Proxy Signature Scheme with Effective Revocation Using Bilinear Pairings

Manik Lal Das[1,2], Ashutosh Saxena[1], and Deepak B. Phatak[2]
(Corresponding author: Manik Lal Das)

Institute for Development and Research in Banking Technology[1]
Castle Hills, Masab Tank, Hyderabad-500057, India
Email: mdas@it.iitb.ac.in
K. R. School of Information Technology[2]
Indian Institute of Technology, Bombay, Mumbai-400076, India

(Received Oct. 07, 2005; revised and accepted Nov. 25, 2005 & Jan. 27, 2006)

Abstract

We present a proxy signature scheme using bilinear pairings that provides effective proxy revocation. The scheme uses a binding-blinding technique to avoid secure channel requirements in the key issuance stage. With this technique, the signer receives a partial private key from a trusted authority and unblinds it to get his private key, in turn, overcomes the key escrow problem which is a constraint in most of the pairing-based proxy signature schemes. The scheme fulfills the necessary security requirements of proxy signature and resists other possible threats.

Keywords: Bilinear pairings, key escrow, proxy revocation, proxy signature

1 Introduction

Proxy signature is a digital signature where an original signer delegates his signing capability to a proxy signer, and then the proxy signer performs message signing on behalf of the original signer. The notion of proxy signature has been evolved over a long time, 16 years now [5]. However, the cryptographic treatment on proxy signature was introduced by Mambo et al [9] in 1996. They classified the delegation capability in three types, namely *full delegation*, *partial delegation* and *delegation by warrant*. In full delegation, an original signer directly gives his private key to a proxy signer and using it the proxy signer signs the document. The drawback of proxy signature with full delegation is that the absence of a distinguishability between the original signer and the proxy signer. In partial delegation, the original signer derives a proxy key from his private key and hands it over to the proxy signer as a delegation of signing rights. In this case, the proxy signer can misuse the delegation of signing rights, because partial delegation does not restrict the proxy signer's signing capability. The weakness of full and partial delegations are eliminated by partial delegation with warrant, where a warrant explicitly states the signers' identity, delegation period and the qualification of the message on which the proxy signer can sign, etc. Once proxy delegation is given, the revocation is an important issue in the proxy signature scheme. For instance, the original signer key is compromised or any misuse of delegation of signing rights is noticed. It may so happen that the original signer wants to terminate his delegation power before the expiry e.g., the manager of a company has come back from his trip before time that he was scheduled for.

Desirable security properties of proxy signatures have evolved over this period and a widely accepted list of required properties are as follows:

- Strong unforgeability: A designated proxy signer can create a valid proxy signature on behalf of the original signer. But the original signer and other third parties cannot create a valid proxy signature.
- Strong identifiability: Anyone can determine the identity of the corresponding proxy signer from the proxy signature.
- Verifiability: The verifier can be convinced of the original signer's agreement from the proxy signature.
- Distinguishability: Proxy signatures are distinguishable from normal signatures by everyone.
- Strong undeniability: Once a proxy signer creates a valid proxy signature, he cannot deny the signature creation.
- Prevention of misuse: The proxy signer cannot use the proxy key for other purposes than it is made for. That is, he cannot sign message with the proxy key that have not been defined in the warrant. If he does so, he will be identified explicitly from the warrant.

After Mambo et al.'s scheme [9], several schemes have been proposed [1, 7, 8, 10, 11]. However, most of the schemes lack proxy revocation mechanism. Recently, the bilinear pairings, namely the Weil pairing and the Tate pairing of algebraic curves have been found important applications [2, 4, 6] in identity (ID) based cryptography. The advantage of an ID-based cryptography [12] is that it avoids public key certification, the public key of a user is his identity, e.g., e-mail, social security number, etc. There are a few proxy signature schemes [3, 13, 14, 15] based on bilinear pairings; however, the schemes lack the key escrow problem and have not addressed the proxy revocation mechanism. In this paper, we present a proxy signature scheme using bilinear pairings that provides effective proxy revocation mechanism. Our scheme is not exactly ID-based, it is a variant of ID-based schemes. The scheme does not require secure channel in the key issuance stage and avoids the key escrow problem.

