An Identity-Based Key Infrastructure Suitable for Messaging Applications

Ayşe Gül Karatop and Erkay Savaş

***Abstract*—Identity-based encryption (IBE) systems are relatively recently proposed; yet they are highly popular for messaging applications since they offer new features such as certificateless infrastructure and anonymous communication. In this paper, we intended to propose an IBE infrastructure for messaging applications. The proposed infrastructure requires one registration authority and at least one public key generator and they secret share the master secret key. In addition, the PKG also shares the same master secret with each user in the system in a different way. Therefore, the PKG will never be able to learn the private keys of users under non-collusion assumption. We discuss different aspects of the proposed infrastructure such as security, key revocation, uniqueness of the identities that constitute the main drawbacks of other IBE schemes. We demonstrate that our infrastructure solves many of these drawbacks under certain assumptions.**

***Index Terms*—Identity Based Encryption, Elliptic Curve Cryptography**

I. Introduction

Identity-based encryption (IBE) scheme is a public key cryptosystem where the public keys are unique identities in arbitrary string forms. For instance, e-mail addresses, names, pseudonyms or IP addresses can serve as a public key in IBE systems. The original concept was initially introduced by Shamir in 1984 [14] while the first practical realization of IBE system is based on pairing-based cryptography by Boneh and Franklin [1]. With the advent of pairing-based cryptography; new applications of IBE cryptosystem as well as new techniques, to realize it more efficiently, become the major focus of the contemporary research. Generally speaking, in IBE cryptosystems, there exists a trusted third party, so-called Private Key Generator(PKG), which is responsible for generating global parameters to be employed in the system as well as the private keys for the registered users. Users obtain their private keys from the PKG, in order to decrypt their messages intended for them. The secure delivery of private keys should be performed over secure channels, where confidentiality and authentication are provided.

IBE is principally a public key cryptosystem, where each user has a public and private key pair. To illustrate, suppose that a user, Alice, wants to send a message to Bob. She encrypts the message with Bob's unique public key, e.g. his e-mail address 'bob@sabanciuniv.edu'. Bob requests the corresponding private key from the PKG, to decrypt the message. The PKG calculates the private key, sends to Bob, and Bob consequently decrypts the message.

Since the bound between the user and the user's public key is based on an inherent or real-word relationship (e.g. user/name, user/e-mail address, user/assumed role etc.), the need for an infrastructure is seen by some not as comprehensive as the conventional public key infrastructure(PKI). Whereas as elaborately pointed out in [4], a fully-functional IBE system would also require a complex infrastructure in which some aspects have not been fully investigated. Firstly, there is the issue of uniqueness of public keys in IBE since real world names or identities tend to be not unique. Therefore, there should be a registration authority to keep track of used names, i.e. public keys. Secondly, the key revocation could lead to some inconvenience since one may find difficult to obtain a new descriptive name for oneself such as finding a new name. One way to revoke a key without actually changing the public key requires that system parameters be changed resulting in changing of private key of every user in the system. And finally, all IBE schemes have the key escrowing property, which is considered as a weakness since the PKG knows the private key of every user. Thus, the PKG can not only decrypt any message but also can fabricate a signature on behalf of any user.

In this paper, our contribution is proposing solutions to some of the shortcomings of IBE systems. Our basic construction follows the idea of secret sharing of the master secret key between two semi-honest parties, namely the private key generator (PKG) and the registration authority (RA). In addition, the PKG shares the same master secret key with each user in a different way, having one share for each user registered in the system. Thus, a user and the PKG have to participate in a protocol to generate the private key for the user. A user's only interaction with the RA is during the registration phase, in which the RA not only checks the uniqueness of the identity but also assists in the protocol that generates two new shares of the master secret key for the user and the PKG. One benefit of our model is that there is no need to employ a secure channel between the PKG and users to deliver private keys since the PKG can send only its share of the private key to users.

We also propose to use ever-changing public keys by attaching date information to the natural identities of users from the perspective of communication models.

