Computationally Efficient ID-Based Blind Signature Scheme in E-Voting

Rachit T. Jain[1] Axaykumar A. Patel[2]
[1, 2] PG-ITSNS Student
[1, 2] Department of Computer Engineering
[1, 2]Gujarat Technological University, Gujarat, India

Abstract—Blind signatures introduced by Chaum, allow a user to obtain a signature on a message without revealing anything about the message to the signer. Blind signatures play an important role in plenty of applications such as e-voting, e-cash system where anonymity is of great concern. ID based public key cryptography can be a good alternative for certificate based public key setting, especially when efficient key management and moderate security are required. In this we propose an ID based blind signature scheme from bilinear pairings.

Keywords: Identity (ID) based cryptography

I. INTRODUCTION

Digital signature is a cryptographic tool to authenticate electronic communications. Digital signature scheme allows a user with a public key and a corresponding private key to sign a document in such a way that anyone can verify the signature on the document (using her/his public key), but no one can forge the signature on any other document. This self-authentication is required for some applications of digital signatures such as certification by some authority.

The concept of a blind signature scheme was introduced by Chaum [10] since then many blind signature schemes have been presented. Blind signature scheme allows a user to acquire a signature from the signer without revealing message content for personal privacy. The basic idea is as follows.

The user chooses some random factors and embeds them into the message to be signed, while

The signer cannot recover the message. Using the blind signature scheme, the user gets the blinded signature and removes the random factors. Then the user outputs a valid signature. This property is very important for implementing e-voting, e-commerce, and e-payment systems, etc.

In public key cryptosystem, each user has two keys, a private key and a public key. The binding between the public key (PK) and the identity (ID) of a user is obtained via a digital certificate. However, in certificate based system before using the public-key of a user, the participant must first verify the certificate of the user. As a consequence, this system requires a large amount of computing time and storage when the number of users increases rapidly.

In 1984, Shamir introduced the concept of ID-based cryptography to simplify key management procedures in public key infrastructures.

In public key cryptosystems, Boneh and Franklin [11] proposed the first practical ID-based encryption scheme in Crypto 2001. Since then, many ID-based encryption and signature schemes have been proposed that use bilinear pairings.

ID-based cryptography helps us to simplify the key management process in traditional public key infrastructures. In ID-based cryptography any public information such as e-mail address, name, etc., can be used as a public key. Since public keys are derived from publicly known information, their authenticity is established inherently and there is no need for certificates in ID based cryptography.

The private key for a given public key is generated by a trusted authority and is sent to the user over a secure channel.

II. BACKGROUND CONCEPT-BILINEAR PAIRINGS IN CRYPTOGRAPHY

Let $G1,G2$ be additive groups and Gt a multiplicative group, all of prime order P. Where P belongs to $G1$ and Q belongs to $G2$ be generators of $G1$ and $G2$ respectively.

A pairing is a map $e: G1\times G2\text{->}Gt$ for which the following holds:

1) Bilinearity:

$e\ (P1+P2;\ Q) = e\ (P1;\ Q)\ e\ (P2;\ Q)$ *and*
$e\ (P;\ Q1+Q2) = e\ (P;\ Q1)\ e\ (P;\ Q2)$ *or*
$e\ (aP;\ bQ) = e\ (P;\ Q)\ ab;$

2) Non-degeneracy: $e(P,Q)\neq 1$
3) For practical purposes, *e* has to be computable in an efficient manner.

III. ID-BASED CRYPTOGRAPHY

A Private-key Generator (PKG). The PKG contains the cryptographic material, known as a master secret, for generating an individual’s IBE private key. A PKG accepts an IBE user’s private key request, and after successfully authenticating them in some way, returns the IBE private key.

A Public Parameter Server (PPS). IBE System Parameters include publicly sharable cryptographic material, known as IBE public parameters, and policy information for the PKG. A PPS provides a well-known location for secure distribution of IBE public parameters and policy information for the IBE PKG.