The rest of the paper is organized as follows. Section 2 discusses some preliminaries. Section 3 presents the scheme. Section 4 analyzes the security and performance of the scheme. Finally, we conclude the paper in Section 5.

2 Preliminaries

2.1 Bilinear Pairings

Suppose G_1 is a cyclic additive group of prime order q, generated by P, and G_2 is a cyclic multiplicative group of the same order q. A map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

- Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in \mathbb{Z}_q^*$;
- Non-degenerate: There exist $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$;
- Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)\ \forall\ P, Q \in G_1$.

In general, G_1 is a group of points on an elliptic curve and G_2 is a multiplicative subgroup of a finite field.

2.2 Computational Problems

Definition 1. *Discrete Logarithm Problem (DLP): Given $Q, R \in G_1$, find an integer $a \in \mathbb{Z}_q^*$ such that $R = aQ$.*

Definition 2. *Decisional Diffie-Hellman Problem (DDHP): Given (P, aP, bP, cP) for $a, b, c \in \mathbb{Z}_q^*$, determine whether $c \equiv ab \bmod q$. The advantage* `Adv` *of any probabilistic polynomial-time algorithm $\mathcal{A}$ in solving DDHP in G_1 is defined as:*

$$\begin{aligned} \mathtt{Adv}_{\mathcal{A},G_1}^{DDH} &= [\mathtt{Pr}[\mathcal{A}(P, aP, bP, cP) = 1] \\ &- \mathtt{Pr}[\mathcal{A}(P, aP, bP, abP) = 1] : a, b, c \in \mathbb{Z}_q^*]. \end{aligned}$$

For every probabilistic polynomial-time algorithm $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A},G_1}^{DDH}$ is negligible.

Definition 3. *Computational Diffie-Hellman Problem (CDHP): Given (P, aP, bP) for $a, b \in \mathbb{Z}_q^*$, compute abP. The advantage of any probabilistic polynomial-time algorithm $\mathcal{A}$ in solving CDHP in G_1 is defined as:*

$$\mathtt{Adv}_{\mathcal{A},G_1}^{CDH} = \left[\mathtt{Pr}[\mathcal{A}(P, aP, bP, abP) = 1 : a, b \in \mathbb{Z}_q^*\right].$$

For every probabilistic algorithm $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A},G_1}^{CDH}$ is negligible.

Definition 4. *Gap Diffie-Hellman Problem (GDHP): A class of problems where DDHP is easy while CDHP is hard.*

Definition 5. *Weak Diffie-Hellman Problem (WDHP): Given (P, Q, aP) for $a \in Z_q^*$, compute aQ.*

3 The Proposed Scheme

To avoid the original signer's forgery and prevention of delegation power misuse, the proxy-protected proxy signature [8] is a secure approach. Our scheme is based on proxy-protected notion and uses the merits of partial delegation with warrant[1].

The participating entities and their roles in the proposed scheme are defined as follows:

- Private Key Generator (PKG): A trusted authority who receives signer's identity (ID) along with other parameters, checks validity of ID and issues partial private key to the signer corresponding to the ID.
- Original Signer: Entity who delegates his signing rights to a proxy signer.
- Proxy Signer: Entity who signs the message on behalf of the original signer.
- Verifier: Entity who verifies the proxy signature and decides to accept or reject.

The scheme has five phases: Setup, KeyGen, ProxyKeyGen, ProxySignGen and ProxySignVerify. The phases work as follows.

Setup Phase:
It takes as input a security parameter; and outputs system parameters *params* and master-key of PKG. The *params* includes a cyclic additive group G_1 of prime order q generated by P, a cyclic multiplicative group G_2 of prime order q, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, hash functions $H_1 : \{0,1\}^* \times G_1 \times G_1 \rightarrow G_1$, $H_2 : \{0,1\}^* \rightarrow G_1$, $h : \{0,1\}^* \times G_1 \times G_1 \rightarrow \mathbb{Z}_q^*$, and public key of PKG. The PKG selects a master-key $s \in \mathbb{Z}_q^*$ and computes public key as $Pub_{PKG} = sP$. The PKG publishes $params = (G_1, G_2, \hat{e}, q, P, Pub_{PKG}, H_1, H_2, h)$ and keeps s secret.