We give a brief information about identity-based encryption systems and their mathematical background in the second section. The third part includes a detailed explanation of our infrastructure. The analysis of our proposed scheme is given in section four. An additional property, namely the anonymity

School of Electrical and Computer Engineering; Sabanci University, Istanbul, Turkey; Email: aysegulk@su.sabanciuniv.edu
School of Electrical and Computer Engineering; Sabanci University, Istanbul, Turkey; Email: erkays@sabanciuniv.edu

of our system is discussed in section five. In the sixth section, implementation details are given. The paper ends up with conclusion and future works.

II. Identity-Based Encryption Systems

In this section, we will give background information of an identity-based encryption scheme.

A. Mathematical Background

We use identity-based encryption (IBE) systems that utilize elliptic curves and pairing operations as proposed in [1]. An elliptic curve $E(F_p)$ over a finite field F_p is defined with the equation,

$$y^2 = x^3 + ax + b \text{ with } a, b \in F_p$$

The solutions to this equation are called elliptic curve points, and shown as $P = (x, y)$, where x and y are the coordinates and elements of the underlying field F_p. The points on elliptic curve along with so-called point at infinity form an additive group. We can denote the point addition as $P+Q$, and define elliptic curve scalar multiplication of an elliptic curve point P by an integer α, as αP. The order of a point is the smallest integer, n, such that $nP = \mathcal{O}$, where $\mathcal{O}$ denotes the point at infinity, which is the identity element of the elliptic curve group. The security of elliptic curves depends on the difficulty of solving elliptic curve discrete logarithm problem (ECDLP). The ECDLP basically states that given two points Q and P from the equation, $Q = \alpha P$, it is computationally difficult to find α.

Bilinear maps over elliptic curve points play a central role in IBE systems. A bilinear map is defined over two groups of the same prime-order q denoted by G_1 and G_2. G_1 is an additive group and is formed of a group of points on elliptic curves while G_2 is a multiplicative group. Bilinear map, therefore, is defined as $G_1 \times G_1 \rightarrow G_2$. Basically, a bilinear map, which is denoted as $\hat{e}(\cdot, \cdot)$, accepts two elements as input from G_1 and returns an element in G_2. The major property of bilinear maps is bilinearity [5] which is explained below.

$$\hat{e}(xP, yQ) = \hat{e}(P, yQ)^x = \hat{e}(P, xyQ) = \hat{e}(P, Q)^{xy}$$
$$\forall P, Q \in G_1, \forall x, y \in Z_q$$

Tate and Weil pairings [10], [6] are the most used pairing functions. Our scheme is based on Tate pairing which is, in general, more efficiently calculated than the Weil pairing.

Public and private keys of users, in IBE systems, are elliptic curve points. For this purpose, a hash function, H_1 which is defined as $H_1 : \{0.1\}^* \rightarrow G_1$, is employed to convert a string of arbitrary length (i.e. identity) to a point on the underlying elliptic curve. In addition to H_1, another hash function, $H_2 : G_2 \rightarrow \{0, 1\}^n$ is used in encryption and decryption phases. For further information about elliptic curves and pairing based cryptography one can profitably refer to [8] and [3].

B. Work Flow of IBE

In general, an IBE System consists of four phases [12]:

1) **Setup phase:** consists of two steps:
   - Selection of the elliptic curve and the master key, s, and the generation of the public key of the system, $P_{SYS} = sP$, where P is the generator point of G_1, group of chosen elliptic curve.
   - Selection of hash functions, H_1, H_2 and the bilinear mapping function.
2) **Extraction:** The private key generator generates the users' private. The public key of a user (ID) is denoted as Q_{ID} while the private key of a user is denoted as D_{ID}.
   $$Q_{ID} = H_1(ID) \text{ and } D_{ID} = sQ_{ID}$$
   where ID is an arbitrary string information
3) **Encryption:** Encryption is performed by using the receiver's public key (say Alice) as follows:
   - $(U, V) = (rP, M \oplus H_2(g_Q))$
   - where $r \in_R Z_q^*$ (i.e. r is randomly selected in Z_q)
   - and $g_Q = \hat{e}(Q_A, P_{SYS})^r$ and $\oplus$ denotes exclusive-OR operation.

