CHAPTER 3

IDENTITY BASED - PROXY BLIND DISTRIBUTED DIGITAL SIGNATURE SCHEME

3.1 INTRODUCTION

This chapter addresses a new type of digital signature scheme named as Identity Based Proxy Blind Distributed Signature Scheme. The identity of a person plays a vital role in the verification process of a signed electronic document or message. In many real time applications like Automated Health Insurance Policies, the claims for the insurance holder need the identity of the person when the digitally signed bill was verified. Secondly by distributing the signing power to the group of servers, the signing process is made more trusted while compared to delegation of signing power to a single person. For example a dishonest hospital may issue a forged claim bill to a client who has not undergone any treatment. Fangguo Zhang and Kwangjo Kim (2002) presented ID-Based Blind Signature and Ring Signature from Pairings and Fangguo Zhang and Kwangjo Kim (2003) presented the Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairings. In this chapter Identity Based Proxy Blind Distributed Signature Scheme is discussed.

In a certificates-based public key system, before using the public key of a user, the participants must verify the certificates of the user at first. As a consequence, this system requires a large storage and computing time to store and verify each user's public key and the corresponding certificate.

Shamir (1984) proposed ID-based encryption and signature scheme to simplify key management procedures in certificate-based public key setting. Since then many ID-based encryption and signature schemes have been proposed.

The main idea of ID-based cryptosystems is that the identity of each user works as his/her public key, in other words, the user's public key can be calculated directly from his/her identity rather than being extracted from a certificate issued by a certification authority (CA). ID-based public key setting can be a good alternative for certificate-based public key setting, especially when efficient key management and moderate security are required.

Many ID based cryptographic schemes were proposed based on bilinear pairings, namely the Weil pairing and the Tate pairing of algebraic curves. Usage of bilinear pairings in cryptography is due to Victor Miller's (1986) unpublished paper and in particular the results of Menezes-Okamoto-Vanstone (1993) and Frey-Ruck (1994). However, most of the initial applications were to attack elliptic curve cryptosystems by transforming the ECDLP to DLP using bilinear pairings in the multiplicative group of a finite field. In the recent years, the bilinear pairings have been found various applications in cryptography. They are the basic tools for construction of ID-based cryptographic schemes. Boneh and Franklin's (2001) ID-based encryption scheme, Smart's (2002) ID-based authentication key agreement protocol and several ID-based signature schemes like Cha and Cheon (2003), Hess (2002), Paterson (2002), Sakai et al (2000), Zhang and Kim (2002) are some such schemes. An identity based signature scheme is specified by the following four steps:

1. *Setup:* An algorithm, executed by the trusted authority, which takes a random parameter l as input and generates system

parameters and master key. System parameters are publicly known, while master key is only known to the trusted authority.

2. *Extract:* An algorithm, executed by the trusted authority that takes as input the system parameters, master key and an arbitrary $ID_i \in \{0,1\}^*$, provided by a user U_i, and returns a private key x_i where $i = 1,2,..n$, n is the maximum number of users. ID_i is an arbitrary string that is used as a public key and x_i is the corresponding private key.

3. *Sign:* An algorithm that takes as input system parameters, x_i and message $m \in \{0,1\}^*$ and returns a signature σ.

4. *Verify:* An algorithm that takes as input a message $m \in \{0,1\}^*$ and its signature σ, the system parameters and a public key ID_i and outputs 1 if the signature is valid and 0 if the signature is invalid.

3.2 BILINEAR PAIRINGS

Let $(E,+)$ and $(V,+)$ denote cyclic groups of prime order q over an elliptic curve. Let P be a generator of E and let $e: E \times E \to V$ be a bilinear pairing satisfying the following conditions

i) For all points $P, Q \in E$ and for all $a, b \in Z$, we have $e(aP, bQ) = e(P, Q)^{ab}$

ii) There exist $P_1, P_2 \in E$ such that $e(P_1, P_2) \neq 1$. That is if P is a generator of E then $e(P, P)$ is the generator of V. Such groups

may be realized using super singular elliptic curves and the Weil pairing.

iii) Computing $e(P,Q)$ for all $P,Q \in E$ should be easy.

