of the private component of the key pair to sign the MIC value provides for non-repudiation of origin and allows for the authenticated message to be forwarded to additional recipients with the authentication intact.

In principle the services could be applied in arbitrary combinations but in practice two combinations are common: integrity and authentication are applied to messages or all services are applied to messages. By convention, the former are called integrity-protected messages and the latter encrypted messages. Integrity-protected messages may optionally be encoded to guarantee the text of the message will not be modified by relaying mail systems, many of which make arbitrary changes to the format of text messages as they pass through the system. However, backward compatibility is enhanced if messages are not encoded since non-PEM compliant user agents may still read the messages even though they will not be able to verify the signature.

Although the PEM protocol is intended to be compatible with a broad range of key management strategies, including those based on both symmetric (secret key) and asymmetric (public key) cryptographic technologies, a public key certificate-based strategy [8], modeled after the directory authentication framework [18], is being specified first. It will provide for an infrastructure compatible with a CCITT X.509 certificate-based key management infrastructure.

The proper association of a public key with an individual user is essential to the secure use of the PEM cryptographic-based security services. The originator of a confidential PEM message must be assured of the identity of each of the intended recipient(s) of the message. Similarly, the recipient(s) of signed PEM message must be assured of the identity of the purported originator. Thus, at the heart of PEM's key management strategy is the use of a public key certificate, which carries a user's distinguished name, public key and other parameters including a version, a serial number, a validity period, and the name of its issuer. The integrity and authenticity of this information, and more importantly, the binding of the specified key to the specified identity, is vouched for by an issuer who signs the certificate. Applied recursively, this yields a hierarchy of issuer and user certificates, and it allows PEM users to authenticate the identities of other users and assure the use of those users' respective public keys.

Finally, the specification of the actual cryptographic algorithms employed by PEM appears in a separate document [2] so that each algorithm may be described individually and easily

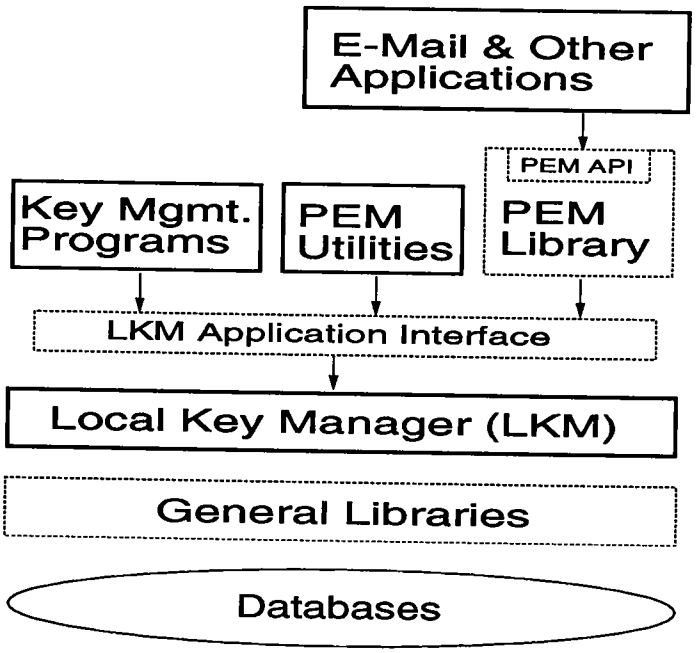

Figure 2: Overview of the TIS/PEM system

revised or replaced, if necessary. This convenience emphasizes the fact that PEM is not dependent on any particular choice of algorithms. In fact, the PEM protocol explicitly allows for the choice of algorithms to be increased or changed in the future. This document and its successors provide definitions, references, and citations for the current algorithms, usage modes, and associated identifiers and parameters used in support of PEM.

3 TIS/PEM

Trusted Information Systems has developed an openly available, UNIX-based implementation of PEM (TIS/PEM). It was designed for multi-user operation on BSD and SYSTEM V derived UNIX systems. It will be distributed in source code form, except for the RSA cryptographic functions, which will be provided as an object library.

An overview of the TIS/PEM system appears in Figure 2, described below. The system is comprised of a collection of applications and other interfaces, libraries and utility programs.

The PEM message processing functionality is implemented as a library with a well-defined application programmers interface. This allows for straightforward integration with other user agents and applications. These library routines depend on the presence of a number of support libraries, but they are the only entry points that need to be integrated into other applications. The routines are designed to function as filters, operating on a text input

stream and providing a text output stream. Typically, an application links directly with the PEM library. Alternatively, the application can execute the filter programs.

