Practical Implementation of a Secure Email System Using Certificateless Cryptography and Domain Name System

Suresh Kumar Balakrishnan[1] and V. P. Jagathy Raj[2]
(Corresponding author: Suresh Kumar Balakrishnan)
School of Computer and Information Sciences, Indira Gandhi National Open University (IGNOU)[1]
New Delhi 110068, India
(Email: suresh@sctimst.ac.in)
School of Management Studies, Cochin University of Science and Technology[2]
Kochi, India
(Email: jagathy@cusat.ac.in)
(Received Oct. 25, 2014; revised and accepted Jan. 16 & June 21, 2015)

Abstract

Email is currently the most widely used communication system in daily life. To improve security and efficiency, most email systems adopt Public Key Infrastructure (PKI) as the mechanism to implement security, but PKI based systems suffer from expensive certificate management and problems in scalability. Identity Based Cryptography (IBC) is another method, but it has the inherent drawback of Key Escrow. This paper proposes an implementation of a practical, secure email system based on certificateless cryptography, which uses Domain Name System (DNS) as the infrastructure for public key exchange and a secure key token/fingerprint authentication system for user authentication. The message payload is encrypted by a per-email symmetric key generated from a secret value, the public and private keys of both the sender and the receiver. The proposed mailing system is secure against standard security model.

Keywords: Certificateless cryptography, domain name system, identity based cryptography, multi-factor authentication, public key infrastructure, secure email system

1 Introduction

The main reason for using email is probably the convenience and speed with which it can be transmitted, irrespective of geographical distances. Similar to a postcard, an email has open access to the systems on its path. If anyone wants to intercept, copy or alter information, they can easily do so. Confidential information, such as bank statements, trade secrets, and even national secret information, is being exchanged through emails. Therefore, the contents of emails are more important and valuable than ever, and their security has raised many concerns. The main reason for not using encryption in email communications is that current email encryption solutions require expensive operations and hard key management. Therefore, research on simple, highly secure and efficient email systems are in great need.

Current email systems that use symmetric and asymmetric cryptographic schemes [12] suffer from key management problems. Identity Based Cryptography (IBC) [4, 19, 30] systems, which have been proposed to address such key management issues, also suffer from the key escrow problem, which violates the non-repudiation feature that should be offered by security systems.

This paper proposes a practical implementation of a secure email system in an open standard as an alternative technology for eliminating the problems with PKI and identity-based cryptographic mailing systems. This system uses the certificateless public key cryptography scheme by Al-Riyami and Paterson [29], Domain Name System (DNS) [11, 25, 26], for publishing a user's public key details and multi-factor user authentication for secure user authentication with the system.

The rest of the paper is organized as follows. Related works on existing email security systems and an introduction to certificateless cryptography are described in Section 2. Section 3 describes the design of the proposed system. Section 4 describes the implementation of the system. The security features of the paper are described in Section 5. Finally, benchmarking and conclusions are

given in Section 6.

2 Literature Review

2.1 Existing Schemes to Secure Email Systems

The majority of client-based email security systems are based on Identity-Based Cryptography or Public Key Infrastructure (PKI) [8, 9, 21] schemes. The above security functions are implemented by these solutions, of which the most competitive ones are S/MIME [27] and PGP [3, 32, 36]. PGP uses hash functions and public key encryption algorithms, for example, RSA [12, 28] and MD5 [12], to enable encryption for content-protection and digital signature for non-repudiation. RSA public keys are attached as PGP certificates along with the message. However, self-signed PGP certificates are used for most users and form a chain-based credential trust network. This trust mode of PGP is only suitable for small-scale groups and is not suitable for large-scale groups or anonymous user environments. Moreover, it is very difficult to notify other users in the network, if the private key of a PGP user has been compromised. S/MIME employs the PKI framework. Due to the difficulty of certificate management in PKI, S/MIME cannot ignore tedious operations, such as certificate revocation, verification, and so on. In addition, both S/MIME and PGP use RSA for encrypting and signing email contents. This results in lower efficiency compared with Elliptic Curve Cryptography (ECC) [2, 14] with the same level of security. In the IBC scheme, it is difficult to prove the self-identity of the Trust Authority (TA) or the Key Generation Center (KGC) [35]. The scheme also suffers from the problem of key escrow, where a central trusted authority issues a private key to a user. Because a central authority is responsible for private key generation, it is able to work as an authorized user and could maliciously decipher the incoming encrypted text or generate false signatures. Several methods [7, 22, 34] have been proposed to solve the key escrow problem in IBC, and they can be easily classified into two groups based on the private key generation technique: (i) Multiple authority approach and (ii) User chosen secret key information approach. As per our survey, numerous techniques [5, 14, 15, 16, 24, 31] follow the multiple authority approach, while very few techniques [17, 29] are based on the secret key information approach. In the multiple authority approach, the critical task of private key generation is distributed among several authorities, and as a result, no single authority can perform any unauthenticated work. Although these methods successfully solve the key escrow problem, they introduce extra overhead on systems and lack of central control on key issuing policy and are not suited for email security systems. User chosen secret information approaches are either certificate based or certificateless. The certificate based scheme completely overcomes key escrow; however, it loses the advantage of an ID based scheme. The secret key exchange protocol based system is also not suited for email systems because a receiver of the email system may not always be online.