3.3 IDENTITY BASED SIGNATURE SCHEME

In this system each user has a public key based on her or his identity, such as an email address. A central trusted authority assigns a corresponding private key to each user. In most public key systems, when Alice wants to send a message to Bob, she looks up Bob's public key. However, she needs some way to being sure that this key actually belongs to Bob, rather than someone such as Eve who is masquerading as Bob. In the present system, the authentication happens in the initial communication between Bob and the trusted authority. After that, Bob is the only one who has the information necessary to decrypt message that are encrypted using his public identity.

The following gives the basic idea of the Identity Based cryptosystem. Boneh and Franklin (2001) strengthen this cryptosystem. To set up the system, the trusted authority does the following. Choose a large prime $p = 6l - 1$ for some prime l and let $q = p^2$. Choose a point P of order l in E. Choose hash functions H_1 and H. The function H_1 takes a string of bits of arbitrary length and outputs a point of order l on E. The function H_2 inputs an element of order l in Z_q^* and outputs a binary string of length n, where n is the length of the message that will be sent. Choose a secret random $s \in Z_q^*$ and compute

$$P_{pub} = sP \tag{3.1}$$

and make $p, H_1, H_2, n, P, P_{pub}$ public, while keeping s secret. If a user with the identity ID wants a private key, the trusted authority does the following. The authority computes

$$Q_{ID} = H(ID) \tag{3.2}$$

This is a point in E (Public Key). Let

$$D_{ID} = sQ_{ID} \tag{3.3}$$

After verifying that ID is the identification for the user with whom he is communicating, send D_{ID} to the user (Secret key).If Alice wants to sign a message $m \in \{0,1\}^*$, she chooses a random k and computes $(R,S) \in E \times E$ where

$$R = kP \tag{3.4}$$

$$S = k^{-1}(H_2(m).P + H_1(R).D_{ID}) \tag{3.5}$$

and outputs the signature (R,S). Checking whether a pair (R,S) is a valid signature on a message $m \in \{0,1\}^*$ with respect to the public key Q_{ID}, the verification can be done as follows. Computes $e(R,S)$ where $e(.)$ is the bilinear function. Check whether

$$e(R,S) = e(P,P)^{H_2(m)}.e(P_{pub}, Q_{ID})^{H_1(R)} \tag{3.6}$$

The signature is accepted if these values match and rejected otherwise.

3.4 MATHEMATICAL MODEL

The participating entities in the digital signature scheme are the original signer whose identity is ID_{ori}, a trusted dealer whose identity is ID_{dealer}, a group of proxy signer (servers) whose identities are $ID_{pro1}, ID_{pro2}, ..., ID_{pron}$ and the user whose identity is ID_{client}. To construct an Identity Based proxy blind distributed signature scheme the following process is required:

1. Distribution and reconstruction of a secrets using Elliptic Curves
2. Private key share distribution and verification for the valid shares
3. Joint generation of keys by the group of servers
4. Common parameter generation algorithm cpga ( )
5. Private key extraction algorithm pkeg ( )
6. Distributed signature generation algorithm dsga ( )
7. Verification algorithm va ( )

Among the above seven process, in process 1, for the distribution of secret shares and reconstruction of secret shares, Shamir's scheme is slightly modified to form a polynomial like function using the elliptic curve points. Using this function the secret share which is a point on the pre defined elliptic curve will be calculated. In the process 2, the distribution and verification for the valid shares is done using Bilinear Pairing. Process 3, 4, 5, 6, and 7 are the major contribution of this chapter.

3.5 PROCESS 1: DISTRIBUTION AND RECONSTRUCTION OF SECRETS USING ELLIPTIC CURVES

Secret sharing forms the basis of threshold cryptography. A secret is shared among n parties such that the cooperation of at least t parties is needed to recover the secret. Shamir (1979) constructed a very efficient such scheme for any n and t.

In order to share a private key which is associated with an identity, the following technique is used. There are two phases involved in it, one is distribution phase and the second is the reconstruction phase. Let us assume that q be a prime order of a group E of point on some elliptic curve.