KeyGen Phase:
It takes user chosen parameters and *params* as inputs;

---

[1] A warrant consists of original signer and proxy signer identities, qualification of the message on which the proxy signer can sign, validity period of the delegation, etc.

and outputs user private key. The entire phase consists of a *partial private key issuance* and a *private key generation* stages. The stages use a binding-blinding technique to avoid the key escrow problem and to eliminate the secure channel requirements. The binding-blinding technique works as follows:

- The user chooses two secret binding factors, calculates the binding parameters and sends them to the PKG over a public channel along with his/her identity.
- As the communication channel between the user and the PKG is a public channel, a dishonest party can construct his/her preferred binding parameters using the targeted user's identity and sends the binding parameters along with user's identity before the user submits a request for partial private key. To avoid this type of attack, the PKG first sends a message to the email-id[2] (email-id acts as the user identity) and asks a confirmation from the email-id owner. If the email-id owner confirms his/her request for a partial private key, then the PKG proceeds to the next step.
- The PKG checks the validity of binding parameters. Upon successful validation of the parameters, the PKG computes signer partial private key. Then, the PKG sends the partial private key to the user in a blinding manner over the public channel.

`PartialPrivateKey` issuance:

- User U_λ computes his own public key $Pub_\lambda = H_2(ID_\lambda)$.
- U_λ picks two secret binding factors $a_\lambda, b_\lambda \in \mathbb{Z}_q^*$ and computes $X_\lambda = a_\lambda Pub_\lambda$, $Y_\lambda = a_\lambda b_\lambda Pub_\lambda$, $Z_\lambda = b_\lambda P$ and $W_\lambda = a_\lambda b_\lambda P$. Then he sends ($X_\lambda$, Y_λ, Z_λ, W_λ, ID_λ) to the PKG over a public channel.
- Once the ID_λ is correct (we assume that identity of the user is his/her email-id and unregistered identity attack can be avoided by the above mentioned email confirmation procedure), the PKG computes $Pub_\lambda = H_2(ID_\lambda)$ and verifies the validity of ID_λ by whether $\hat{e}(Y_\lambda, P) = \hat{e}(X_\lambda, Z_\lambda) = \hat{e}(Pub_\lambda, W_\lambda)$.
- The PKG computes U_λ's partial private key as $D_\lambda = sY_\lambda$ and creates a *registration-token* $Reg_\lambda = sZ_\lambda$ corresponding to ID_λ. Then, PKG publishes $(Reg_\lambda, ID_\lambda)$ in a public directory and sends D_λ to U_λ over a public channel.

We note that the PKG controls the public directory and checks every request before issuance of any partial private key. If the identity is present in the directory, the PKG denies the request, thereby the registration-token replacement is not possible by any other party.

`PrivateKey` Generation:

- On receiving the partial private key D_λ, the signer U_λ checks its correctness by whether $\hat{e}(D_\lambda, P) = \hat{e}(Y_\lambda, Pub_{PKG})$. If D_λ is valid, U_λ unblinds it and generates his private key as $S_\lambda = a_\lambda^{-1} D_\lambda$.

Original Signer Private Key: Let ID_o be the identity of an original signer. The original signer chooses binding secret factors a_o and b_o and runs the `KeyGen` algorithm to get his partial private key as

$$D_o \leftarrow \texttt{PartialPrivateKey}(X_o, Y_o, Z_o, W_o, ID_o).$$

After validating D_o, the original signer generates his private key as $S_o = a_o^{-1} D_o$.