Domain Keys Identified Mail (DKIM) [10] permits users to claim some responsibility for a message by associating it with a domain name that they are authorized to use. This claim is validated through a cryptographic signature and by querying the Signer's domain directly to retrieve the appropriate public key. The approach taken by DKIM differs from previous approaches to message signing such S/MIME and OpenPGP [3] in that:

- The message signature is written as a message header field instead of part of the message body, so that neither human recipients nor existing MUA (Mail User Agent) software are confused by signature-related content appearing in the message body.
- There is no dependency on well-known trusted authority public and private-key pairs.

A new concept called Lightweight Signatures for Email (LES) [1], proposed by Ben Adida, David Chau, Susan Hohenberger, and Ronald L. Rivest, is an extension to DKIM. In LES, individual users authenticate within a domain, without requiring additional user authentication infrastructure. LES allows a user to send emails via any outgoing mail server, not just the official outgoing mail server mandated by DKIM. LES also supports repudiable signatures to protect users' privacy. Both DKIM and LES focus only on email authentication. LES requires a modified email client for authentication.

The Proxy based email system [6, 13, 18, 23] is another scheme that has the key escrow problem.

The scheme described as "An End-to-End Secure Mail System Based on Certificateless Cryptography in the Standard Security Model" [20] based on of Certificateless Public Key Cryptography (CL-PKC) [29] is not suitable for practical implementation with different domains. None of these works are satisfactory. Therefore, efficient email security systems are in great need.

This paper proposes a practical implementation of a secure email system using certificateless cryptography as an alternative technology for eliminating the problems with PKI and IBC based mailing systems.

2.2 Certificateless Public Key Cryptography

The concept of Certificateless Public Key Cryptography (CL-PKC) is introduced by Al-Riyami and Paterson [29] in 2003, to overcome the key escrow problem of identity-based cryptography. In CL-PKC, a trusted third party, called the Key Generation Center (KGC), supplies a user with a partial private key. The user then obtains his/her full private key by combining the partial private key with a secret value that is defined by the user and is unknown to the KGC. Thus, the KGC does not know the user's private keys. Subsequently, the user combines the his/her se-

cret value with the KGC's public parameters to compute his/her public key.

Compared to identity based public key cryptography (IDPKC), the trust assumptions regarding the trusted third party in this scheme are significantly reduced. In CL-PKC, users can generate any number of private-public key pairs for the same partial private key. To guarantee that the KGC does not replace a user's public keys, they proposed a binding technique to bind a user's public key with his/her private key. In their binding scheme, the user first fixes his/her secret value and generates his/her public key and supplies the public key to KGC. Then, the KGC redefines the user identity as the user's identity concatenated with his/her public key. Using this binding scheme, the replacement of a public key of a user in the system by the KGC is equivalent to certificate forgery by a CA in a traditional PKI system.

2.3 Al-Riyami and Paterson Scheme

In the proposed secure mailing system, the CL-PKC concept, as proposed by Al-Riyami and Paterson, is used. The general description of the algorithms introduced by Alriyami and Paterson is provided. These algorithms are Setup, Set-Secret-Value, Partial-Private-Key-Extract, Set-Private-Key and Set-Public-Key.

Let k be a security parameter given to the Setup algorithm and IG be a Bilinear Diffie-Hellman Problem(BDH) parameter generator with input k.

Setup. (This operation is performed by the KGC): This algorithm runs as follows:

1) Run IG on input k to generate output $< G_1, G_2, e >$, where G_1 and G_2 are groups of some prime order q and e: $G_1 \times G_1 \rightarrow G_2$ is a pairing.
2) Choose an arbitrary generator $P \in G_1$.
3) Select a master-key s uniformly, at random, from Z_q^*, and set $P_0 = sP$.
4) Choose cryptographic hash functions H_1: $\{0,1\}^* \rightarrow G_1^*$ and H_2: $G_2 \rightarrow \{0,1\}^n$, where n is the bit-length of the plaintexts taken from some message space $M = \{0,1\}^n$ with a corresponding cipher text space $C = G_1 \times \{0,1\}^n$. Then, the KGC publishes the system parameters $params = < G_1, G_2, e, n, P, P_0, H_1, H_2 >$, while the secret master-key s is securely saved by the KGC.