Let $R \in E$ be a secret point to share. Suppose that t and n are such that $1 \le t \le n \le q$. Choose points $P_1, P_2, ...P_{t-1} \in E_q^*$ and define $F : N \cup \{0\} \to E$ such that

$$F(x) = R + \sum_{i=1}^{t-1} x^i P_i \tag{3.7}$$

and $x \in N$. Now $F(0) = R$ and we compute $F(i) = S_i$ for $i = 1,2,...n$ where S_i is a point on the elliptic curve group and send (i, S_i) to the i^{th} member of the group of cardinality n. In the reconstruction phase let $A \subseteq \{1,2,...n\}$ be a set such that $|A| \ge t$. The function $F(x)$ can be reconstructed by computing

$$F(x) = \sum_{j \in A} \pi_{xj}^A S_j \quad \text{where } \pi_{xj}^A = \prod_{l \in A, l \ne j} \frac{x - l}{j - l} \tag{3.8}$$

This scheme has perfect security, i.e., the shares held by every group of t or fewer servers are statistically independent of $R \in E$.

3.6 PROCESS 2: PRIVATE KEY SHARE DISTRIBUTIONS AND VERIFICATION FOR THE VALID SHARES BY THE INDIVIDUAL PROXY SERVERS

In cryptography, if auxiliary information is included in the process then it allows the users to verify their shares as consistent. Verifiable secret sharing ensures that even if the dealer is malicious there is a well-defined secret that the users can later reconstruct. (In standard secret sharing, the dealer is assumed to be honest.) The concept of verifiable secret sharing (VSS) was first introduced by Chor et al (1985).

In this chapter a Secure Secret Sharing Scheme based on the Bilinear Pairing is presented. This scheme will be used to distribute a private key associated with an identity into a group of signature generation servers. The description of this scheme is as follows. Let (E,q,P,e) be the set of parameters and t and n such that $1 \le t \le n \le q$. To share a secret $R \in E$ among n parties, a dealer performs the following steps. Choose points $P_1, P_2, ...P_{t-1} \in E_q^*$ and define $F : N \cup \{0\} \to E$ such that $F(x) = R + \sum_{i=1}^{t-1} x^i P_i$ and $x \in N$. Now $F(0) = R$ and we compute $F(i) = S_i$ for $i = 1,2,...n$. Send $(S_i, \alpha_o, \alpha_1, ...\alpha_{t-1})$ to the i^{th} member of the group of cardinality n where

$$\alpha_0 = e(R,P) \tag{3.9}$$

$$\alpha_j = e(P_j,P) \text{ for } j = 1,2,...t-1 \tag{3.10}$$

Each party then checks whether its share S_i is valid by computing

$$e(S_i,P) = \prod_{j=0}^{t-1} \alpha_j^{i^j} \tag{3.11}$$

3.7 PROCESS 3: JOINT GENERATION OF KEYS BY THE GROUP OF SERVERS

The joint generation of secret key by the group of servers is also known as distributed key-generation protocol (DKG). This protocol is used for generating a public key and a sharing of the corresponding secret key. This protocol ensures that the corrupted parties learn no information about the secret key. These protocols have been implemented for the common public key types, discrete logarithm and RSA. As discussed in chapter 2, section 2.5, the same method is used in this process. After execution of this protocol, the public key $y = g^x$ where $x = \sum_{i \in F_0} x_i \in Z_q$ is the corresponding secret key, and $x_j = \sum_{i \in F_0} s_{ij}$ is the share of server j corresponding to the secret x is generated.