Proxy Signer Private Key: Let ID_p be the identity of the proxy signer. The proxy signer chooses the binding factors a_p and b_p and runs the `KeyGen` algorithm to get his partial private key as

$$D_p \leftarrow \texttt{PartialPrivateKey}(X_p, Y_p, Z_p, W_p, ID_p).$$

After validating D_p, the proxy signer generates his private key as $S_p = a_p^{-1} D_p$.

`ProxyKeyGen` Phase:

- The original signer and proxy signer agree on a warrant m_w.
- The original signer computes $U_o = S_o + b_o H_1(m_w, Pub_o, Pub_p)$, $\psi_o = b_o P$ and sends the tuple $(m_w, U_o, \psi_o, Pub_o)$ to the proxy signer over a public channel as the delegation capability.
- The proxy signer checks whether
$$\begin{aligned} & \hat{e}(U_o, P) \\ = \; & \hat{e}(\psi_o, H_1(m_w, Pub_o, Pub_p))\hat{e}(Pub_o, Reg_o). \end{aligned}$$
- If the delegation capability is valid, the proxy signer computes proxy key as
$$V_p = U_o + S_p + b_p H_1(m_w, Pub_o, Pub_p).$$

`ProxySignGen` Phase:
To sign a message m, the proxy signer computes the following steps:

- Select a random $r \in \mathbb{Z}_q^*$ and computes $R = rP$.
- Compute $a = h(m, R, Pub_p)$ and $\psi_p = b_p P$.
- Compute $V = (r + a)^{-1} V_p$.

The proxy signature on m is the tuple $(m_w, m, R, V, \psi_o, \psi_p, Pub_o, Pub_p)$.

---

[2] At this juncture, we assume that the email-id acts as the user identity; however, other identity could play the same role if it avoids the unregistered identity attack. We note that it is a difficult task to avoid the unregistered identity attack for any types of identity if there is no off-line (secure channel) interaction between the PKG and the user, in turn it opens a prominent future scope of our proposed work.

`ProxySignVerify` Phase:
The proxy signature $(m_w, m, R, V, \psi_o, \psi_p, Pub_o, Pub_p)$ is valid if and only if

$$\begin{aligned} & \hat{e}(R + h(m, R, Pub_p)P, V) \\ = \ & \hat{e}(\psi_o + \psi_p, H_1(m_w, Pub_o, Pub_p))\hat{e}(Pub_o, Reg_o) \\ & \hat{e}(Pub_p, Reg_p). \end{aligned}$$

4 Analysis of the Scheme

4.1 Correctness of Proxy Signature Verification

$$\begin{aligned} & \hat{e}(R + h(m, R, Pub_p)P, V) \\ = \ & \hat{e}((r + h(m, R, Pub_p))P, (r + a)^{-1}V_p) \\ = \ & \hat{e}((r + a)P, (r + a)^{-1}V_p) \\ = \ & \hat{e}(P, U_o + S_p + b_pH_1(m_w, Pub_o, Pub_p)) \\ = \ & \hat{e}(P, S_p + S_o + (b_p + b_o)H_1(m_w, Pub_o, Pub_p)) \\ = \ & \hat{e}(P, S_p)\hat{e}(P, S_o)\hat{e}(P, (b_p + b_o)H_1(m_w, Pub_o, Pub_p)) \\ = \ & \hat{e}(Pub_o, Reg_o)\hat{e}(Pub_p, Reg_p)\hat{e}((b_o + b_p)P, \\ & H_1(m_w, Pub_o, Pub_p)) \\ = \ & \hat{e}(Pub_o, Reg_o)\hat{e}(Pub_p, Reg_p)\hat{e}(\psi_o + \psi_p, \\ & H_1(m_w, Pub_o, Pub_p)) \end{aligned}$$

4.2 Security Analysis

In this section, we show that the proposed scheme satisfies the security properties of a proxy signature, mentioned in Section 1. In addition, the scheme withstands some other possible threats.