Set-Secret-Value. (performed by the user): This algorithm takes as inputs *params* and entity m's identifier ID_m. Entity m selects $x_m \in Z_q^*$ at random and outputs x_m as m's secret value. Then, it computes $X_m = x_m P$ and sends X_m to the KGC.

Partial-Private-Key-Extract. (performed by the KGC): This algorithm takes as inputs an identifier $ID_m \in \{0,1\}^*$ and X_m and carries out the following steps to construct the partial private key for entity m with identifier ID_m.

1) Compute $Q_m = H_1(ID_m || X_m)$.
2) Output the partial private key $D_m = sQ_m \in G_1$.

Entity m, when armed with its partial private key D_m, can verify the correctness of the partial private key D_m by checking $e(D_m, P) = e(Q_m, Q_m)$.

Set-Private-Key. (performed by the user): This algorithm takes as inputs *params*, entity m's partial private key D_m and m's secret value $x_m \in Z_q^*$. Entity m transforms the partial private key D_m to private key S_m by computing

$$S_m = x_m D_m = x_m s Q_m \in G_1.$$

Set-Public-Key. (performed by the user): This algorithm takes as input *params* and entity m's secret value $x_m \in Z_q^*$ and constructs m's public key as $P_m =< X_m, Y_m >$, where $X_m = x_m P$ and $Y_m = x_m Q_m = x_m sP$.

3 System Design

The proposed secure e-mail system should securely exchange 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 email communication system. In order to achieve this goals, some decisions are 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 email clients only, and this will allow any organization to apply this system without having to modify the underlying network architecture.

- The analysis of the current encryption schemes shows that different aspects of the key distribution technology have attracted criticism, and shows that most of these aspects are related directly to the digital certificates management complexity. On the other hand, CLPKC provides a comparable security and an equivalent functionality, and does not need any digital certificates. Thus, CLPKC represents an excellent replacement to the existing email security technology, and it will be adopted in the design of this system.

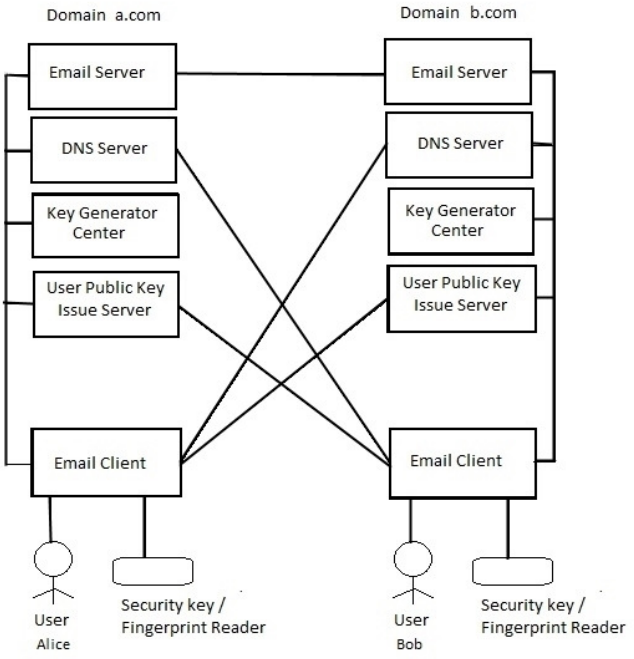

Figure 1: System architecture

3.1 Building Blocks

Figure 1 illustrates the architecture of the proposed system. Consider a user Alice at the domain *a.com* who wishes to send a secure email to user Bob at the domain *b.com*. Various components and operations of the proposed system are described as follows.

A user public key issue server issues public keys for users. The functionality of the user public key issue service may be implemented as part of the KGC server for small user base domains.

The user registration module is incorporated within the user public key issue server. During the user registration phase, the system asks for an email address, a password and a secret value (x_m). The user database can be linked to the email server. The registration module will contact KGC to obtain public parameters *params* and update the KGC with $x_m P$. The registration module also calculates the public key of the user and stores it in the user public key issue server for distribution. To keep the secret value with the user safely, we propose a usb security key token to store the secret with password protection.

An email plug-in attached to an email client is proposed for signing, verifying, encrypting and decrypting. If web-based email is used, all of the security functionalities should be incorporated in the email web server with the help of a client side code execution module (for example, java code under java virtual machine). During the security operation, the plugin module attached to the email client or client side code execution module reads the secret value stored in the USB security token with the help of software drivers.

Multi-factor (i.e., two or more factors) authentication is now a requirement for effective secure authentication. Multi-factor authentication is commonly defined as:

- Something the user knows (e.g., Password, PIN);
- Something the user has (e.g., ATM card, smart card, security token); and
- Something the user is (e.g., Biometric characteristic, such as a fingerprint, palm pattern).