   Here M is the plaintext and the pair (U, V) is the ciphertext, which is consequently sent to Alice.
4) **Decryption:** In decryption phase, the ciphertext (U, V) can only be decrypted if the receiver's private key (D_A) is known. The following steps are applied in decryption process:
   $$V \oplus H_2(g_{Q'}) = M \text{ where } g_{Q'} = \hat{e}(D_A, U)$$

III. Our Infrastructure

This section describes the main steps in the proposed infrastructure omitting the encryption and decryption phases since they are identical to the original IBE encryption and decryption schemes outlined above.

A. Setup Phase

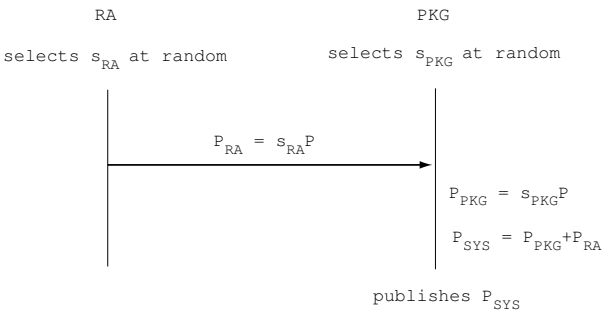

Fig. 1. The two-party protocol for computing P_{SYS}

In our infrastructure, we utilize secret sharing of the master key, s. With this purpose, two semi-honest parties are formed,

the Private Key Generator(PKG) and the Registration Authority(RA). The RA is responsible for registering users in the beginning while the PKG is responsible for key distribution. In addition, the RA and PKG share the master secret key as follows: Initially, the RA and the PKG choose two random secret keys, s_{RA} and s_{PKG}, where $s = s_{RA} + s_{PKG}$ is the master key. Since s must not be known by two semi-honest parties, we stipulate that the RA and the PKG do not collude with each other. A two-party protocol for generating the secret share and the public key of the system P_{SYS} is illustrated in Figure 1.

After selecting its secret share of master secret key, the RA computes $P_{RA} = s_{RA}P$, which is its share of public key of the system, and sends it to the PKG. Similarly, the PKG computes its share of system public key, $P_{PKG} = s_{PKG}P$ and performs the elliptic curve addition $P_{SYS} = P_{PKG} + P_{RA}$. Consequently, the PKG publishes the system public key, P_{SYS}.

B. Registration phase

In the registration phase, the user is first introduced to the system by a secure three-party protocol that involves the user, the RA, and the PKG. The aim of the three-party protocol is two-fold: i) check the uniqueness of the user identity, and ii) securely compute new shares of the master secret and give one share to the user and the other to the PKG. The protocol steps are illustrated in Figure 2. The registration phase utilizes public key cryptography and we assume that the user (i.e. Alice in Figure 2) knows the public keys of the RA and PKG. $E_{RA}[x]$ and $E_{PKG}[x]$ stands for the encryption of x with public key of the corresponding party, i.e. the public keys of RA and PKG, respectively. The PKG uses a homomorphic public key cryptosystem similar to the one in [13]. Therefore, we have $E_{PKG}[m_1] \cdot E_{PKG}[m_2] = E_{PKG}[m_1 + m_2]$.

The protocol steps are explained as follows:

- **Step 1** The user (Alice in Figure 2), for the first and last time, contacts the RA by sending her identity (A) in the first message. Alice also encrypts the difference between her secret share s_A and random number r_1 using the public key of the RA and sends the resulting ciphertext $X = E_{RA}[r_1 - s_A]$ along with her identity A to the RA.
- **Step 2** The RA first checks whether the ID of Alice, A is unique; if not, it helps Alice choose a unique identity. It then obtains the difference $r_1 - s_A$ by decrypting X and adds its own share of master secret, s_{RA} to the difference. It, subsequently, encrypts $r_1 - s_A + s_{RA}$ and sends the resulting ciphertext $Y = E_{PKG}[r_1 - s_A + s_{RA}]$ back to Alice.
- **Step 3** Alice removes the random number r_1 by performing the operation $E_{PKG}[r_1 - s_A + s_{RA}] \cdot E_{PKG}[-r_1] = E_{PKG}[s_{RA} - s_A]$. The resulting ciphertext $E_{PKG}[s_{RA} - s_A]$ is sent to the PKG. Alice also sends $E_{PKG}[s_A P]$ to the PKG, which serves as her public key to authenticate Alice to the PKG in their subsequent transactions.
- **Step 4** The PKG first performs the following operation $E_{PKG}[s_{RA} - s_A] \cdot E_{PKG}[s_{PKG}] = E_{PKG}[s - s_A]$. It then decrypts the resulting ciphertext and obtain its share of master secret $s_{A\prime}$. Note that $s = s_A + s_{A\prime}$. The PKG also decrypts $E_{PKG}[s_A P]$ and obtains Alice's public key $s_A P$ that is used only in authentication protocol described in subsequent sections.

