SECURE E-MAIL SYSTEM USING S/MIME AND IB-PKC

S. T. Faraj
College of IT, Nahrain University, Al-Jaderiya, Baghdad, Iraq
M. T. Ibrahem
Dept. of Computer Engineering, University of Baghdad, Al-Jaderiya, Baghdad, Iraq
Correspondence Email: sufyantaih@yahoo.com

ABSTRACT

Although e-mail security solutions have been introduced for more than two decades, most of the e-mail messages are sent nowadays without being secured by any of these techniques. This is due to the complexity of using these secure e-mail systems and protocols. The complexity mainly arises from the difficulty associated with managing certificates and public keys. The main objective of this study was to find a solution that can make secure e-mail systems easier to use while maintaining the same level of security. This paper proposes a secure e-mail system that is based on the S/MIME standard where the public key and signature algorithms have been replaced by their Identity-Based Cryptography analogue algorithms.

Using Identity-Based Cryptography has eliminated the need for digital certificates, and provided a solution to the usability problem present in the existing secure e-mail systems. Users can determine the public key of the recipient without having to contact any trusted third party, and can start encrypting or verifying messages as long as they have the public system parameters that can be publicly available. Users need to contact the Private Key Generator (PKG) only once in order to retrieve their private key before being able to decrypt or sign messages.

Keywords: E-Mail, S/MIME, Identity-Based Cryptography, IBE, IBS, Security, Usability.

1. INTRODUCTION

E-mail is one of the most popular activities on the Internet. The ease of communicating over e-mail makes it the optimum communication medium for many people and businesses. In order for e-mail to be used in important communications, secure e-mail systems were developed. E-mail Security solutions such as PEM [1], PGP [2] and S/MIME [3] have been in place for more than a decade, and have been implemented by most e-mail clients. However, most e-mail messages are sent without being secured by any of the available solutions. This is mainly because of the complexity of using these secure e-mail systems and protocols.

The PGP key certification model is of a user-centric type that allows individuals to create their own public/private key pairs and then optionally have those public keys certified by one or more individuals, but this comes at the cost of added responsibility on the part of end-user. On the other hand, S/MIME was constructed with an eye toward integration with the Public Key Infrastructure (PKI). As such, S/MIME allows the use of certificates for both signing and encrypting e-mail, along with the integration with Certificate Revocation Lists (CRL) and cryptographically signed return receipts. Current PKI have several aspects that attracted criticism and controversy, some of which are: difficulty in retrieving keys and certificates, certificate processing complexity, costly certificates, and naming semantics problem [4]. Several solutions have been made in order to reduce this complexity; some of them were in the form of commercial security appliances that manage the key processing for the users, while the others were academic solutions [5, 6].

In Identity-Based Public Key Cryptography (IB-PKC), the public keys are constructed from users' publicly available identities that are uniquely associated with them (like their e-mail addresses). The motivation for such a scheme was to simplify key management and eliminate the use of certificates. Since the introduction of the first successful Identity-Based Encryption (IBE) scheme, replacing the current PKI services with their equivalent IB-PKC techniques has become a very popular research area. This ranges from making some kind of combination between the PKI services and the IB-PKC algorithms to totally replacing some PKI services with their IB-PKC alternatives. Several studies have been made on the subject of securing e-mails with the IB-PKC principles (for example see [7, 8, 9, 10, 11]). However, most of these approaches can only be deployed with client or server updates. They were not concerned with specifying a standard secure e-mail message format as they were mainly describing methods for deploying an Identity-Based PKI.

In order for two e-mail users to communicate securely, it is necessary for them to use programs that are compatible with each other. Hence secure e-mail systems that utilize IB-PKC algorithms needs to make use of existing e-mail security standards in order to be widely adopted and make secure messaging easier to use. In this paper, a secure e-mail system is proposed that is based on S/MIME standard where the public key and signature algorithms have been replaced by their IB-PKC analogue algorithms. We believe that our approach would result in a better usability of secure e-mail systems.

The rest of this paper is organized as follows: Section 2 briefly introduces S/MIME and Identity-Based Cryptography. Some important design considerations are outlined in Section 3. The specifications of messages used in the proposed system are summarized in Section 4, while the overall system architecture is presented in Section 5. Then, some notes on the implementation and performance are given in Section 6. Also, a possible important extension to existing system implementation is highlighted in Section 7.