3.8 PROCESS 4: COMMON PARAMETER GENERATION ALGORITHM cpga ( )

The PKG (Public Key Generator) performs the following to generate common parameters using the following steps:

1. Consider the group E of prime order q and its generator P
2. Specify the bilinear pairing $e : E \times E \to V$
3. Pick a master key $s \in Z_q^*$ uniformly at random and compute $P_{pub} = sP$
4. Choose two hash function $H_1 : \{0,1\}^* \to E_q^*$ and $H : \{0,1\}^* \to Z_q^*$
5. Keep the master key s as secret and return the common parameters $(E, q, P, P_{pub}, H, H_1)$

3.9 PROCESS 5: PRIVATE KEY EXTRACTION ALGORITHM pkeg ( )

On receiving a private key extraction query ID from any user, the PKG (Public Key Generator) performs the following:

1. Compute $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$
2. Return S_{ID}

3.10 PROCESS 6: DISTRIBUTED SIGNATURE GENERATION ALGORITHM dsga ( )

The original signer makes a warrant w, which states application dependent delegation information explicitly. Then randomly chooses $r \in Z_q^*$, $r \neq 1$ and computes

$$U_{ori} = rP \tag{3.12}$$

$$v_{ori} = H(w \,||\, e(P_{pub}, U_{ori})) \tag{3.13}$$

and $$\overline{S_{ori}} = v_{ori}.S_{ID_{ori}} + rP_{pub} \tag{3.14}$$

The original signer distributes $\overline{S_{ori}}$ to the group of proxy singers. Any subset of the proxy servers with the threshold value can reconstruct $\overline{S_{ori}}$. The original signer publishes (w, U_{ori}) as a public information. After reconstruction of the secret key by any sub set of servers (with threshold value t), the common proxy server checks the validity of the secret key by computing

$$v_{ori} = H(w \| e(P_{pub}, U_{ori})) \tag{3.15}$$

and accept if

$$e(U_{ori}, P_{pub}) = e(\overline{S_{ori}}, P) e(Q_{ID_{ori}}, P_{pub})^{-v_{ori}} \tag{3.16}$$

otherwise reject it. Then the common proxy server computes the proxy key pairs

$$S_{prox} = \overline{S}_{ori} + v_{ori} S_{ID_{prox}} \tag{3.17}$$

where

$$S_{ID_{pro}} = s.Q_{ID_{pro}} = s.H_1(ID_{pro1} + ID_{pro2} + ... + ID_{pron}) \tag{3.18}$$

generated by the main server (TA) and

$$Q_{prox} = v_{ori}.(Q_{ID_{ori}} + Q_{ID_{pro}}) + U_{ori} \tag{3.19}$$

When the user wants the group of proxies to blindly sign a message (here receipt of the claim bill)

$$m = H(claim \| ID_{client}) \tag{3.20}$$

they execute the ordinary signing operation using S_{prox} as a secret key as follows. The common proxy server computes $K = kP$ where k joint generation of shared secret by the group of proxies, and sends K to the user. The user chooses randomly $a, b \in Z_q^*$ and computes

$$t = e(bQ_{prox} + K + aP, P_{pub}) \tag{3.21}$$

$$c = H(m \| t) + b \bmod q \tag{3.22}$$

and send c to the common proxy server. The common proxy server computes

$$S = cS_{prox} + kP_{pub} \tag{3.23}$$

and sends it to the user. The user computes

$$S^1 = S + aP_{pub} \tag{3.24}$$

and $$c^1 = c - b \tag{3.25}$$

The Identity based proxy blind distributed signature of m = (S^1, c^1).

3.11 PROCESS 7: VERIFICATION ALGORITHM va ( )

The verifier or the receiver accept the signature if and only if

$$c^1 = H(m \| e(S^1, P)e(Q_{prox}, P_{pub})^{-c^1}) \tag{3.26}$$

where $m = H(claim \| ID_{client})$

Now, based on the above mathematical equations the following claim can be made. The validity of the signature scheme is true if and only if $c^1 = H(m \| e(S^1, P)e(Q_{prox}, P_{pub})^{-c^1})$