As a result of registration phase, the user and the PKG come to have different shares of the master secret s. Therefore, a user and the PKG must collaborate to generate a private key corresponding to any public key chosen by the user. Provided that none of the users and the PKG do not collude, the master secret will never be revealed. Note that no coalition of users is able to construct the master secret since the user shares themselves do not contain any information about the master secret.

We have two motivations to believe that non-collusion assumption is valid and realistic: i) the PKG is semi-honest, and therefore does not try to learn about the secret shares of the users unless openly told by the users, and ii) a user does not want to reveal its share to the PKG since doing so gives the PKG the ability to access the messages intended for the user and to generate signatures on behalf of the user. Furthermore, the secret share s_A of a user can always be kept in a trusted zone of its hardware and will never leave this zone in the clear. For instance, the first step of the registration phase calculates the difference of this secret share and a random number, $r_1 - s_A$. The difference will be known; but neither r_1 nor s_A can be deduced from the difference. Consequently, the user itself would never learn what s_A is in the first place to reveal it to PKG. The user secret share is just a randomly selected number and can be changed easily. In case of compromise, the revocation only requires that the user secret share is changed and the registration phase is repeated.

C. Public Key Selection and Private Key Extraction

In identity-based encryption system, public keys are generally arbitrary strings that contain identity of the user and other relevant publicly available information. Furthermore, the public keys can contain descriptive information about the intended recipient. This clearly alleviates the problem of public key certification used to establish a binding between the public key and the identity of public key owner. Apparently, this bond is inherent in IBE systems. This, nevertheless, complicates the key revocation problem since changing a user's public key entails changing of its identity. Changing one's identity raises certain concerns since finding another descriptive name for an individual may be difficult on it own right. However, the more important point is the complicated infrastructure (e.g. certification revocation lists) required for informing other users on the compromised or stale public keys.

In messaging applications, on the other hand, the problem of key revocation can be addressed using "ever-changing" public keys. Namely, public key of a user can contain strings related to situational information such as the location, time, date, and role of the user besides the unique identity of the user. We simply propose to include date (or time) information in the identity (hence the public key) of the user. Therefore, the users in our messaging infrastructure has public keys, that are updated frequently. For instance, a user ID may contain date