The scheme can withstand the strong unforgeability security property.
To create a valid proxy signature, one should need the original signer and proxy signer private keys. Though the adversary can intercept signer partial private key D_i (i.e., $sa_ib_iPub_i$), he cannot construct the private key S_i (i.e., sb_iPub_i) without the knowledge of a_i, because it is a WDHP (Definition 5) which is assumed to be a hard problem. As our scheme is proxy protected, i.e., the proxy signer has to use his private key and original signer's delegation power to sign a message, thus, the original signer is also prohibited from forging a valid proxy signature. Moreover, the PKG cannot frame the signers' with the knowledge of binding parameters (X_i, Y_i, Z_i, W_i), as extracting the binding factors a_i, b_i from the binding parameters is as hard as CDHP (Definition 3).

The scheme can resist the identifiability, undeniability and distinguishability security properties.
A valid proxy signature of a message m is the tuple (m_w, m, R, V, ψ_o, ψ_p, Pub_o, Pub_p). The public keys Pub_o, Pub_p and warrant m_w are the straightforward witnesses (i.e., identities) of the signers. In addition, a verifier will come to know the agreement between original and proxy signers from m_w.

From the correctness of the proxy signature, given in Section 4.1, it is clear that the proxy signer cannot deny his signature creation. The verification of a valid proxy signature needs the proxy signer's public key, in turn, proves that the signature was created by the proxy signer. Further, the PKG can also prove the identity of the proxy signer, as the tuple (Reg_p, ID_p) in the PKG public directory is a supporting identification of a proxy signer and is also required in the proxy signature verification phase. Any verifier will receive the proxy signature that contains warrant m_w and the public key of signers, by which the verifier can easily distinguish the proxy signature from the normal signature.

The scheme is secure against misuse of the proxy delegation.
In the Proxy key generation phase, the original signer signs the tuple (m_w, Pub_o, Pub_p) and gives it to the proxy signer as his delegation capability. The proxy signer signs a message with the proxy key that is being created by his private key and original signer's delegation capability. The qualification of message and limitation of proxy is clearly defined in m_w and the delegation is made for the designated proxy signer only. If the proxy signer misused the delegation capability, the proxy signer will be detected by any verifier from m_w. The original signer's misuse is also prevented because he cannot create a valid proxy signature against the name of the proxy signer.

Apart from the above security properties, the scheme withstands the following possible threats.

Threat 1. Registration-token replacement: The PKG creates *registration-token* corresponding to each registered signer and publishes it along with signer-ID in a public directory, which is controlled by PKG only. If a request comes from signer identity ID^* for issuance of a partial private key, the PKG first checks whether ID^* is in the public directory. If it is found in the public directory, the PKG rejects the request, otherwise executes the `KeyGen` algorithm for ID^*. Thus, the registration-token replacement is not possible by any party (the PKG itself can replace the registration-token, but we assume that the signer trusts PKG for not to do it).

Threat 2. Man-in-the-middle attacks: In our scheme, the communication channel of the key issuance stage is a public channel, thus an attacker may try to calculate the private key or binding factors of a signer by intercepting the binding parameters and partial private key. On intercepting the binding parameters, the adversary can formulate the following problem: *Given params, binding parameters (a_iPub_i, $a_ib_iPub_i$, b_iP, a_ib_iP, ID_i) and partial private key D_i (i.e., $sa_ib_iPub_i$); Compute private key S_i (i.e. sb_iPub_i) or binding factors (a_i, b_i).* To solve this problem, one has to solve either the CDHP or the WDHP, which is assumed to be computationally hard.

Threat 3. ONE partial private key → MANY private keys: The scenario of generating more than one private key from a partial private key is nor possible, because the private key S_i (i.e. sb_iPub_i) and the registration-token Reg_i are linked by the secret binding factor b_i. If a signer generates another private S_i^* from S_i and signs a message by S_i^*, then the verification of the signature fails because the change from S_i to S_i^* is not reflected in Reg_i. Thereby, the signer cannot perform this type of attempt without being detected.

Theorem 1. *The proxy signature scheme is said to be secure against adaptive chosen-massage attacks under random oracle model if no polynomially bounded adversary (in k) has non-negligible advantage (in k).*

Proof. The proof of the theorem is ascertained by the following challenger-adversary game.