Proof

We have $H : \{0,1\}^* \to Z_q^*$, $H_1 : \{0,1\}^* \to E^*$, therefore

$H(m \| e(S^1, P)e(Q_{prox}, P_{pub})^{-c^1}$

$= H[m \| e(S + aP_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$

$$= H[m \,||\, e(cS_{prox} + kP_{pub} + aP_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(S_{prox}, P)^c e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(\overline{S}_{ori} + v_{ori}S_{IDprox}, P)^c e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(v_{ori}S_{IDori} + rP_{pub} + v_{ori}S_{IDprox}, P)^c .\, e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(v_{ori}.s.Q_{IDori} + r.sP + v_{ori}.sQ_{IDprox}, P)^c .\, e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(v_{ori}.Q_{IDori} + rP + v_{ori}Q_{IDprox}, P_{pub})^c .\, e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(v_{ori}\left(Q_{IDori} + Q_{IDprox}\right) + U_{ori}, P_{pub})^c .\, e(k+a)P_{pub}, P)e(Q_{prox}, P_{pub})^{-c^1}]$$

$$= H[m \,||\, e(Q_p, P_{pub})^c e(Q_{prox}, P_{pub})^{-c^1} e(k+a)P_{pub}, P)]$$

$$= H[m \,||\, e(Q_{prox}, P_{pub})^{c-c'} e(k+a)P, P_{pub})]$$

$$= H[m \,||\, e(\left(c-c'\right)Q_{prox}, P_{pub})e(k+a)P, P_{pub})]$$

$$= H[m \,||\, e(bQ_{prox}, P_{pub})e(k+a)P, P_{pub})]$$

$$= H[m \,||\, e(bQ_{prox} + (k+a)P, P_{pub})]$$

$$= H[m \,||\, e(bQ_{prox} + kP + aP, P_{pub})]$$

$$= H[m \,||\, e(bQ_{prox} + K + aP, P_{pub})]$$

$$= H[m \,||\, t]$$

$$= c - b = c'$$

2.12 SECURITY ANALYSIS

Now, based on the above mathematical equations the following security analysis is carried out. 'Provable security '- approach is used for evaluating the security of the digital signature scheme. The security of this digital signature scheme is based on the intractability of the Computational Diffie-Hellman Problem.

Theorem 3.1

Any one except the proxy signer's cannot generate a valid proxy private key. That is an attacker cannot construct a valid proxy private key S_{pro}.

Proof

$$S_{pro} = \overline{S}_{ori} + v_{ori} S_{ID_{prox}}$$

$$= \overline{S}_{ori} + v_{ori}[s.Q_{ID_{pro}}]$$

$$= \overline{S}_{ori} + v_{ori}[s.H_1(ID_{pro1} + ID_{pro2} + ... + ID_{pron})]$$

since $H_1 : \{0,1\}^* \rightarrow E_q^*$, let $H_1(ID_{pro1} + ID_{pro2} + ... + ID_{pron}) = T \in E_q^*$ hence

$$S_{pro} = \overline{S}_{ori} + v_{ori}.s.T$$

by Equation (3.14) $\overline{S_{ori}} = v_{ori}.S_{ID_{ori}} + rP_{pub}$, therefore

$$S_{pro} = v_{ori}.S_{ID_{ori}} + rP_{pub} + v_{ori}.s.T$$

$$= v_{ori}.s.Q_{ID_{ori}} + rP_{pub} + v_{ori}.s.T \text{ , since } P_{pub} = sP$$

$$= v_{ori}.s.Q_{ID_{ori}} + r.s.P + v_{ori}.s.T$$

$$= v_{ori}.s.H_1(ID_{ori}) + r.s.P + v_{ori}.s.T \text{ , let } H_1(ID_{ori}) = T_1 \in E_q^*$$

$= v_{ori}.s.T_1 + r.s.P + v_{ori}.s.T$

$= v_{ori}.s.(T + T_1) + r.s.P$, let $T + T_1 = T_2 \in E_q^*$

$= v_{ori}.s.(T_2) + r.s.P$

Since $v_{ori} = H(w \| e(P_{pub}, U_{ori}))$ and $H : \{0,1\}^* \to Z_q^*, v_{ori} \in Z_q^*$, moreover $r, s \in Z_q^*$. We know that for $a, b \in Z_q^*$, given P, aP, bP, computing abP is mathematically hard by CDHP discussed in section 1.9.4.