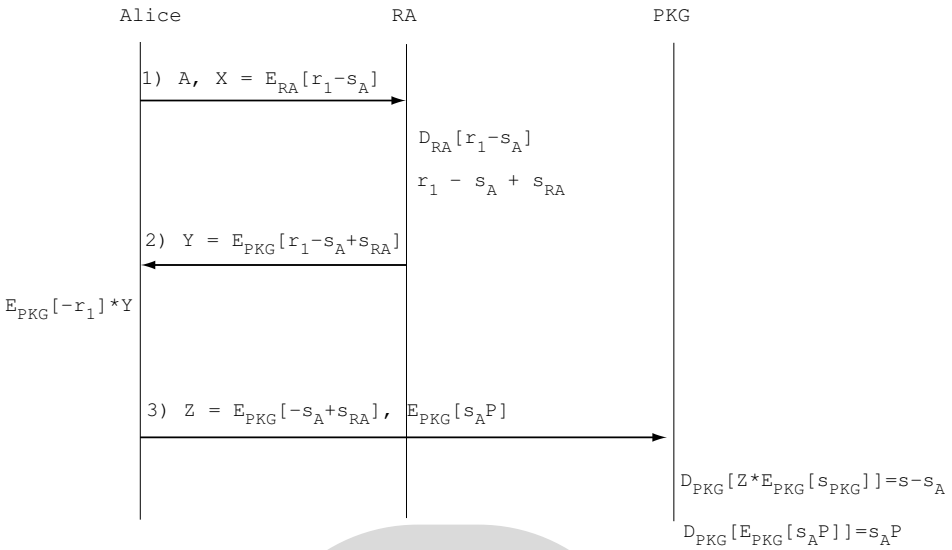

Fig. 2. The registration phase

information, such as August 02, 2007, which is is public information and can be appended to the ID easily. The string "erkays@sabanciuniv.edu:08/07/2007" is an example for ever-changing public keys.

If the public keys change as frequently as every day, then the corresponding private keys must be re-computed as frequently. As mentioned earlier, both the user and the PKG must participate in the private key generation procedure. In classical IBE systems, the secret key is generated by the PKG and then securely transmitted to the user. Before, the key generation, the user must authenticate itself to the PKG and secure channel must be established between the user and the PKG. Otherwise, the private key can be fallen in the hands of other users or worse yet adversaries. The proposed scheme, on the other hand, utilizes only implicit authentication of the user and does not require a secure channel. The private key generation scheme is illustrated in Figure 3.

The user, Alice, selects a public key by appending date and other relevant information to her identity and obtains Q_A, which is sent to the PKG. The PKG then computes its share of the public key $s_{A\prime} \cdot Q_A$ and sends it to Alice. Alice then computes $D_A = s_A \cdot Q_A + s_{A\prime} \cdot Q_A = s \cdot Q_A$, which is her private key, D_A corresponding to the public key Q_A.

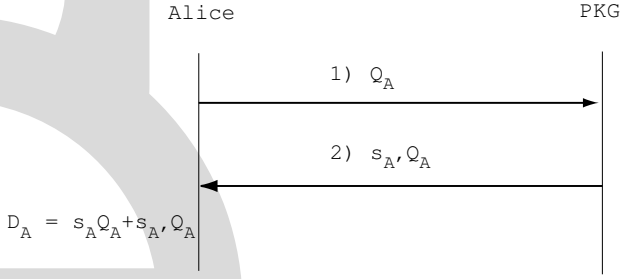

Fig. 3. Private key extraction scheme

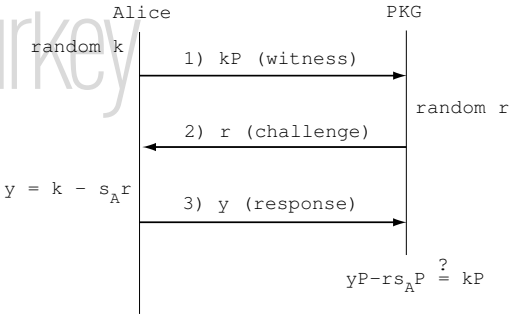

Fig. 4. User Identification to the PKG

D. Identification

In case there is a need for explicit identification of the user to the PKG, they can use a modified version of Schnorr's identification protocol as illustrated in Figure 4. The effort undertaken by the user is one elliptic curve point multiplication with a scalar and one multiplication and one subtraction in modulo n, where n is the order of base point P.

The steps of the identification protocol are summarized below:

- **Step 1** The user Alice, first selects a random integer k and performs the elliptic curve scalar multiplication, kP, where P is the base point for the underlying elliptic curve group. Alice sends kP to the PKG as a witness.
- **Step 2** The PKG selects a random integer r and sends it to Alice as a challenge.
- **Step 3** Alice, upon reception of r, computes $y \equiv k - s_A r \pmod{n}$ and send the resulting value y to the PKG.
- **Step 4** The PKG computes $yP - rs_AP$, where s_AP serves

as the public key of Alice obtained during the registration, and authenticate Alice if the result is the same as the witness kP.

If PKG needs to authenticate itself to Alice, they can use any identification scheme utilizing the public key of the PKG which is assumed to be in possession of Alice.

IV. Analysis of the Proposed Infrastructure

In this section, we analyze the proposed infrastructure from four different perspectives, namely i) security, ii) non-repudiation, iii) validity of public keys, and iv) key revocation.