For an efficient security system, it is recommended to use "authentication methods that depend on more than one" of these three factors (i.e., "multi-factor" authentication).

The same security token or a finger print system [33, 35] can be used along with a conventional username and password for authentication during registration or the update of a secret value. Biometrics, which refers to distinctive physiological and behavioral characteristics of human beings, are more reliable means of authentication than a traditional password based system. The fingerprint is the most widely used biometric because of its uniqueness and immutability. For the fingerprint system, the user has to be enrolled with the user public key server, while registration and a special software plug-in are required during authentication to verify the current captured image with the previously recorded fingerprint.

3.2 User Public-Key Distribution

DNS provides a well-established and trusted naming and routing infrastructure for domains. Hostname to IP address mappings and mail routing (using MX records) rely on it. Recently, DNS has been proposed as a public-key infrastructure with Domain Keys Identified Mail (DKIM) [2] by the Internet Engineering Task Force (IETF). In the DKIM scheme, public keys generated by RSA scheme are stored in specially named DNS records. Specifically, DKIM reserves the domainkey subdomain for every domain with an MX record. Any entries within this subdomain are public keys in base64-encoding with some associated parameters. By keeping each public-key record short (i.e., less than the size of a single UDP packet), this DNS-based key distribution mechanism is functional across a large portion of existing DNS servers.

Our proposed system uses the same technique as in DKIM for specifying the address of the user public key issue server of the domain.

3.3 Domain Setup

Figure 2 shows the domain setup steps. In the basic domain setup, Alice, with email address alice@a.com, will obtain her partial secret key from a key server (KGC) dedicated to her domain a.com.

The detailed setup procedure of the domain is defined as follows:

- Choose an identity-based signature scheme from the various schemes, e.g., the Boneh and Franklin method.

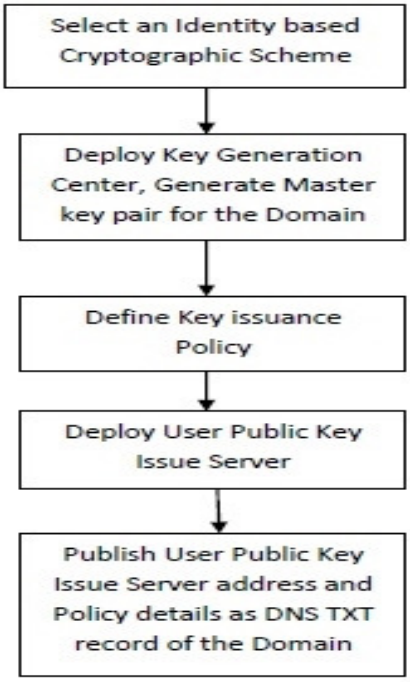

Figure 2: Domain setup

- Generate a master key pair (public and master secret) for this scheme.
- Define key issuance policy *Policy* for the system. This includes definition on whether emails originating from this domain can, should, or must be signed.
- Publish the user public key issue server details and *Policy* as a DNS TXT record corresponding the MX record for a.com. The DNS record content is formatted as ibclses=<name of user public key issue server>, policy=<Policy>.

User Identities. A user has to first register to generate a secret value, store it in a USB security token and then pass the generated public key to the public key issue server. A user's public key can be derived by calling the user public key web service and passing the user's identity through *id_string*. We propose a standard format for *id_string* to specifically support email address authentication with domain policies and key expiration mechanisms.

Key Expiration. To provide simple key revocation capabilities, the user identity string includes key expiration information. Specifically, *id_string* includes the last date on which the key is valid - the expiration date -as a formatted character string. An id string is thus constructed as: <email>, <expiration date>.

For example, an identity string *id_string* for Bob would be: bob@b.com, 2014-07-10.

3.4 Domain Policies

Once an email domain decides to deploy an email security system, it simply needs to create a key server and a user public key distribution server for the domain and specify this server address in the appropriate DNS record. We propose that this record include a Policy parameter to specify how the domain chooses to participate in the secure email architecture. Policy determines the domain's requirements on its email users as well as its guarantees to any recipients. Three external Sign policy values are used:

None. Users of this domain may sign their emails. If the signature and verification fails, no warning header will be added by the recipient email signature verification system.

Basic. Users of this domain may sign emails with keys issued by this key server. If the signature and verification fails, a warning header will be added by the recipient email signature verification system.

Strong. Users of this domain are required to sign all of their emails with a key issued by this domain. The message will be rejected if verification fails.

Internal policies can also be implemented by adding some standing instructions to the email client. Examples include:

alice@a.com * E - Encrypt all messages from alice@a.com

alice@a.com bob@b.com S - Sign the message from Alice to Bob.

*@a.com tax@gov.gv ES - Sign and Encrypt all messages from domain a.com to tax@gov.gv

3.5 The Proposed System

Figure 3 shows the steps for sending secure email.