2. PRELIMINARIES

This section briefly presents the basic building blocks of our approach to develop a better usability secure e-mail system. These are: S/MIME and IB-PKC.

2.1 S/MIME

MIME (Multipurpose Internet Mail Extensions) extends the format of Internet Mail to allow non-ASCII textual messages, non-textual messages, multi-part message bodies, and non-ASCII information in message headers. Messages in MIME format can contain files of different types. The file type is described in a “content-type” header that is used by the recipient e-mail program to determine how to handle that file. MIME also defines a set of content transfer encodings which can be used to represent 8-bit binary data using characters from the 7-bit ASCII character set.

S/MIME (Secure/Multipurpose Internet Mail Extensions) provides a consistent way to send and receive secure MIME data. Based on the MIME standard, S/MIME provides the following cryptographic security services for electronic messaging applications: data confidentiality (using encryption), authentication, message integrity and non-repudiation of origin (using digital signatures). S/MIME messages are made up of MIME bodies and Cryptographic Message Syntax (CMS) objects. There are several CMS content types. Of these, only four content types are currently used by S/MIME:

- *Data:* Sending agents must use the "id-data" content type identifier to identify the "inner" MIME message content.
- *SignedData:* This content type is used to apply a digital signature to a message or, in a degenerate case where there is no signature information, to convey certificates.
- *EnvelopedData:* It is used to apply data confidentiality to a message. A sender needs to have access to a public key for each intended message recipient to use this service.
- *CompressedData:* This content type is used to apply data compression to a message. It is only used to reduce message size.

2.2 Identity-Based Cryptography

What makes IB-PKC differs from the ordinary PKC is that the public key can be an ordinary string. For example, if Alice wants to encrypt a message to Bob at “bob@foo.com”, then she simply encrypts the message using “bob@foo.com” as the public key. Thus, any user in the system can send encrypted e-mail messages to anyone, even if the recipient hasn't yet registered and created his/her own private key. Once a recipient receives the encrypted message, that recipient contacts the Private Key Generator (PKG) in order to request his/her private key that is needed for decryption, and it's the only time in which this user has to contact any Trusted Third Party (TTP) until he/she needs to update the private key later.

The primary advantage of IB-PKC is that a system participant can make a public key without having to contact any TTP. Moreover, it can be used for managing user credentials, delegation of decryption keys, or for implementing short lived public keys. It has also been used to build forward-secure encryption schemes.

While efficient solutions for Identity-Based Signature (IBS) schemes were quickly found, most IBE schemes proposed since 1984 were unsatisfactory because they were too computationally intensive, they required tamper resistant hardware, or they were not secure if users colluded. The first efficient and secure IBE scheme did not appear until 2001 when Boneh and Franklin [12] presented their IBE scheme. The Boneh-Franklin IBE scheme (BF-IBE) is based on bilinear maps between groups. The BF-IBE scheme has chosen ciphertext security in the random oracle model assuming a variant of the computational Diffie-Hellman problem. Moreover, its performance is comparable to the performance of ElGamal encryption.

After that, several other identiy-based schemes were proposed based on the BF-IBE scheme. For example, the Cha-Cheon IBS scheme (CC-IBS) [13] shared the same system parameters with the BF-IBE scheme. Combining these two schemes yields a complete solution for of an Identity-Based Public Key system. In an IBE scheme there are four algorithms:

- *Setup*: takes as input a security parameter and outputs params (system parameters) and the master-key. The system parameters must include the description of the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. The system parameters would be publicly known while the master-key is known only to the Private Key Generator (PKG).
- *Extract*: takes as input the system parameters (params), the master-key and an arbitrary string ID $\in \{0,1\}^*$ and outputs the private key d_{ID} corresponding to the public key ID.
- *Encrypt*: takes as input the system parameters (params), a public key ID and a plaintext $\mathrm{M} \in \mathcal{M}$ and outputs the corresponding ciphertext.
- *Decrypt*: takes as input the system parameters (params), a private key d_{ID} and a ciphertext $\mathrm{C} \in \mathcal{C}$ and outputs the corresponding plaintext.