A logical architecture would be to have a PKG/PPS per name space, such as a DNS zone. The organization that controls the DNS zone would also control the PKG/PPS and thus the determination of which PKG/PSS to use when creating public and private keys for the organization’s members. In this case the PPS URI can be uniquely created by the form of the identity that it supports. This architecture would make it clear which set of public parameters to use and where to retrieve them for a given identity. IBE-encrypted messages can use standard message formats, such as the Cryptographic Message Syntax. Note that IBE algorithms are used only for encryption, so if digital signatures are required, they will need to be provided by an additional mechanism.

Sending a Message That Is Encrypted Using IBE In order to send an encrypted message, an IBE user must perform the following steps:

1) Obtain the recipient's public parameters. The recipient's IBE public parameters allow the creation of unique public and private keys. A user of an IBE system is capable of calculating the public key of a recipient after he obtains the public parameters for their IBE system. Once the public parameters are obtained, IBE-encrypted messages can be sent.

2) Construct and send an IBE-encrypted message. All that is needed, in addition to the IBE public parameters, is the recipient's identity in order to generate their public key for use in encrypting messages to them. When this identity is the same as the identity that a message would be addressed to, then no more information is needed from a user to send someone a secure message than is needed to send them an unsecured message. This is one of the major benefits of an IBE-based secure messaging system. Examples of identities can be an individual, group, or role identifiers.

Sender Obtains Recipient's Public Parameters

The sender of a message obtains the IBE public parameters that he needs for calculating the IBE public key of the recipient from a PPS that is hosted at a well-known URI. The IBE public parameters contain all of the information that the sender needs to create an IBE-encrypted message except for the identity of the recipient. [13] describes the URI where a PPS is located, the format of IBE public parameters, and how to obtain them. The URI from which users obtain IBE public parameters MUST be authenticated in some way; PPS servers MUST support Transport Layer Security (TLS) 1.1 [15] to satisfy this requirement and MUST verify that the subject name in the server certificate matches the URI of the PPS. [13] also describes the way in which identity formats are defined and a minimum interoperable format that all PPSs and PKGs MUST support. This step is shown below in Figure 1

```
IBE Public Parameter Request
----------------------------->
Sender PPS
<-----------------------------
IBE Public Parameters
```

Fig. 1-Requesting IBE Public Parameters

The sender of an IBE-encrypted message selects the PPS and corresponding PKG based on his local security policy. Different PPSs may provide public parameters that specify different IBE algorithms or different key strengths, for example, or require the use of PKGs that require different levels of authentication before granting IBE private keys.

Construct and Send an IBE-Encrypted Message

To IBE-encrypt a message, the sender chooses a content encryption key (CEK) and uses it to encrypt his message and then encrypts the CEK with the recipient's IBE public key (for example, as described in [12]). This operation is shown below in Figure 2 This document describes the algorithms needed to implement two forms of IBE. [14] describes how to use the Cryptographic Message Syntax (CMS) to encapsulate the encrypted message along with the IBE information that the recipient needs to decrypt the message.

```
CEK ----> Sender ----> IBE-encrypted CEK
                              /\
                              |
                              |
  Recipient's Identity and IBE Public Parameter
```

Fig. 2: Using an IBE Public-Key Algorithm to Encrypt

Receiving and Viewing an IBE-Encrypted Message

In order to read an encrypted message, a recipient of an IBE-encrypted message parses the message (for example, as described in [14]). This gives him the URI he needs to obtain the IBE public parameters required to perform IBE calculations as well as the identity that was used to encrypt the message. Next, the recipient must carry out the following steps:

1) Obtain the recipient's public parameters:

An IBE system's public parameters allow it to uniquely create public and private keys. The recipient of an IBE-encrypted message can decrypt an IBE-encrypted message if he has both the IBE public parameters and the necessary IBE private key. The PPS can also provide the URI of the PKG where the recipient of an IBE-encrypted message can obtain the IBE private keys.

2) Obtain the IBE private key from the PKG:

To decrypt an IBE-encrypted message, in addition to the IBE public parameters, the recipient needs to obtain the private key that corresponds to the public key that the sender used. The IBE private key is obtained after successfully authenticating to a private key generator (PKG), a trusted third party that calculates private keys for users. The recipient receives the IBE private key over an HTTPS connection. The URI of a PKG MUST be authenticated in some way; PKG servers MUST support TLS 1.1 [15] to satisfy this requirement.