Assume that client Alice has a private key, $S_{Alice} = x_{Alice}D_{Alice}$, and a public key $P_{Alice} =< X_{Alice}, Y_{Alice} >$, while client Bob has a private key, $S_{Bob} = x_{Bob}D_{Bob}$, and a public key, $P_{Bob} =< X_{Bob}, Y_{Bob} >$. The public keys of all clients are available through the secure web service on the User public key issue server. The working of the security functionality is based on how the internal and external domain policies are specified for the domain.

The basic operation of the security functionality is as follows:

Encryption.

1) Assemble id_string_{Bob}, an identity string for Bob, using the current date+15 days as the expiration date to view the message within 15 days: bob@b.com, 2012-07-31.

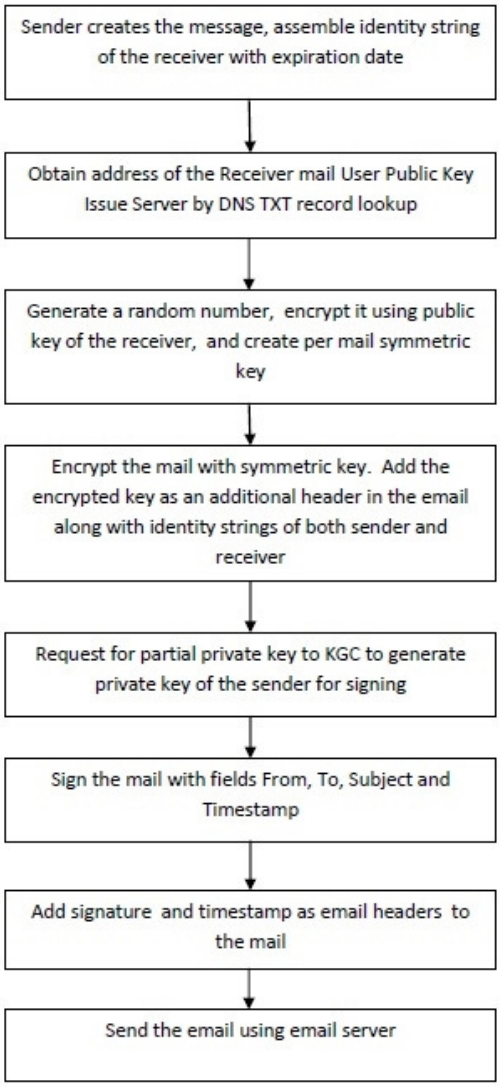

Figure 3: Sending secure email

2) Obtain the address of the User public key issue server for the domain b.com using DNS TXT record lookup. Collect the public key of Bob P_{Bob} from the user key issue server for Domain b.com using secure web services. Then, check that $\hat{e}(X_{Bob}, Q_m) = \hat{e}(Y_{Bob}, P)$ to authenticate the public key of client Bob.
3) Generate a random number $t \in Z^*$ and encrypt it using the public key of client Bob as $t^* = EP_{Bob}(t)$. The random number has the feature of Perfect Forward and Backward Secrecy, which is always fresh and unrelated to any past or future sessions.
4) Compute $K_{AliceBob} = tx_{Alice}X_{Bob}$, and then compute the per-mail symmetric key

$$K = H_2(Q_{Alice}||Q_{Bob}||K_{AliceBob}).$$

5) Encrypt the mail M by using the AES algorithm with the symmetric key K as $M^* = E_K(M)$.
6) Add the encrypted value t^* at the beginning of the encrypted mail M^* (i.e., $M^* = M||t^*$) as additional headers as shown below:
   - The exact id_string_{Alice} in SMTP header *X-IBCLSES-Sender-IDString*.
   - The exact id_string_{Bob} in SMTP header X-IBCLSES-Recipient-IDString.
   - t^* in base64-encoding in SMTP header X-IBCLSES-Encryption-Key

Signing.

1) Request a partial private key from the KGC to generate the private key for client Alice.
2) Sign the encrypted mail M^* (or M is encryption not required) along with the fields: From, To, Subject, and Timestamp, to produce the signature S using client Alice's private key.
3) Add additional headers to the mail messages as given below:
   - S in base64-encoding in SMTP header X-IBCLSES -Signature;
   - The exact id_stringAlice in SMTP header X-IBCLSES-Sender-IDString (if not already added);
   - The time stamp used in the signature in SMTP header X-IBCLSES-Timestamp.

The Email client sends Alice's mail using SMTP.

Figure 4 shows the steps involved in receiving the secure email. The detailed steps for signature verification and decryption is given below.

Signature Verification.

1) Download the secure mail from the mail server to the Email client of Bob using the IMAP/POP3 protocols.