IBS schemes also have four algorithms. However, *Encrypt* and *Decrypt* are replaced by the algorithms *Sign* and *Verify*. The algorithm *Setup* is run by the PKG. The PKG also runs the algorithm *Extract* at the request of a user who wishes to obtain the private key corresponding to some string. The user should prove to the PKG that he/she is the legitimate owner of this string. The algorithms *Encrypt*, *Decrypt*, *Sign*, and *Verify* are run by the users to encrypt, decrypt, sign, and verify messages.

3. SOME IMPORTANT DESIGN CONSIDERATIONS

The proposed secure e-mail system should securely transmit e-mail messages, be easy to use, make use of the existing secure e-mail standards, and it should be applied without making significant changes to the structure of the network. In order to achieve the previous goals, some decisions had to be made before designing the system:

- The first question to be answered is whether to apply security to both the e-mail client and server, or just one of them. Any change in the e-mail servers is not recommended, since this implies that all the e-mail servers around the world should be updated to implement the new changes. Hence, this design would apply security to e-mail clients only, and this will allow any organization to apply this system without having to modify the underlying network architecture.

- Distributing private keys to the users of the system is a main concern when designing any secure e-mail system. There are more than options to choose from here. The technique adopted in this work is based on the E-mail Based Identification and Authentication (EBIA) approach [14]. It involves developing an e-mail-based application that would be able to receive private key request e-mail messages, and respond to them. One important reason for this choice is that the PKG design would be similar to the secure e-mail client design, and would simplify the process of generating private keys.
- The cryptographic algorithms to be used should have the best security, efficiency, and usability characteristics compared to other available candidates. Thus, the AES would be used for the conventional encryption, the Boneh-Franklin IBE scheme for the public-key algorithm, the Cha-Cheon IBS scheme for the Signature algorithm, and the SHA-1 for the secure hash function. The Boneh-Franklin IBE scheme (BF-IBE) and the Cha-Cheon IBS scheme (CC-IBS) have been chosen together because they have a similar initial steps. The public system parameters of both algorithms can be combined to compose general public system parameters.
- The only needed information for any participant to start using the system is the public system parameters. These parameters can be stored in a file that would be publicly available. They are needed for using the Identity-Based Cryptography algorithms. These parameters are the result of combining BF-IBE scheme and CC-IBS scheme public system parameters (where the master key is s, and it is chosen randomly provided that $s \in Z_p^*$):
    1. G_1, G_2, $\hat{e}$, p: G_1 and G_2 are cyclic groups of prime order p together with a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ corresponding to this security parameter.
    2. P: a random generator $P \in G_1$.
    3. P_{pub}: $P_{pub} = s*P$.
    4. Cryptographic hash functions.

4. SECURE MESSAGE SPECIFICATIONS

Since S/MIME is the basis for this system, the message format would follow the S/MIME message specifications [15] and the CMS standard [16] with slight modifications to some fields in order to add the functionality of the Identity-Based Cryptography. Hence, some modifications had been applied to the CMS EnvelopedData and SignedData types. However, the CompressedData doesn't need to be modified. The Data content type also doesn't need to be modified since it only exists to contain the message content (like MIME body part) that is going to be processed by S/MIME. Indeed, it is important to notice that:

- The CMS Version of the current S/MIME is 3.1; while in our design it is considered 5; in order to differentiate it from the existing S/MIME standard that depends on a different set of public-key encryption algorithms.
- The Digest algorithms and the Message encryption algorithm are exactly as specified in the S/MIME specification. However, the Digital signature algorithms and the Session key encryption algorithms are replaced by IB-PKC algorithms. The Cha-Cheon IBS Scheme is going to be the Digital signature algorithm, and the Boneh-Franklin IBE Scheme is going to be the Session key encryption algorithm.
- All the optional fields in the S/MIME message specification will be discarded in our first implementation; as they are mostly related to Digital Certificates, and they won't be used in our system.
- There can be more than one recipient for an encrypted message, and more than one signer for a signed message according to the S/MIME Specifications. However, in order to

simplify the explanation of the system; only one sender and one receiver are assumed in the following description of the SignedData and EnvelopedData types.