A. Security

The security of the proposed infrastructure is based on two basic assumptions on the involved parties: i) non-collusion property between certain parties, and ii) semi-honest nature of the PKG and the RA.

Employing two or more trusted parties that do not collude was already proposed by Boneh and Franklin in [1] and also in [2]. In both schemes, a user has to contact all trusted parties to obtain its private key and furthermore the user has to establish a secure channel with each trusted party in this key extraction phase. Our scheme diverges from the previous schemes in two aspects. Firstly, it introduces two trusted-third parties, the private key generator (PKG) and registration authority (RA), which secret share the master secret s and again do not collude with each other. Secondly, each user shares the same master secret with the PKG in a different way. Therefore, a user does not need to contact both trusted parties to acquire his/her private key since s/he can do so using a protocol involving itself and the PKG. Furthermore, the communication between the user and the PKG does not need to be encrypted.

Our second assumption involves the semi-honest nature of the PKG and the RA. Property of semi-honest party was first introduced by Goldreich in [7] and it simply assumes that such parties are honest but curious. In other words, they do not participate in extra protocol activities but gather any leaked information from the protocol. For instance, the PKG will never try to register as a user in the system since this would compromise the master secret to the PKG. The interface for user registration is not available to the PKG. Unless the users openly encrypt their private shares of the master secret with the public key of the PKG and send it to the PKG the semi-honest PKG will never learn the private shares of the users. A user will not reveal his/her private share to the PKG or RA since this share also serves as his/her private key in the identification protocol illustrated in Figure 4. In other words, a user should not collude with the PKG since doing so will enable the PKG to calculate the master secret s.

Another advantage of the proposed infrastructure is that it provides convenience in key distribution. Only assumption we hold in key distribution is that a user who would like to register knows the public keys of the PKG and RA. Users can acquire this knowledge from publicly available resources such as web pages. Furthermore, a user does not necessarily authenticate oneself to the PKG to obtain the private key since the value sent by the PKG, i.e. $s_A Q_A$, does not contain any information on the private key of the user. The information sent by the PKG becomes useful only if it is received by the intended user.

Considering the difficulty of initial identification of users during the registration as pointed out in [4], we assume that the user is able to prove her identity to the RA during the registration protocol. It could be the case where the user personally goes to the RA and show a piece of identification to prove her identity. Any further elaboration on this issue is beyond the scope of this paper.

B. Non-Repudiation

Non-repudiation, by which a user cannot deny her own transactions with the entities in the system, is a property almost non-existent in IBE systems. Our infrastructure provides the non-repudiation property under non-collusion assumption. Since a user's share of the master secret serves also her private key in her interaction with the PKG, such as identification protocol, she can be held responsible for protecting her share from compromise as in the case of private key in conventional public key cryptosystems. Therefore, IBE signature scheme provides the non-repudiation.

C. Validity Period of Public Keys

Another issue in the proposed infrastructure is the validity duration of users' public keys. As mentioned earlier, we propose to append date information to IDs of the users. The issue then becomes what sort of date information to use in the IDs. Our approach is to define the duration, depending on the application and the underlying communication model used in the message exchange. For instance, we propose to append day information to the IDs in instant messaging applications where users must be on-line and the communication is transient. The user acquires the PKG's part of her private key in the first login in that day and it computes her private key, which expires next day.

For asynchronous messaging systems such as e-mail, where users are most of the time off-line, we propose to use either date of current week or month information appended to users ID. We believe that to change the public key of the user every week does not constitute too much overhead in e-mail applications. Considering many e-mail messages an average user receives in a week, storing PKG's share of user's private key (a point on the underlying elliptic curve) in the same directory as the e-mails received in that week only marginally increases the storage requirements allocated for that user. Once the user connects to the exchange server for the first time in a week it downloads PKG's share of her private key for that week. If the user has not connected to the mail server for more than a week, the user downloads PKG's share of belonging private key for previous weeks as well. Note that these downloading operations are done transparently to the user or rather it is pushed to the user by the mail server. By adding the user's own share to the PKG's share, the user obtains the private key for the current week and will be able to decrypt any message received that week. Note that when the need arises, for instance by an explicit request from the user, PKG can re-generate its share of any user's private key.