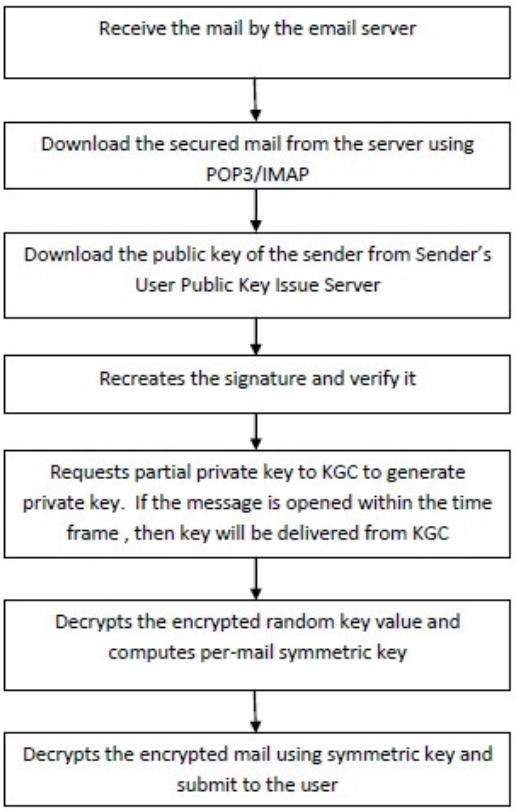

Figure 4: Receiving secure email

2) If the message header has the *X-IBCLSES-Signature* identifier, then the message is signed; determine the sender's email address, alice@a.com, and domain name, a.com, according to the email's From field and sender *id_string* email header *X-IBCLSES-Sender-IDString*.
3) Obtain the address of the User public key issue server for the domain a.com, by DNS TXT record lookup. Collect the public key of Alice P_{Alice}, from the user key issue server for Domain b.com using secure web services. Then, check that $\hat{e}(X_{Alice}, Q_m) = \hat{e}(Y_{Alice}, P)$ to authenticate the public key of client Alice.
4) Recreate the message M (or M^*) that was signed, using the declared From, To, and Subject fields, the email body, and the timestamp declared in *X-IBCLSES-Timestamp*.
5) Verify the signature.

Decryption.

1) If the message header has the *X-IBCLSES-Encryption-Key* identifier, then the message is in encrypted form.
2) Request a partial private key from the KGC to generate the private key for client Bob by passing the *id_string*. The system will not provide the partial key if the expiry date mentioned in the *id_string* is over.
3) Decrypt t^* using client Bob's private key
$$t = DS_{Bob}(t*).$$
4) Compute $K_{BobAlice} = tx_{Bob}X_{Alice}$.
5) Compute the per-mail symmetric key $K_{Bob} = H_2(Q_{Bob}||Q_{Alice}||K_{BobAlice})$. (i.e., $K_{Bob} = K_{Alice}$).
6) Decrypt the encrypted mail M^* using the symmetric key K as $M = D_K(M^*)$.

4 Implementation of the Proposed Secure Email System

The prototype system was developed using the C++ programming language. To implement the IBC protocol, there is a need for a cryptographic library thatcan provide both Elliptic Curve Cryptography (ECC) and bilinear pairing functions. From the available literature, we selected the Miracl library. Miracl is an open source C/C++ Multiprecision Integer and Rational Arithmetic Cryptographic library. All of the basic encryption functions, such as setup, extract, encrypt and decrypt functions,were developed using the C++ language.

Security services for the email client were implemented as an extension to the Mozilla Thunderbird email client using JavaScript and the C++ library.

5 Security Discussion

The proposed system is secure against the standard security model because it is based on the Elliptic Curve Discrete Logarithm and Collision Resistant hash function standard cryptographic primitives. Other security properties provided by our solution follows.

1) End-to-end user authentication: Because the proposed mailing security system uses the CLPKC with the binding technique, both sender and receiver will authenticate each other using a pairing operation.
2) Key agreement between sender and receiver without interaction: The sender and receiver of the proposed system compute the shared secret key using their own secret values, the other party's public key and a randomly generated number, that is encrypted by the receiver's private key without any interaction between sender and receiver. Therefore, the proposed system is secure against the man-in-the-middle type attack.
3) Confidentiality of the message: The mail content is encrypted by a symmetric crypto system(such as AES), which guarantees the confidentiality of the message. The symmetric key can only be decrypted by the receiver.

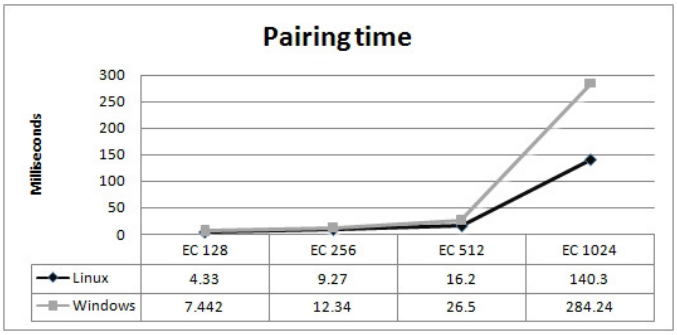

Figure 5: Pairing time

4) Message integrity and non-repudiation: Because the sender signs the email message using his/her private key, the integrity and non-repudiation feature of the message can be verified, and the sender cannot disown the ownership of the email message.