Setup: A challenger $\mathcal{C}$ takes a security parameter k and runs the `Setup` phase as mentioned in Section 3. Then $\mathcal{C}$ returns the resulting system parameters *params* to $\mathcal{A}$ and keeps master-key s with itself.

Queries: The adversary $\mathcal{A}$ issues adaptively the queries $q_1, q_2, \cdots, q_m$ in any order for the following:
`ProxyKeyGen` query on Pub_j, where $j = 1, \cdots, m$:
$\mathcal{C}$ runs the `ProxyKeyGen` phase and generates proxy key V_j using S_j and b_j corresponding to Pub_j, and sends it to $\mathcal{A}$.
`ProxySignGen` query on (Pub_j, M'):
$\mathcal{C}$ runs the `ProxyKeyGen` phase and generates the proxy key V_j. Then, $\mathcal{C}$ signs the message M' and returns the proxy signature $(\omega, M', R', V(M'), \psi_o, \psi_j, Pub_o, Pub_j)$ to $\mathcal{A}$.

Guess: $\mathcal{A}$ outputs a proxy signature for message M^*, where M^* did not appear in the `ProxySignGen` query.

Result: $\mathcal{A}$ wins if his produced proxy signature on M^* is valid. The advantage of $\mathcal{A}$ in attacking the scheme is defined to be the probability that $\mathcal{A}$ produces a valid proxy signature in the game. We say that our scheme is secure against adaptive chosen-message attacks under random oracle model if no polynomially bounded adversary has non-negligible advantage in this game. □

4.3 Performance

Proxy Revocation: The revocation of delegation capability (i.e., proxy revocation) is an important concern in any proxy signature scheme. It is observed that the schemes [3, 13, 14, 15] have not addressed the proxy revocation issues, which is a practical requirement. In our scheme, proxy revocation can be easily done by revoking the registration-token from the PKG's public directory. If the original signer wants to revoke his delegation of signing rights, he sends a revoke-request tuple (M_r, m_w, Rev, Pub_o, Pub_p, ψ_o) to the PKG and proxy signer, where $Rev = S_o + b_oH_1(M_r, Pub_o, Pub_p)$ and M_r states the identity of the signer along with the reason for proxy revocation. The PKG first checks the authenticity and validity of the revoke-request and if the request is valid then PKG revokes the tuple (Reg_o, ID_o) and (Reg_p, ID_p) from the public directory. We note that the proxy signer will not object if the PKG removes (Reg_p, ID_p) without his consent (the original consent is with PKG), because if the delegation capability is no longer authorized, the delegated proxy signer is no longer required. The PKG validates the revoke-request as follows:

$$\begin{aligned}\hat{e}(Rev, P) &= \hat{e}(S_o + b_oH_1(M_r, Pub_o, Pub_p), P)\\ &= \hat{e}(sb_oPub_o + b_oH_1(M_r, Pub_o, Pub_p), P)\\ &= \hat{e}(Reg_o, Pub_o)\hat{e}(H_1(M_r, Pub_o, Pub_p), \psi_o).\end{aligned}$$

Key Escrow: In our scheme, the PKG issues a `PartialPrivateKey` to the signer and with this the signer computes his private key. The PKG is not having knowledge of signer private key. To construct a private key from the partial private key, one has to know the secret binding factor or has to solve DLP. As the binding factor is retained with the signer only, other party can not obtain signer private key because solving DLP is a hard problem. Thus, our scheme avoid the key escrow problem, which occurs in the schemes [3, 13, 14, 15].

No Need of Secure Channel: To eliminate the secure channel in the key issuance stage, we used a binding-blinding technique where the signer requests for a partial private key from the PKG. We considered a simplest procedure to verify the genuineness of signer's identity while partial private key issuance. After validating the signer request, the PKG issues a partial private key in a blinded manner. Finally, the signer unblinds the partial private key to get his private key. This binding-blinding technique avoids the secure channel in the key issuance stage.