One of the entry points to the PEM library is used to PEM enhance a text message. It performs all of the required processing steps based on the setting of a bit flag in its argument list. This includes the retrieval of required certificates, canonicalization of the input stream, MIC and signature computation, encryption key generation and message encryption, printable encoding, and the addition of header fields containing control information necessary for the de-enhancement process.

Another entry point is used to de-enhance PEM messages. It expects the output of the sending entry point as input and also performs all of the required steps. The processing is directed by the control information found in the input, including parsing of the control information, validation of certificates found in the control information, retrieval of required certificates not found in the input, printable decoding, message decryption, MIC and signature verification, and decanonicalization.

We have integrated TIS/PEM with the Rand MH Message Handling System (Version 6.7.2). In addition, we have developed a couple of other applications, constructed as UNIX *cat*-like filters, useful for applying PEM to non-mail related objects, for example files.

The Local Key Manager (LKM), as its name implies, is responsible for all the local key management activities on its host system. These activities include the following.

- Maintaining a database of local users' private keys.
- Providing controlled access to private keys for use in signing messages and in decrypting message encryption keys.
- Maintaining a database of both local and remote users' public key certificates.
- Providing access to validated certificates.
- Registering a local user, that is, the generation of a public/private key pair and the construction and signing of a certificate embodying the public key.

The LKM is implemented as a stand-alone program with a well-defined interface. An application uses the services of the LKM by linking with an LKM application interface library which handles all interactions (establishment of a pipe, invocation via fork-and-exec, data communications, etc.) between the application program and the LKM.

Finally, a set of ancillary application programs, including key management programs and various utilities are provided that make use of the LKM interface to support administrators and users of the TIS/PEM system.

4 Threats

The use of cryptographic mechanisms within the PEM message processing protocols provides protection between the originator's UA and the recipient's UA. In particular, the mechanisms provide protection within the end systems (from the UA level down to the network) and across the network (between the end systems). Given the reliance on cryptographic mechanisms, an implementation is vulnerable to attacks on the cryptographic keys themselves (for example, substitution or unauthorized use), to attacks on the PEM software (for example, trojan horses), and to attacks targeted on system components between the user and the user agent (for example, capturing key strokes or redirecting display output).

For each of the above vulnerabilities we examined the threats and have identified 5 classes of protection from them:

- Protection of a user from himself or herself
- Protection of a user from other users
- Protection of a user from the system administrator
- Protection of a user from the organization
- Protection of an organization from a user

In protecting a user from him or herself, the principle concern is preventing accidental misuse of the system. Typically, the mechanisms employed are found principally in the user interfaces, where a user is always asked to confirm destructive functions. In addition, since the cryptographic keys used are identified by distinguished names, the name bound to each key is always displayed for the user.

A common concern for users and their respective system administrators is knowing if the system is vulnerable to idle terminals or workstations. In many environments it is common for users to login when they first arrive and remain logged in until they are ready to leave.

Although it is not possible (in an ordinary UNIX environment) to protect against maliciously installed trojan horses, it is possible to provide some protection from inquiring users with system administrator privileges. For example, users' login passwords are stored on a UNIX machine in a one-way encrypted form, thus keeping knowledge of the actual value a secret from system administrators.

Many organizations accord users some personal privacy in the work place. For example, a user's desk or computer files may be considered private. To protect a user from an organization required support for extending local policies to protect a user's PEM messages.

In contrast, organizations have as much, if not more, responsibility to protect themselves. For example, some cryptographic mechanisms are based on patented techniques requiring

the payment of a fee for their use. An organization may need to restrict who uses PEM or what they use it for.

Of paramount importance is the proper use and management of the system hosting the PEM software. A PEM implementation should complement proper administration and reduce the risks from the threats identified above. In TIS/PEM we have designed and implemented a number of points of control, which can be combined to support a broad range of security policies.

5 Points of Control

The LKM of TIS/PEM supports several points of control, each of which includes several mechanisms that can be used to achieve the desired protection. The mechanisms are enabled at compile time, run time, or as a result of administrative fiat. These points of control can be used to support a variety of security policies. In particular, organizations may rank the classes of protection listed in the previous section in order of priority, and use the ordered list to balance the controls described below with the needs of the organization and its PEM user community.

There are four points of control, each with several options. Each control point requires knowledge of the identity of the user. The identification of the user is obtained from the UNIX user's uid. This value is combined with the name of the local host to create a UNIX specific unique identifier, called a host-uid or huid for short. This identifier is only required to be unique across the local environment, i.e., the set of users who consider themselves part of the same PEM user community. Succeeding sections describe each point of control.

5.1 Authorization

In order for a user to make use of the TIS/PEM system, the user's huid must be known to the system, i.e. the system must be configured to recognize each user of the PEM services. This configuration includes designating a few users as privileged. It is the privileged users who are responsible for configuring TIS/PEM to recognize all other users of its services. As expected, the special case is bootstrapping the system.