Therefore an attacker cannot construct $r.s.P$ even if P, rP, sP is known. Hence an attacker cannot construct a valid proxy private key S_{pro}.

Theorem 3.2

The attacker cannot construct a valid Identity based proxy blind distributed signature for a selected message m^1 without the knowledge of c and k.

Proof

We know that the Identity based proxy blind distributed signature for the message m is $(S^1, c^1) = (S + aP_{pub}, c - b)$

Now $S = cS_{prox} + kP_{pub}$ and $S_{prox} = \overline{S}_{ori} + v_{ori} S_{ID_{prox}}$

Therefore $S = c[\overline{S}_{ori} + v_{ori} S_{ID_{prox}}] + kP_{pub}$ since $\overline{S_{ori}} = v_{ori}.S_{ID_{ori}} + rP_{pub}$

$S = c[(v_{ori}.S_{ID_{ori}} + rP_{pub}) + v_{ori} S_{ID_{prox}}] + kP_{pub}$

$= c[(v_{ori}.s.Q_{ID_{ori}} + r.s.P) + v_{ori}(s.Q_{ID_{prox}})] + k.sP$

$= c.v_{ori}.s.Q_{ID_{ori}} + c.r.s.P + c.v_{ori}.s.Q_{ID_{prox}} k.sP$

Moreover $c = H(m \| t) + b \bmod q$ therefore

$c - b = H(m \| t)$

Therefore $(S^1, c^1) = (c.v_{ori}.s.Q_{ID_{ori}} + c.r.s.P + c.v_{ori}.s.Q_{ID_{prox}} k.sP, H(m \| t))$

Using the assumption that CDHP is very hard to solve, an attacker cannot construct a valid S^1 without the knowledge of c.

If the attacker obtain $c = b + H(m^1 \| t)$, since $t = e(bQ_{prox} + K + aP, P_{pub})$ and $K = kP$ where k joint generation of shared secret by the group of proxies, which was send to the user, he cannot find a corresponding k by checking $K = kP$. Therefore an attacker cannot construct a valid Identity based proxy blind distributed signature for a selected message m^1.

Theorem 3.3

Each parties share S_i is valid if and only if $e(S_i, P) = \prod_{j=0}^{t-1} \alpha_j^{i^j}$

Proof

We know that $P_1, P_2, ...P_{t-1} \in E_q^*$, $\alpha_0 = e(R, P)$ and $\alpha_j = e(P_j, P)$ for $j = 1,2,...t-1$.

Now $\prod_{j=0}^{t-1} \alpha_j^{i^j} = \alpha_0.\alpha_1^{i}.\alpha_2^{i^2}.\alpha_3^{i^3}...\alpha_{t-1}^{i^{t-1}} = e(R, P).e(P_1, P)^{i}.e(P_2, P)^{i^2}...e(P_{t-1}, P)^{i^{t-1}}$

Also $S_i = F(i)$ for $i = 1,2,...n$ and $F(0) = R$

But $F(x) = R + \sum_{i=1}^{t-1} x^i P_i$ therefore

$S_i = F(i) = R + \sum_{i=1}^{t-1} i^i P_i = R + iP_1 + i^2 P_2 + ... + i^{t-1} P_{t-1}$

$$e(S_i, P) = e(R + iP_1 + i^2P_2 + ... + i^{t-1}P_{t-1}, P)$$

$$= e(R, P).e(iP_1, P).e(i^2P_2, P)...e(i^{t-1}P_{t-1}, P)$$

$$= e(R, P).e(P_1, P)^i.e(P_2, P)^{i^2}...e(P_{t-1}, P)^{i^{t-1}}$$

$$= \prod_{j=0}^{t-1} \alpha_j^{i^j}$$

Hence each parties share S_i is valid if and only if $e(S_i, P) = \prod_{j=0}^{t-1} \alpha_j^{i^j}$.