5) Forward and backward secrecy: In the proposed system, each message session key during the message transmission is unique because both the sender and receiver use the random number t, which is generated by the sender and encrypted by the public key of the receiver in each session. The random number has the feature of Perfect Forward and Backward Secrecy, which is always fresh and unrelated to any past or future sessions.

6 Benchmark and Conclusion

The most costly procedure of the proposed system is the pairing operation. Benchmarking tests were performed on the operation, and the mean of 500 iterations was taken. The test only counts the time for a pairing, while the random point generation part is not considered. We conducted the test on an Intel Core i5-2400 CPU @3.10 GHz with 4GB RAM 1066MHz under the Windows 8 32 bit operating system and Oracle Linux 5.5 64-bit. For ECC 512 bits, we obtained an average speed of 16.2 milliseconds in Linux and 26.5 milliseconds in Windows, and for 1024 bits, we obtained 140.3 milliseconds in Linux and 284.24 milliseconds in Windows. Detailed performance analysis is given in Figure 5. From the test results, it is clear that we get better performance in 64 bit Linux system than 32-bit Windows and the proposed open standard system is very efficient and can be used in secure messaging as an alternative to the certificate based conventional PKI system.

This paper proposed an end-to-end secure mailing system based on certificate-less public key Cryptography, with DNS as the mechanism to publish a user's public key server address. The sender and receiver are able to compute the same secure secret symmetric key without any message exchange between them. This avoids a man-in-the-middle attack to obtain details of encryption/decryption key and hence the contents of the email.

Additionally, the proposed mail system is based on Elliptic Curve Cryptography, which is very efficient compared to conventional RSA based email systems and is also free from the heavy burden of certificate management of PKI. It avoids the key escrow problem of IBC based email systems by incorporating a partial private key generation system. The usability of IBE based systems are much better than PKI or PGP based systems. Moreover, the proposed system is based on standard cryptographic primitives, which makes it secure against the standard security model.

References

[1] B. Adida, D. Chau, S. Hohenberger, and R. L. Rivest, "Lightweight email signatures (extended abstract)," in *Proceedings of the 5th international conference on Security and Cryptography for Networks (SCN'06)*, pp. 288–302, 2006.

[2] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'01)*, LNCS 2139, Springer Verlag, pp. 213–229, 2001.

[3] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, *Open PGP Message Format*, Technical Report RFC 2440, Nov. 1998.

[4] S. Chatterjee and P. Sarkar, *Identity-based Encryption*, Springer Science+Business Media, LLC, 2011.

[5] L. Chen, K. Harrison, N. Smart, and D. Soldera, "Applications of multiple trust authorities in pairing based cryptosystems," in *International Conference on Infrastructure Security (InfraSec'02)*, LNCS 2437, Springer, pp. 260–275, 2002.

[6] T. Chen and S. Ma, "A secure email encryption proxy based on identity-based cryptography," in *International Conference on MultiMedia and Information Technology (MMIT'08)*, pp. 284–286, 2008.

[7] S. S. M. Chow, "Removing escrow from identity-based encryption," in *Public Key Cryptography (PKC'09)*, LNCS 5443, Springer, pp. 256–276, 2009.

[8] M. Cooper, *Internet X.509 Public Key Infrastructure (Latest Draft)*, IETF Internet Drafts, Jan. 2005.

[9] M. Crispin, *Internet Mail Access Protocol (ver. 4)*, Technical Report RFC 1730, Dec. 1994.

[10] D. Crocker, T. Hansen, and M. Kucherawy, *Domainkeys Identified Mail (DKIM) Signatures*, Technical Report RFC 6376, Sep. 2011.

[11] D. Eastlake, *Domain Name System Security Extensions*, Technical Report RFC 2535, Mar. 1999.

[12] B. A. Forouzan, *Cryptography and Network Security*, India: Tata McGraw-Hill Publishing Company Limited, 2007.

[13] Fortinet, *Forti Mail Identity Based Encryption*, Jan. 2014. (http://www. fortinet. com)

[14] M. Franklin and D. Boneh, "Identity based encryption from the weil pairing," *Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[15] R. Gangishetti, M. C. Gorantla, M. Das, and A. Saxena, "Threshold key issuing in identity-based cryptosystems," *Computer Standards & Interfaces*, vol. 29, no. 2, pp. 260–264, 2007.