The LKM is required to be a setuid program. It is strongly recommended that the uid used be that of a non-login user. In addition, a group must be created for use by the LKM. Users who are members of the LKM group are considered privileged users. The LKM process must be setuid to a privileged user.

A host's system administrator adds the users who are to be privileged users to the LKM group. One of these privileged users must be the first user invoke the TIS/PEM system. Although the system will not recognize the huid of this user, it will note that this user is privileged. The first thing this user is required to do is to configure the system to recognize himself or herself. Following that the user may configure the system to recognize other

users.

In addition to distinguishing between privileged and non-privileged users, at compile time the system may be configured to require users to identify themselves by entering a PEM specific password prior to the execution of a set of functions. Finally, at compile time or at run time, the system may be configured to require users to contact a privileged user when attempting any function except a query function.

5.2 Passwords

The TIS/PEM system may be configured to allow users to make use of a password. This password is specific to the TIS/PEM system, i.e., there is no relationship between it and a users UNIX login passwords.

There are two purposes for the TIS/PEM password. If enabled, the system would use the password to protect a user's private key. In this way, not even a system administrator would have access to the private key. In addition, an organization may require the password to be entered when the user attempts to invoke a certain set of commands (as determined at compile time). This provides protection from persons who attempt to use the idle terminals or workstations of PEM users.

There are five levels of password support provided:

1. No password permitted — At compile time, the system may be configured to disallow the use of passwords.
2. User discretionary password permitted — At compile time, the system may be configured to allow a user to set a password, if the user so chooses.
3. Login period — By default, if a user sets a password, the user will be required to enter that password every time it is needed. Alternatively, the system may be configured to allow a user to login to TIS/PEM for an extended period of time specified by the user.
4. Password required — As of this writing, the current version of TIS/PEM does not support mandatory passwords. However, inclusion of this feature is a natural enhancement.

It is important to note that use of the login period leaves a user vulnerable to persons with system administrator privileges. This is because the LKM stores the state of a user's login in a file. Ordinarily this file would only be accessible to the LKM process. However, if this file is accessed by any one else, the user's password, and thus his or her private key, will be compromised.

5.3 Registration Management

Registering a user is a two step process. First, the user's huid must be known to the system. Second, a user must create, or have created, a certificate to embody the user's public key.

Although a user may be authorized to use the system, it may be necessary for an organization to control the creation and deletion of registration. For example, there may be fees associated with the creation of a registration and there is the need to record and distribute the deletion of a registration.

As of this writing, if a user is permitted to perform non-query related functions (see Section 5.1), then a user is permitted to perform all registration related functions. The following list represents our current view of how to separate the various registration functions.

1. Creation of a Registration — Once a user's huid is known to the system, allowing a user to create a registration is a convenience for the privileged users. The alternative is to require each user who needs to be registered to contact a privileged user.
2. Deletion of a Registration — An organization that issues certificates may also have a requirement to record and distribute the existence of deleted certificates. Controlling who may delete registration is an important component of this responsibility.
3. Importing a Registration — An organization that issues certificates or provides access to the TIS/PEM system may need to control the origin of the public keys embodied in a certificate or the origin of certificates in general.
4. Exporting a Registration — If a certificate embodies an affiliation with the organization that issued it, it may be necessary for the organization to control whether or not the registration may be used elsewhere.
5. Multiple Registrations — As of this writing, the TIS/PEM user interfaces do not support multiple registrations per user, although the LKM does. Some users may need more than one registration, since they may have more than one role in an organization. Supporting this feature is a natural enhancement of the system.

5.4 Database Access Management

The database mechanism used to store private keys and certificates is the UNIX file system.

Although applications may make direct use of the database files, the permissions on the files would prevent them from unauthorized access. Only the LKM is permitted to either read or write a private key file. Certificates in the database are intended to be public information and would be directly readable. However, when certificates are requested from the LKM, it is careful to validate the certificate, that is, check its integrity, signature, and validity period. Only the LKM is permitted to write a public key certificate file.

A critical component of protecting access to the database is the checking of the permissions set on the files and directories associated with the database. For this reason, the desired permissions are required to be configured into the LKM at compile time. In this way, although neither UNIX nor the LKM can control permission changes, the LKM will refuse to operate if it does not find the permissions it is expecting. In this way, changes to the database will be immediately obvious.

The initial setting of the permissions allows an organization to choose one of two broad access policies. The permissions can be set to prevent access by all users except via the LKM process, or the permissions can be set to allow all users access to all certificates but still prevent access to the private keys.