- Sending agent is software that creates S/MIME CMS objects, MIME body parts that contain CMS objects, or both. Receiving agent is software that interprets and processes S/MIME CMS objects, MIME body parts that contain CMS objects, or both. S/MIME agents are user software that is a receiving agent, a sending agent, or both.

5. SYSTEM ARCHITECTURE

The proposed system is composed of three parts: the Secure E-mail Client, the Private Key Generator (PKG), and an E-mail Server as shown in Figure 1. The e-mail server is represented in dotted lines since it is not affected by our system approach, and any existing e-mail server would work well when integrated with system. Thus, there is no need to re-design the e-mail server.

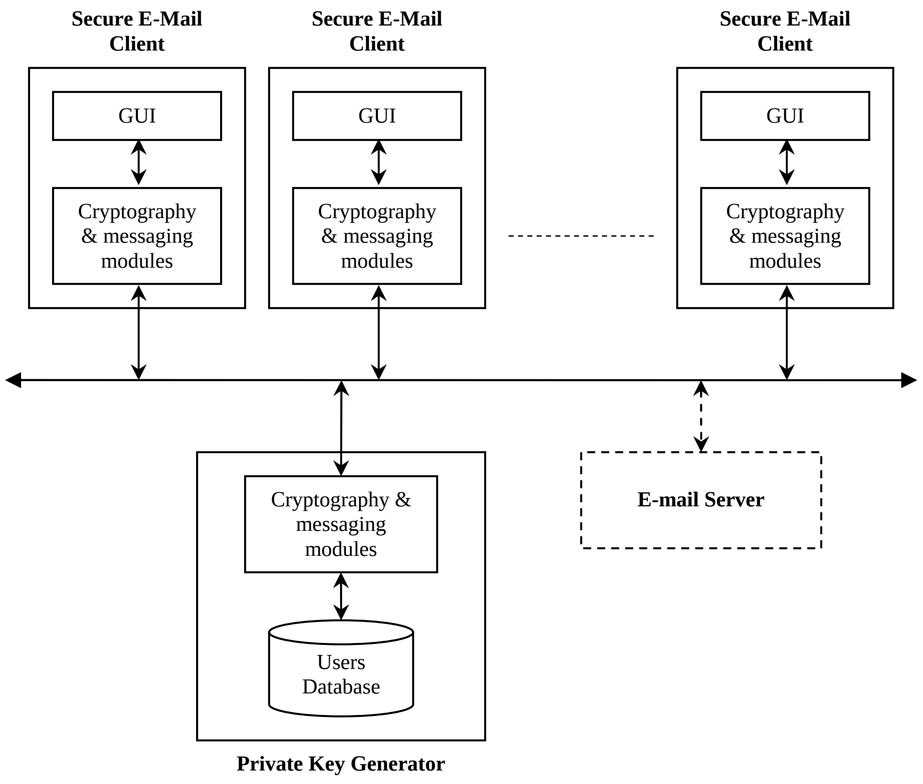
Secure E-Mail Client
Secure E-Mail Client
Secure E-Mail Client
GUI
GUI
GUI
Cryptography & messaging modules
Cryptography & messaging modules
Cryptography & messaging modules
Cryptography & messaging modules
Users Database
Private Key Generator
E-mail Server

Figure 1: Schematic of system architecture.

5.1 Secure E-mail Client

Secure E-mail clients are mainly used for viewing incoming e-mail messages, composing outgoing e-mail messages, managing messages' folders, and applying and managing e-mail security services. The modules needed by this application can be identified as follows (see Figure 2):

- E-mail message module that should provide the ability to create and parse e-mail messages according to the Internet message format and the MIME.
- S/MIME module should contain a set of functions that provide ability to encrypt and/or sign MIME entities, and the ability to decrypt and/or verify MIME entities. It has to comply with the modifications to the S/MIME as mentioned in section 4.