[16] R. Gangishetti, M. C. Gorantla, M. L. Das, A. Saxena, and V. P. Gulati, "An efficient secure key issuing protocol in id-based cryptosystems," in *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing (ITCC 2005)*, pp. 674–678, Las Vegas, USA, Apr. 2005.

[17] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology (EUROCRYPT'03)*, LNCS 2656, Springer, pp. 272–293, 2003.

[18] E. Gerck, *Secure Email Technologies X.509/PKI, PGP, IBE and Zmail: A Usability and Security Comparison*, ICFAI University Press, pp. 171–196, 2007.

[19] C. Gu and Y. Zhu, "New efficient searchable encryption schemes from bilinear pairings," *International Journal of Network Security*, vol. 10, no. 1, pp. 25–31, 2010.

[20] M. Hassouna, N. Mohamed, B. Barry, and E. Bashier, "An end-to-end secure mail system based on certificateless cryptography in the standard security model," *International Journal of Computer Science Issues*, vol. 10, no. 2, pp. 264–271, 2013.

[21] P. Hoffman, *SMTP Service Extension for Secure SMTP over Transport Layer Security*, Technical Report RFC 3207, Feb. 2002.

[22] M. P. Hoyle and C. J. Mitchell, "On solutions to the key escrow problem," in *State of the Art in Applied Cryptography*, LNCS 1528, Springer, pp. 277–306, 1998.

[23] HP, *HP Security Voltage*, Jan. 7, 2014. (http:// voltage. com)

[24] S. Kwon and S. H. Lee, "Identity-based key issuing without secure channel in a broad area," in *Information Security Applications*, LNCS 4298, Springer, pp. 30–44, 2007.

[25] P. Mockapetris, *Domain Names - Concepts and Facilities*, Technical Report RFC 1034, Nov. 1987.

[26] P. Mockapetris, *Domain Names - Implementation and Specification. IETF - Network Working Group, The Internet Society*, Technical Report RFC 1035, Nov. 1987.

[27] B. Ramsdell, *S/MIME Version 3 Message Specification*, Technical Report RFC 2633, June 1999.

[28] RSA Laboratories, "What is public-key cryptography," Jan. 20, 2014. (http://www. rsa. com/ rsalabs/)

[29] A. R. Sattam and P. Kenneth, "Certificateless public key cryptography a full version," in *Asiacrypt'03*, LNCS 2894, Springer, pp. 452–473, 2003.

[30] A. Shamir, "Identity based cryptosystems and signature schemes," *Computer Science*, vol. 196, pp. 47–53, 1984.

[31] D. Sharma and D. Jinwala, "Key generation protocol in IBC," *International Journal of Network Security*, vol. 15, no. 5, pp. 341–349, 2013.

[32] T. S. Sobh and M. I. Amer, "PGP modification for securing digital envelope mail using COM+ and web services," *International Journal of Network Security*, vol. 13, no. 2, pp. 79–91, 2011.

[33] J. Tian, L. Li, and X. Yang, "Fingerprint-based identity authentication and digital media protection in network environment," *Journal of Computer Science and Technology*, vol. 21, no. 5, pp. 861–870, 2006.

[34] J. Wang, X. Bai, J. Yu, and D. Li, "Protecting against key escrow and key exposure in identity-based cryptosystem," in *Theory and Applications of Models of Computation (TAMC'07)*, LNCS 4484, Springer, pp. 148–158, 2007.

Lecture Notes in Computer Science Volume 4484, 2007, pp 148-158

[35] Z. Wu, J. Tian, L. Li, C. P. Jiang, and X. Yang, "A secure email system based on fingerprint authentication," in *Scheme Intelligence and Security Informatics*, pp. 250–253, 2007.

[36] P. Zimmerman, *Pretty Good Privacy*, Jan. 20, 2014. (http://www. pgp. com)

Mr. Suresh Kumar Balakrishnan, currently working as an Engineer in Computer Division at Sree Chitra Tirunal Institute for Medical Sciences & Technology, Thiruvananthapuram, India is an M.Tech holder in Software Engineering from Cochin University of Science and Technology, Kochi, India. He has well-versed experience in implementing Computer Infrastructure, Network Security, Systems Management, Database Management and Hospital Information System. He has more than 17 years of experience in Computer Engineering fields. He is pursuing Ph.D in Computer Science at Indira Gandhi National Open University, New Delhi, India.

Dr. V. P. Jagathy Raj, currently working as Professor in Operations and Systems Management at School of Management Studies, Cochin University of Science and Technology, Kochi. He is equally well-versed in both Engineering and Management related areas. He has more than 24 years of teaching and research experience in Engineering and Management areas. He has more than 160 research publications in National and International Journals and Conference Proceedings to his credit in both Engineering and Management related areas. He has also presented number of research papers in National and International conferences.