6 Conclusions

Initially, PEM will improve the security of electronic mail. Potential uses of PEM include both intra- and inter-organization communications, software distributions, security incident reports, business documents, and many others. The TIS/PEM implementation is specifically designed to provide application programmer interfaces that facilitate the integration of the PEM technology into a variety of Internet mail programs as well as a variety of other distributed text-based applications.

Currently, the Internet Engineering Task Force (IETF) PEM Working Group is focusing attention on the integration of the core PEM functions into the recently published specifications for MIME (Multipurpose Internet Mail Extensions) [3]. MIME specifies extensible mechanisms for specifying and describing the format of Internet message bodies. These mechanisms redefine the format of RFC 822 message bodies to allow for multi-part text and non-text message bodies. A planned document will specify how to integrate the PEM functionality with the MIME mechanisms.

Perhaps most importantly, the cryptographic technology employed by PEM for text messages is ideally suited for distributed applications in general. Widespread use of this technology in applications other than electronic mail is facilitated with the development of PEM's supporting certificate-based infrastructure. It is expected that the introduction of PEM to the Internet will precipitate the further introduction of much needed security services into a wide variety of Internet applications.

Acknowledgements

Lots of people at Trusted Information Systems have contributed to the design and development of the TIS/PEM system. Some of the developers deserving special thanks include Bryan Buck, Paul Clark, Pam Cochrane, Steve Crocker, Mark Feldman, and Sheila Haghighat.

We would also like to express our thanks to the folks at RSA Data Security who have provided the RSA cryptographic software for the version of the TIS/PEM system which will be openly available on the Internet.

References

[1] ANSI X3.92-1981. *Data Encryption Algorithm.* American National Standards Institute, December 30, 1980.

[2] David M. Balenson. Privacy Enhancement for Internet Electronic Mail: Part III – Algorithms, Modes, and Identifiers. Internet Draft, Trusted Information Systems. RFC in progress. Will obsolete RFC 1115.

[3] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extentsions). RFC 1341, Bellcore and Innosoft, June 1992.

[4] David H. Crocker. Standard for the Format of ARPA Internet Text Messages. RFC 822, University of Delaware, August 1982.

[5] FIPS Publication 46-1. *Data Encryption Standard.* National Institute of Standards and Technology, U. S. Department of Commerce, Washington, D.C. Federal Information Processing Standard (FIPS); Supersedes FIPS Publication 46, January 15, 1977; Reaffirmed January 22, 1988.

[6] Burton S. Kaliski. Privacy Enhancement for Internet Electronic Mail: Part IV – Key Certification and Related Services. Internet Draft, RSA Data Security, Incorporated. RFC in progress.

[7] Burton S. Kaliski. MD2 Message-Digest Algorithm. RFC 1319, RSA Data Security, Incorporated, April 1992. Updates RFC 1115.

[8] Steve Kent. Privacy Enhancement for Internet Electronic Mail: Part II – Certificate-Based Key Management. Internet Draft, BBN Communications. RFC in progress. Will obsolete RFC 1114.

[9] Steve Kent and John Linn. Privacy Enhancement for Internet Electronic Mail: Part II – Certificate-Based Key Management. RFC 1114, BBN Communications, August 1989.

[10] John Linn. Privacy Enhancement for Internet Electronic Mail: Part I – Message Encipherment and Authentication Procedures. Internet Draft, Digital Equipment Corporation. RFC in progress. Will obsolete RFC 1113.

[11] John Linn. Privacy Enhancement for Internet Electronic Mail: Part I – Message Encipherment and Authentication Procedures. RFC 989, BBN Communications, February 1987. Obsoleted by RFC 1040.

[12] John Linn. Privacy Enhancement for Internet Electronic Mail: Part I – Message Encipherment and Authentication Procedures. RFC 1040, BBN Communications, January 1988. Obsoletes RFC 989. Obsoleted by RFC 1113.

[13] John Linn. Privacy Enhancement for Internet Electronic Mail: Part I – Message Encipherment and Authentication Procedures. RFC 1113, Digital Equipment Corporation, August 1989. Obsoletes RFC 1040.

[14] John Linn. Privacy Enhancement for Internet Electronic Mail: Part III – Algorithms, Modes, and Identifiers. RFC 1115, Digital Equipment Corporation, August 1989.

[15] Jonathan B. Postel. Simple Mail Transfer Protocol. RFC 821, Information Sciences Institute, University of Southern California, August 1982.

[16] Ronald L. Rivest. MD5 Message-Digest Algorithm. RFC 1320, RSA Data Security, Incorporated, April 1992.

[17] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[18] The Directory - Authentication Framework. Recommendation X.509, The International Telegraph and Telephone Consultative Committee, November 1988. Developed in collaboration, and technically aligned, with ISO 9594-8.