- Cryptography module should provide a set of functions that would be required to carry out the cryptographic operations needed by the secure message specification.
- SMTP module should contain a set of functions responsible for sending e-mail messages using the SMTP protocol.
- POP3 module should contain a set of functions responsible for receiving e-mail messages using the POP3 protocol.
- Database module that should provide all the necessary data structures and the functions needed to interact with them. This can include an options' file structure, a folders' indexing structures (to enhance the performance of the application by reducing the time needed to search for the e-mail message files and scan them each time a folder is selected), and a contacts' information structure.
- Private Key management module that should provide a set of functions responsible for requesting the private key corresponding to the user's public key.
- An easy-to-use graphical user interface (GUI) that would allow the user to easily compose and view messages, apply security services to them, send and receive messages, manage the messages' folders, and manage all the other necessary operations needed by the application.

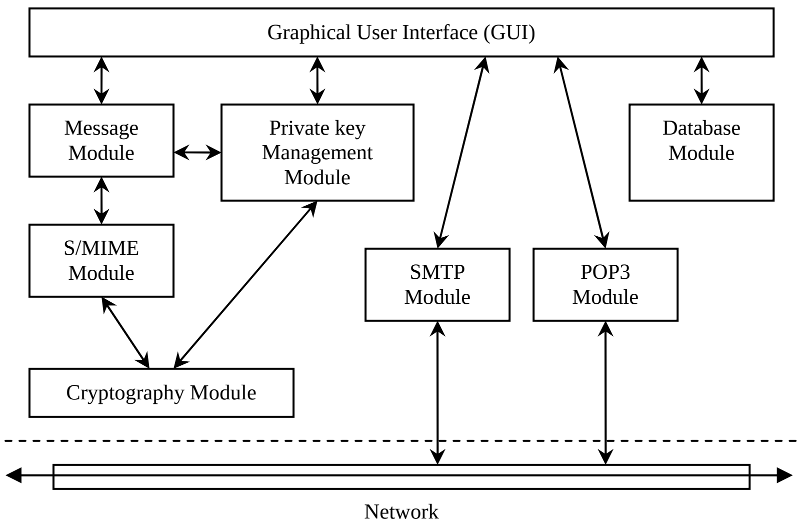

Figure 2: Schematic of the secure e-mail client architecture.

5.2 Private Key Generator (PKG)

PKG is the part of the system representing the trusted authority that would perform the first two steps needed by any system that uses the Identity-Based Cryptography, setup and extract. It first runs the setup algorithm to generate the public system parameters, and the master key, and it does that only once. Then the PKG runs the extract algorithm for every request to generate the private key out of the public key (e-mail) supplied. The extraction process is carried out according to the protocol of Subsection 5.3, where a request would be received from the e-mail client using a special e-mail message format. Then the PKG generates the private key, and send it back to the e-mail client. To accomplish these tasks, the modules needed by the PKG can be identified as follows:

- Cryptography module should provide a set of functions that would be required to run the setup and extract algorithm according to the IBE Scheme, and to carry out the cryptographic operations needed by the secure message specification.

- E-mail message module that should provide the ability to create and parse e-mail messages according to the Internet message format and the MIME.
- S/MIME module should contain a set of functions that provide the ability to encrypt and/or sign MIME entities, and the ability to decrypt and/or verify MIME entities.
- SMTP module should contain a set of functions responsible for sending an e-mail message using the SMTP protocol. POP3 module should contain a set of functions responsible for receiving an e-mail message using the POP3 protocol.
- Message Management module should execute synchronously with the rest of the application. It checks the outbox folder periodically to see whether there are messages to be sent. It also checks the e-mail server for messages to receive them and store them in the inbox folder.
- Database module that should provide all the necessary data structures and the functions needed to interact with them. This can include an options' file structure, details of the rejected requests, a list of the public keys that has their private keys extracted, and a list of the allowed public keys (e-mail addresses) and domain names.
- Private Key extraction module should contain a set of function that would be required to check for any new request, check the e-mail address to see if it's in the allowed list, extract the private key, send it back to client that requested it, and record all the necessary information in the data structures.
- An easy-to-use graphical user interface (GUI) that would allow the user to easily configure the PKG and view the rejected requests, the executed request, and the application's log file.