Our infrastructure can easily accommodate role-based messaging applications as proposed in [11]. Instead of using names, e-mail addresses, pseudonyms, any description for a role or time and space constraints can be used as a public key in our infrastructure.

D. Key Revocation Problem

In the proposed IBE scheme, revocation becomes an issue in two different circumstances: i) a particular time-dependent private key, e.g. sQ_A, or ii) secret share of any particular user, e.g. s_A, is compromised. When the former happens, the adversary can decrypt the messages intended for the corresponding user or sign messages on behalf of that user until the expiration date of the corresponding public key. This is the reason why we would like to use frequently changing public keys. The shorter the validity period of a public key, the less likely the corresponding private key being fallen in the hands of an adversary assuming that capturing a private key requires substantial efforts. In order to guarantee that no compromised key is used in encryption or signature verification operations, the PKG can publish a (revocation) list of compromised keys. Compromised public keys can be extended with a known public information to generate a new public/private key pair.

If a user compromise her share of the master secret, which we believe is less likely than the former case, the situation must be handled in a different way. The adversary that has the secret share can impersonate the corresponding user, decrypt any messages intended for the user and sign messages on behalf the user. In addition, the adversary can generate a new private key in collaboration with the PKG. Therefore, shortening the validity period does not remedy this situation. In this case, the user must change its share and initiate a new registration phase. With the new share of the master secret, the user implicitly invalidate the old one, with which the adversary cannot extract the private keys. There is no need to keep revocation lists since the user does not have to change its public key after the new share is generated. Adversary revealing the compromised share to the PKG will, however, result in loss of non-repudiation property.

V. Using Pseudonyms for Anonymity

The users, for anonymity reasons, may want to use nicknames or so-called pseudonyms in their interaction with other users in the system. In the classical setting of IBE systems, the PKG knows both the public key (identity) and private key of every user; hence the anonymity cannot be achieved.

Our approach is based on a technique we call *blinding* of the pseudonyms. As illustrated in Figure 5, after selecting a pseudonym, Q_{PN}, Alice blinds it by performing elliptic curve scalar multiplication, kQ_{PN}, where k is the randomly selected blinding factor. The resulting blinded point Q_{BL} is then sent to the PKG that computes $s_A Q_{BL}$ and sends it back to Alice. Alice, finally, computes $k^{-1}(s_A Q_{BL}) + s_A Q_{PN} = sQ_{PN}$. Consequently, Alice declares Q_{PN} (or more specifically PN) as her public key and uses $D_{PN} = sQ_{PN}$ as her private key.

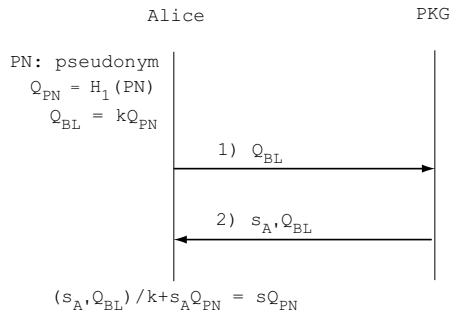

Fig. 5. Pseudonym generation

Another issue with the anonymity is the uniqueness of chosen pseudonyms. As pointed out in [4], having two users sharing the same pseudonym will result in the loss of security and privacy. Thus, users should check whether it is available before they adopt a pseudonym. One solution to this problem is that the RA publishes an authentic list of used pseudonyms. The users check the pseudonym against this list and notify the RA that the chosen pseudonym is no longer available if it is not in the list. The RA, in turn, updates the list of used pseudonyms.

VI. Some Implementation Issues

Our infrastructure is built upon a supersingular elliptic curve with the equation $y^2 = x^3 + 1 \ mod \ p$ where p has a size of 512 bits so as to provide an equivalent security to 1024-bit RSA. For the implementation, we utilize the elliptic curve and pairing classes of MIRACL library [9] which has been developed by Shamus Software Limited. In addition, Intel Celeron 1.5 Ghz computer is used as a base platform together with its Windows XP operating system. Table 1[1] features the execution times for the cryptographic operations in different protocols for each party, namely PKG, RA and the user. The numbers indicated below each party, show the execution times of the corresponding process, relevantly in terms of milliseconds. Clearly, our infrastructure not only offers a secure infrastructure but also provides an efficient system with high execution performance.