3) Decrypt the IBE-encrypted message:

The IBE private key decrypts the CEK, which is then used to decrypt encrypted message.

The PKG may allow users other than the intended recipient to receive some IBE private keys. Giving a mail filtering appliance permission to obtain IBE private keys on behalf of users, for example, can allow the appliance to decrypt and scan encrypted messages for viruses or other malicious features.

Recipient Obtains Public Parameters from PPS

Before he can perform any IBE calculations related to the message that he has received, the recipient of an IBE-encrypted message needs to obtain the IBE public parameters that were used in the encryption operation. This operation is shown below in Figure

IBE Public Parameter Request
---------------------------->
Recipient PPS
<----------------------------
IBE Public Parameters

Fig. 3: Requesting IBE Public Parameters Recipient Obtains IBE Private Key from PKG

To obtain an IBE private key, the recipient of an IBE-encrypted message provides the IBE public key used to encrypt the message and their authentication credentials to a PKG and requests the private key that corresponds to the IBE public key. PKGs MUST support TLS [15] for transport of IBE private keys. This operation is shown below in Figure

IBE Private Key Request
--------------------------->
Recipient PKG
<---------------------------
IBE Private Key

Fig. 4: Obtaining an IBE Private Key Recipient Decrypts IBE-Encrypted Message

After obtaining the necessary IBE private key, the recipient uses that IBE private key, and the corresponding IBE public parameters, to decrypt the CEK. This operation is shown below in Figure He then uses the CEK to decrypt the encrypted message content (for example, as specified in [14]).

```
IBE-encrypted CEK ----> Recipient ----> CEK
                            /\
                            |
                            |
  IBE Private Key and IBE Public Parameters
```

Fig. 5: Using an IBE Public-Key Algorithm to Decrypt

IV. PROPOSED SCHEME E-VOTING

1) *Phase I: Key Generation:*

– *Setup:*

The PKG chooses $S \in Z*q$ as his master key and computes the global public key Ppub as The PKG also selects a map-to-point hash function *H1* :{0,1}→ *G1** and another cryptographic hash function h:{0,1}* $\times G2 \rightarrow Z*q$ PKG publishes system parameters params <G1,G2,e,p,Ppub,H1,h> and keeps the master key as secret.

– Extract:

Given signer's public identity ID{0,1}* compute the public key QID=H1(ID) and the corresponding private key Did= sQID   .

– Initialization:

The signer randomly chooses k∈z*q compute and sends R=e(P,P)k and sends R to the user a commitment.

2) *Phase II: Blinding:*

Here, the voter elects the vote (message). As the votes of the individuals should be kept confidential the votes (message) are blinded. A blinding factor is selected and the vote (message) is then treated with this blinding factor to blind the vote that is to hide the vote from others. One thing to note is that the blinding factor chosed should possess an existing inverse of itself so that the message blinded could also be unblinded when required.

The user randomly chooses a,b∈ Z*q as blinding factors, compute R'=e(bQID+ aP,Ppub)R   and sends V= h(m,R')+b to the signer.

3) *Phase III: Requester Phase:*

In this phase, the voter generates a digital signature using his private key. The voter then sends in entire four entities to the signer as a request for authentication. The entities comprise of identification details, blinded message computed in phase II, digital signature and a proving factor that proves the voter to be a valid citizen. Here, the factor that proves the voter to be a valid citizen. A valid citizen possesses a private key to oneself but to prove oneself to be a valid citizen one cannot reveal the private key as it is to be kept confidential or intruder may misuse it.

4) *Phase IV: Signing Phase:*

In this phase, the signer initially will have the incoming request from the voter with four entities. After receiving the request message the signer verifies for two matters. First, whether requester is a valid voter or not and this is done by cross verifying the proving factor. Secondly, signer notes the identification details and checks whether requester has already voted or not. In other words, signer verify for the actuality of the user applying the voter's (requester) public key and also for the redundancy of voter. If the requester through both the matters the signer generates blind signature for the particular requester and authenticates the voter. The signer then replies the requester with message – signature pair. The signer displays the identification details and the public keys of the voters those whose have voted. In this way all the voters they get authenticated without revealing any secret information of them that is zero knowledge proof.