5.3 Private Key Distribution Method

This subsection explains the method that the e-mail client can use to extract the private key associated with its public key (e-mail address). It's an e-mail dependent method that is based on the E-mail Based Identification and Authentication (EBIA) method. EBIA uses an e-mail address as a universal identifier and the ability to receive e-mail at that address as a kind of authenticator. Mailing lists and forums subscription confirmations, Web password reminders, and e-commerce notifications all use this method. In our system, private keys, which are very sensitive data, are distributed using EBIA. This is why EBIA needs to be modified by the use of fine cryptographic principles in order to maintain its security.

This key distribution method is carried out between the Private Key management module in the e-mail client and the Private Key extraction module in the PKG, as shown in Figure 3. Before being able to extract its private key, the e-mail client should obtain the public system parameters, a common information file that should be provided by the PKG. The user should also provide the e-mail address of the PKG. In general, the steps needed to accomplish this task are:

1. The secure e-mail client sends a request to the PKG.
2. The PKG responds to the request.
3. The Secure e-mail client receives the private key.

6. IMPLEMENTATION AND PERFORMANCE

The language used in programming this system is C/C++, and the compiler used is the Microsoft Visual C++. Two applications were developed: the secure e-mail client and the Private Key Generator (PKG). Some modules were needed by both application, and they are the cryptography, S/MIME, E-mail Message, SMTP, and POP3 modules. On the other hand, some modules were specific to each one of the two applications. Some open source libraries were used like the SFL library [17] and the MIRACL library [18] in order to speed up the development time, to avoid increasing the number of bugs in the applications, and to ensure that the best performance has been achieved since these libraries have been thoroughly tested by many developers around the world.

The system was tested on a PC with Intel Celeron 2GHz processor, 256 MB of RAM, 100Mbps Fast Ethernet PCI adapter, and has Windows XP Professional installed. The secure e-mail client uses an average of 9 MB of RAM, while the PKG uses an average of 5.6 MB of RAM. A code profiler has been used to measure the time needed to create encrypted or signed e-mail messages. These timing results are shown in Figure 4 and Figure 5 for encrypted and signed messages respectively. It is obvious that encrypting messages requires a higher time than signing them given the same message size.