5 Conclusion

We proposed a proxy signature scheme using bilinear pairings that provides effective proxy revocation. The scheme uses a binding-blinding technique to eliminate the secure channel requirements in the key issuance stage. We considered a mechanism to avoid the unregistered identity attacks when identity is user's email-id, though the mechanism does not provide a generic solution for other types of identities. We leave this problem as a future scope of the proposed work. Our scheme is not exactly ID-based scheme; however, it avoids the key escrow problem, which remains constraint in most of the existing pairing-based proxy signature schemes. We showed that the scheme satisfied the security requirements of a proxy signature and also withstood other possible threats.

References

[1] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure signature schemes for delegation of signing rights," Availbale at http://eprint.iacr.org/2003/96/

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

[3] X. Chen, F. Zhang, and K. Kim, "ID-based multi-proxy signature and blind multisignature from bilinear pairings," in *Proceedings of Korean Institute of Information Security and Cryptology (KIISC'03)*, pp. 11-19, 2003.

[4] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding*, LNCS 2260, pp. 360-363, Springer-Verlag, 2001.

[5] M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson, "The digital distributed system security architecture," in *Proceedings of National Computer Security Conference*, pp. 305-319, 1989.

[6] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proceedings of Selected Areas in Cryptography (SAC'02)*, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

[7] J. Herranz and G. Saez, "Revisiting fully distributed proxy signature schemes," in *Proceedings of Indocrypt'04*, LNCS 3348, pp. 356-370, Springer-Verlag, 2004.

[8] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Proceedings of ICICS'97*, LNCS 1334, Springer-Verlag, pp. 223-232, 1997.

[9] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages," *IEICE Trans. on Fundamentals*, vol. E79-A, no. 9, pp. 1338-1353, 1996.

[10] T. Okamoto, M. Tada, and E. Okamoto, "Extended proxy signaures for smart card," in *Proceedings of Information Security Workshop'99*, LNCS 1729, pp. 247-258, Springer-Verlag, 1999.

[11] H. M. Sun, "On proxy multi-signature scheme," in *Proceedings of International Computer Symposium*, pp. 65-72, 2000.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Crypto'84*, LNCS 196, pp. 47-53, Springer-Verlag, 1984.

[13] J. Xu, Z. Zhang, and D. Feng, "ID-Based proxy signature using bilinear pairings," Available at http://eprint.iacr.org/2004/206/

[14] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," in *Proceedings of Australasian Conference on Information Security and Privacy*, LNCS 2727, pp. 312-323, Springer-Verlag, 2003.

[15] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proceedings of Public Key Cryptography*, LNCS 2947, pp. 277-290, Springer-Verlag, 2004.

Manik Lal Das received his M. Tech. Degree in 1998. He is working in Institute for Development and Research in Banking Technology, Hyderabad as Research Officer and pursuing his Ph.D. degree in K. R. School of Information Technology, Indian Institute of Technology-Bombay, India. He has published over 15 research articles in refereed Journals/Conferences. He is a member of Cryptology Research Society of India and Indian Society for Technical Education. His research interests include Cryptography and Information Security.

Ashutosh Saxena received his M.Sc. (1990), M. Tech. (1992) and Ph.D. in Computer Science (1999). Presently, he is working as Associate Professor in Institute for Development and Research in Banking Technology, Hyderabad. He is on the Editorial Committees of various International Journals and Conferences, and is a Life Member of Cryptology Research Society of India and Computer Society of India and Member of IEEE Computer Society. He has to his credit more than 50 research papers published in National/ International Journals and Conferences. His main research interest is in the areas of Authentication Technologies, Smart Card, Key Management and Security Issues and Payment Systems in Banking.

Deepak B. Phatak received his Ph.D. degree from Indian Institute of Technology-Bombay, India. He is Subrao M. Nilekani Chair Professor with K. R. School of Information Technology, Indian Institute of Technology-Bombay, India. His research interests include Data Bases, System performance evaluation, Smart Cards and Information Systems.