TABLE I
The Performance

Process	PKG(ms)	RA(ms)	User(ms)
Computing P_{SYS}	16	17	-
Registration	242	143	140
Private Key Extraction	16	-	20
Pseudonym Generation	16	-	60

IBE systems, as pointed out earlier, are convenient for messaging applications. Therefore, we aim to integrate this proposed infrastructure with an electronic mail application,

[1]Note that the communication latencies are excluded. Table 1 is constructed by running user side only one time, considering the latency requirement on user side; and both PKG and RA 100 times, since throughput is a concern.

such as Firefox Thunderbird. The architecture is built on four entities; the mail server, PKG, RA and the user. Mail server plays the central role and generally is responsible for the transfer and retrieval of mails for the user. For registration and setup phases, user interacts both with PKG and RA. In addition, mail server works as a bridge between RA and PKG, especially for the phase where both parties participate in generation of the system public key.

VII. Conclusion and Future Works

In this paper, we proposed a new IBE infrastructure that is intended for utilization in messaging applications. The proposed infrastructure aims to solve some inherent drawbacks of the IBE systems while retaining their advantage. Key escrowing problem is solved by a method where users and the private key generator secret shares the master secret key. The omniscient private key generator in classical IBE systems which knows all private keys is replaced by a semi-honest third party that does not have information about these private keys. In the presence of the semi-honest private key generator, it is possible to have anonymous communication and non-repudiation property under the non-collusion assumption. We implemented the cryptographic protocols used in the proposed infrastructure and demonstrated that computational requirements for the parties are acceptable. We are, currently, in the process of integrating an e-mail system with the proposed infrastructure. As a future work, we aim to explore the hierarchical pseudonym management protocol within the proposed infrastructure.

VIII. Acknowledgment

The work described in this paper is supported by the Scientific and Technological Research Council of Turkey under project number 105E089.

References

[1] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA*, volume 2139, page 213, CA, USA, 2001. Springer Berlin / Heidelberg.

[2] L. Chen, K. Harrison, D. Soldera, and N. Smart. Applications of multiple trust authorities in pairing based cryptosystems. Hewlett-Packard Trusted Systems Laboratory, HPL-2003-17, February 2003.

[3] H. Cohen and G. Frey. *Handbook of elliptic and hyperelliptic curve cryptography*. Chapman and Hall/CRC, 2006.

[4] Y. Desmedt and M. Burmester. *Identity-Based Key Infrastructures*, page 167. IFIP International Federation for Information Processing. Springer, 2004.

[5] R. Dutta, R. Barua, and P. Sarkar. Pairing based cryptography: A survey. 2004.

[6] G. Frey and H.-G. Ruck. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, 1994.

[7] Oded Goldreich. Secure multi-party computation. Working Draft, 2000.

[8] N. Koblitz. *A Course in Number Theory and Cryptography*. Springer-Verlag, 1994.

[9] Shamus Software LTD. Miracl. http://www.shamus.ie, 2005.

[10] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curves logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[11] M. C. Mont, P. Bramhall, C. R. Dalton, and K. Harrison. A ¤exible role-based secure messaging service: Exploiting ibe technology in a health care trial. Hewlett-Packard Trusted Systems Laboratory, HPL-2003-21, February 2003.

[12] L Owens, A. Duffy, and T. Dowling. An identity based encryption system. In *ACM International Conference Proceeding Series; Vol. 91, Las Vegas, Nevada, USA*, volume 2139, pages 154–159, Las Vegas, Nevada, USA, 2004. Trinity College Dublin.

[13] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, LNCS Vol. 1592*, pages 223–238, Las Vegas, Nevada, USA, 1999. Springer.

[14] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology, Santa Barbara, California, USA*, pages 47–53, CA, USA, 1985. Springer-Verlag New York, Inc.