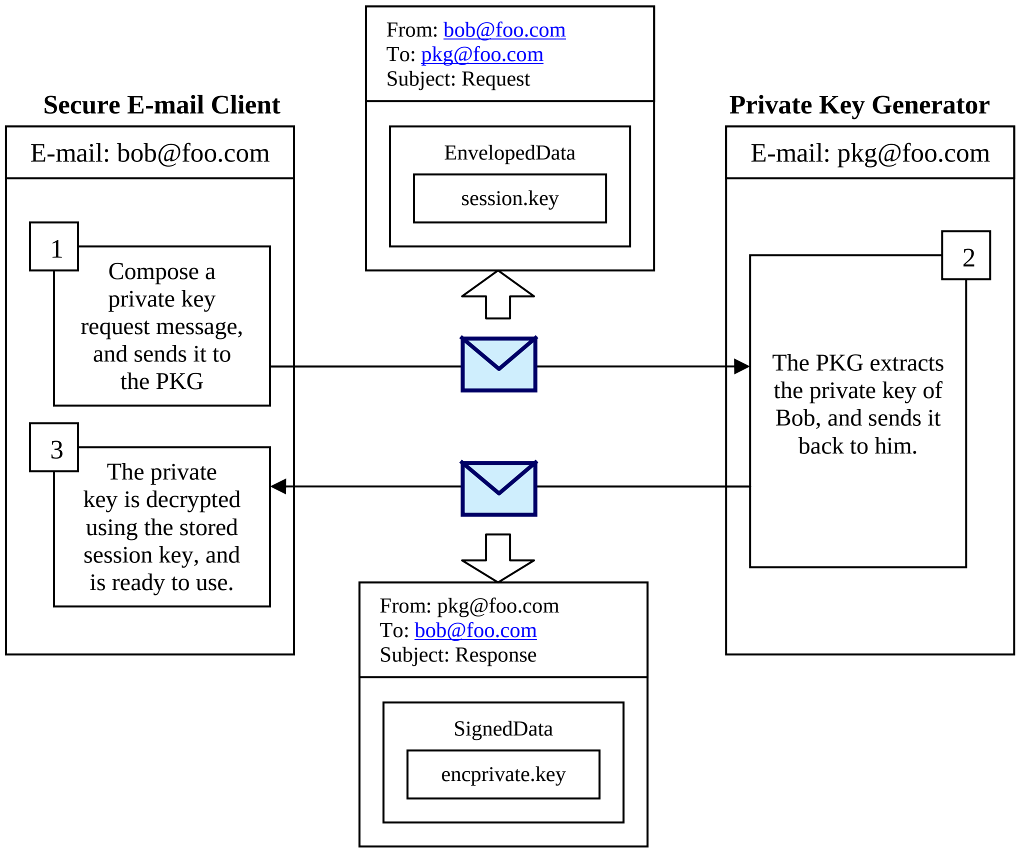

Figure 3: Schematic of the private key distribution method.

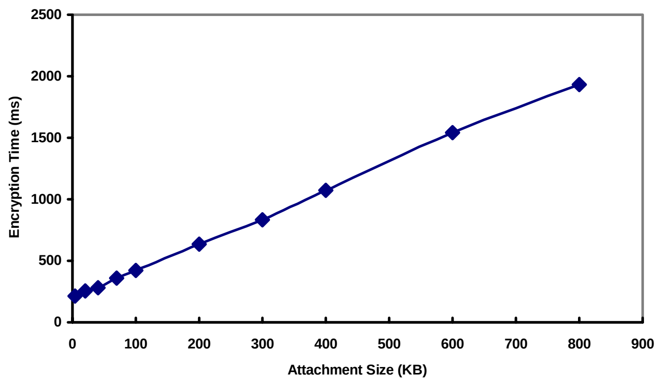

Figure 4: Encryption Time Graph.

7. EXTENSION TO IDENTITY-BASED PKI

Using a single PKG from which all users request their private keys may not be practical in some situations. Each institution may want to have its own PKG and manage its users' keys. Moreover, a PKG's secret key must not exist on a single machine since it can be used to decrypt all the messages of this PKG's users. The Lightweight PKI, described in [10] and [11], can be used to solve these problems. In this Identity-Based PKI, each domain has its own PKG that extracts private keys for its users. The Master Public Key (*MPK*) of a domain is distributed via the DNS, as a TXT record associated with the hostname of the domain's Mail Exchange (MX) record.

In fact, a domain should maintain several PKGs that independently generate master key shares (MPK_i, MSK_i). The Master Public Key shares are combined into a single *MPK* that is distributed via the DNS. Each server with Master Secret Key MSK_i individually sends Alice secret key share $SK_{Alice,i}$. Alice then can combine all shares into a single private key SK_{Alice}. EBIA is used to distribute these secret key shares [11].

The integration of our system with the Identity-Based PKI, described above, can be carried out without modifications to our system. In [10], EBIA is used to distribute secret key shares, while our system uses a modified version of EBIA to distribute private keys. Our modified EBIA is more secure and can be used in the distribution of the secret key shares. A user can send request e-mail messages to all the PKG servers of a single domain. Each one of these PKGs receives a different session key in the request message, generates a secret key share, encrypts it with the received session key, and sends it back the user. The user then decrypts these secret key shares using the stored session keys. After that he/she can combine these secret key shares to form his/her private key. Thus, all the secret key shares would never be sent without encryption.

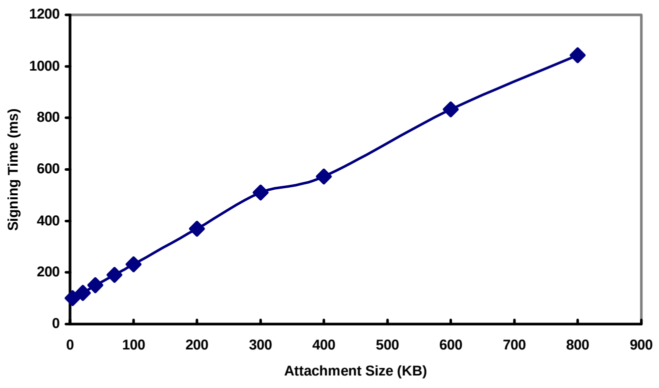

Figure 5: Signing Time Graph.