Signing: The signer computes s= VDid+kP *and* send S to the user.

5) *Phase V: Unblinding:*

Voter after receiving the message - signature pair, the message is unblinded and the unblinded message – signature pair is sent to the voting centre acting as a verifier and the counter of the votes. Here, the message is unblinded as when the message – signature pair is sent to the voting centre the counter must know to whom the voter has voted to be able to count the number of vote for individual elective.

Unblindig: The user compute S'=S+aPpub, V'=V-b and outputs (m, S', V'), then (S', V') is the blind signature of the message m.

6) *Phase: VI: Verification:*

Verifier after receiving the unblinded message – signature verifies the signer's blind digital signature using the public key of the signer. As the signature is verified the count is incremented for elective that is voted. Verifier now displays all the digital signatures and blind digital signatures pairs. Hence the voter is ensured that his/her vote is counted. And no would come to know who voted to whom because only voter know about his own digital signature and blind digital signature received from signer. The voter after choosing the vote blinds it as the signer should not be able to know to whom the voter has voted so the voter's vote remains confidential. Next, signer signs the blinded message and hence the blind digital signature. Now when the blind digital signature – message pair is received by the voter, the message is unblinded. This unblended message along with blind digital signature is sent to the verifier so that the verifier would see to whom the voter has voted for and update the counters Verification: Accept the signature if and only if V'=h (m, e(S', P) e (QID, Ppub) v')

V. CONCLUSION

The ID-based public key setting can be an alternative for certificate-based public key setting, when efficient key management and moderate security are required in particular. so no need of certificate authority. In this , we proposed an ID-based blind signature scheme using the bilinear pairing. ID-based signature scheme can be easily combined to design electronic voting scheme or electronic cash scheme.

REFERENCES

[1]. F. Zhang and K. Kim, "ID-based blind signature and ring signature from pairings", Proc. of Asiacrpt 2002, LNCS 2501, pp. 533-547, Springer-Verlag, 2002.

[2]. D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind Signatures", Journal of Cryptology, Vol.13, No.3, pp.361-396, 2000.4. Secure Identity-Based Blind Signature Scheme in the Standard Model* by XIAO-MING HU AND SHANG-TENG HUANG

[3]. S. Kalkan, K. Kaya, and A. A. Selcuk. Generalized ID-based ElGamal signatures. In The 22nd International Symposium on Computer and Information Sciences (ISCIS 2007), 2007.

[4]. F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In Proc. of ACISP2003, volume 2727 of LNCS, pages 312–323, 2003.

[5]. Z. Huang, K. Chen, and Y. Wang. Efficient identity-based signatures and blind signatures. In Proc. of CANS 2005, volume 3810 of LNCS, 2005

[6]. Dennis Effort. "Bilinear Pairings in Cryptography" Journal of Cryptology: the journal of the International Association for Cryptologic Research 12 (1999), no. 3, 193{196.

[7]. Song Han, and Elizabeth Chang " A Pairing-based Blind Signature Scheme with Message Recovery" International Journal of Information and Communication Engineering 2:7 2006

[8]. Garran Kumar Verma "New ID-based Fair Blind Signatures" International Journal of Computer Science and Security Volume (3): Issue (2).

[9]. Xiaofeng Chen1, Fangguo Zhang2 and Shingle Liu3 "Proc. of Asiacrpt 2004, LNCS 2501, pp. 241-265, Springer-Verlag, 2004.

[10]. D. Chaum, "Blind signatures for untraceable payments," in *Proceedings of Crypto*, 1983, pp. 199-203.

[11]. Boneh and Franklin (2001), "Identity Based Encryption from the Weil Pairing"

[12]. Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[13]. G. Appenzeller, L. Martin, and M. Schertler, "Identitybased Encryption Architecture", Work in Progress.

[14]. L. Martin and M. Schertler, "Using the Boneh-Franklin and Boneh-Boyen identity-based encryption algorithms with the Cryptographic Message Syntax (CMS)", Work in Progress.

[15]. Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.