CONCLUSIONS

Our system provides a solution to the usability problem present in the existing e-mail security solutions. Anyone can start using the system after getting a copy of the common system parameters that can be publicly provided. When a user wants to get his/her private key, the only thing needed is the e-mail address of the PKG. The system is compatible with the S/MIME protocol. In fact, it’s based on S/MIME to benefit from the structure of S/MIME that has gone through extensive testing and improvement for several years now. This also decreases the time and effort needed to add this system’s functionality in the existing e-mail applications that implement S/MIME. The changes made by this system need to be applied on the e-mail clients only, while the e-mail servers stay intact. These changes to the e-mail clients don’t need to be integrated into them. They can be provided in the form of Add-Ins to these applications.

REFERENCES

[1] Linn, J., "*RFC 989: Privacy Enhancement for Internet Electronic Mail, Part I: Message Encipherment and Authentication Procedures*", Internet Engineering Task Force (IETF), February 1987.

[2] Atkins, D., Stallings, W. and Zimmermann, P., "*RFC 1991: PGP Message Exchange Formats*", Internet Engineering Task Force (IETF), August 1996.

[3] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L. and Repka, L., "*RFC 2311: S/MIME Version 2 Message Specification*", Internet Engineering Task Force (IETF), March 1998.

[4] Linn, J., and Branchaud, M., "*An Examination of Asserted PKI Issues and Proposed Alternatives*", In Proceedings of the 3rd Annual PKI R&D Workshop, pages 34-47, March 2004.

[5] Whitten, A. and Tygar, J., "*Safe staging for computer security*", Proceedings of the 2003 Workshop on Human-Computer Interaction and Security Systems**,** April 2003. "http://www.andrewpatrick.ca/CHI2003/HCISEC/".

[6] Garfinkel, S., "*Enabling e-mail Confidentiality through the use of Opportunistic Encryption*", The 2003 National Conference on Digital Government Research, National Science Foundation, pages 173-176, 2003.

[7] Baldwin, M., Identity Based Encryption from the Tate Pairing to Secure E-mail Communications. Master of Engineering Thesis, University of Bristol, May 2002.

[8] Ding, X. and Tsudic, G., "*Simple Identity-Based Cryptography with Mediated RSA*", Cryptographer's Track RSA Conference, 2003.

[9] Adida, B., Chau, D., Hohenberger, S. and Rivest, R., "*Lightweight Signatures for Email*", a preliminary version presented in the DIMACS Workshop on Theft in E-Commerce, April 2005.

[10] Adida, B., Chau, D., Hohenberger, S. and Rivest, R., "*Lightweight Email Signatures*", February 2006. Available at " http://theory.lcs.mit.edu/~rivest/publications.html".

[11] Adida, B., Hohenberger, S. and Rivest, R., "*Lightweight Encryption for Email*", In Proceedings of Usenix's Symposium on Reducing Unwanted Traffic on the Internet, pages 93-99, July 2005.

[12] Boneh, D. and Franklin, M., "*Identity Based Encryption from the Weil Pairing*", Proceedings of Crypto 2001, Lecture Notes in Computer Science (LNCS) 2139, pp 213 - 229, Springer-Verlag, 2001.

[13] Cha, J. and Cheon, J., "*An Identity-Based Signature from Gap Diffie-Hellman Groups*", Practice and Theory in Public Key Cryptography-PKC 2003. Also Cryptology ePrint Archive 2002/018, 2002.

[14] Garfinkel, S., "*Email-based identification and authentication: An alternative to PKI*?", Security & Privacy Magazine, 1:20–26, Nov. - Dec. 2003.

[15] Ramsdell, B., "*RFC 3851: Secure/Multipurpose Internet Message Extensions (S/MIME) Version 3.1 Message Specification*", Internet Engineering Task Force (IETF), July 2004.

[16] Housley, R., "*RFC 3852: Cryptographic Message Syntax (CMS)*", Internet Engineering Task Force (IETF), July 2004.

[17] SFL, S/MIME Freeware Library, "http://www.digitalnetgov.com/hot/ sfl_home.htm".

[18] MIRACL, Multiprecision Integer and Rational Arithmetic C/C++ Library, "http://indigo.ie/~mscott/".