3.13 NUMERICAL RESULTS

Let E be the elliptic curve $y^2 = x^3 + x$ with $p = 7$ and let $E = \{(0,0),(1,3),(1,4),(3,3),(3,4),(5,2),(5,5),O\}$. To construct a bilinear mapping, consider the torsion points $(0,0)$ and $(i,0)$ which are linearly independent and lies on the curve E. Here $i^2 = -1$ and the point $(i,0) \in E$ since $(i,0)$, satisfies the elliptic curve equation E. Therefore $f_{(0,0)} = \frac{x(x-3)^2}{(y+x)^2}$ and $f_{(i,0)} = x - i$. Hence the Weil Pairing evaluation output for $(0,0)$ and $(i,0)$ is $e((0,0),(i,0)) = 6$.

Let $s = (5,5)$ be the secret point to be shared among the group of servers. Let P= {1, 2, 3, 4, 5} be the set of servers with n = 5. To construct the polynomial function, consider the points $(1,3),(3,3)\ \&\ (3,4)$. Then the secret shares for the individual servers are $s_1 = (1,4)$, $s_2 = (3,4)$, $s_3 = (0,0)$, $s_4 = (3,4)$, $s_5 = (1,3)$. Any set of servers not less than the threshold value can construct the secret. Let A= {1, 3, 4, 5} be the authorized subset of P with threshold value t=4. They now combine to get the secret value.

Let $P = (5,2)$ and the master secret of the trusted authority $s = 3$ therefore $P_{pub} = 3(5,2) = (3,4)$. Let $Q_{IDori} = (1,3)$ therefore $S_{IDori} = 3(1,3) = (1,4)$. Let $Q_{IDprox} = (3,3)$ therefore $S_{IDprox} = 3(3,3) = (5,5)$ with $r = 4$, $U_{ori} = 4(5,2)$ and $e((3,4),(1,3)) = 1$. Let $H(m \| t) = 5$. Then the Identity Based Proxy Blind distributed signature for the message m is $\text{Sign(m)} = (S^1, c^1) = ((5,2),5)$.

For the verification of the signature LHS = $c^1 = 5$ and the RHS = $c^1 = H(m \| e(S^1, P)e(Q_P, P_{pub})^{-c^1}) =$ $c^1 = H(m \| t) = 5$. Hence the signature scheme is verified.

3.14 PERFORMANCE EVALUATION

The performance of the scheme in terms of number of keys, computational complexity has been analyzed in this section. The following notations were used to analyze the performance of the scheme

- **SK** and **PK** are the number of secret and public keys respectively
- **T** Trusted third Party
- **M** is the time for modular multiplication
- **I** is the time for a modular inverse Computation
- **H** is the time for performing a one-way Hash
- **n** is total number of persons involved.
- l is Order of the Elliptic Curve

Here the time for performing modular addition/subtraction computation is ignored. The performance evaluation of the proposed scheme in Table 3.1 and the comparison of time needed is given in Table 3.2.

Table 3.1 Performance evaluation of the proposed scheme

		Our Signature Scheme
The number of Keys	SK(T)	1
	PK(T)	1
Computational Complexity	Key Generation	$(3nl+18l+6)M+(nl+3l+4)I$
	Signing	$(24l+16)M+(5l+4)I+H$
	Verify	$(18l+32)M+8I+H$

Table 3.2 Comparison of time needed

Description	**Key Size**	**Our scheme time (nano seconds)**	**Zhang's scheme time (nano seconds)**
Key Generation	4 bit	22246156	18238567
	8 bit	189711165	130375927
	16 bit	5047744870	4292749573
	32 bit	89676655756	73243659672
Signature Generation	4 bit	12058664	13926483
	8 bit	16307550	18173645
	16 bit	39337452	47296745
	32 bit	273655755	398361859
Signature Verification	4 bit	14345652	19386710
	8 bit	19025475	23937562
	16 bit	44287026	58275968
	32 bit	309776766	483978254
Overall process	4 bit	48650472	51551760
	8 bit	225044190	172487134
	16 bit	5131369348	4398322286
	32 bit	90260088277	74125999785

3.15 INFERENCE AND CONCLUSION

Based on the numerical data, the time needed for the overall processes in the proposed digital signature scheme had been plotted in Figure 3.1. The ID based scheme is compared with the Zhang's scheme.

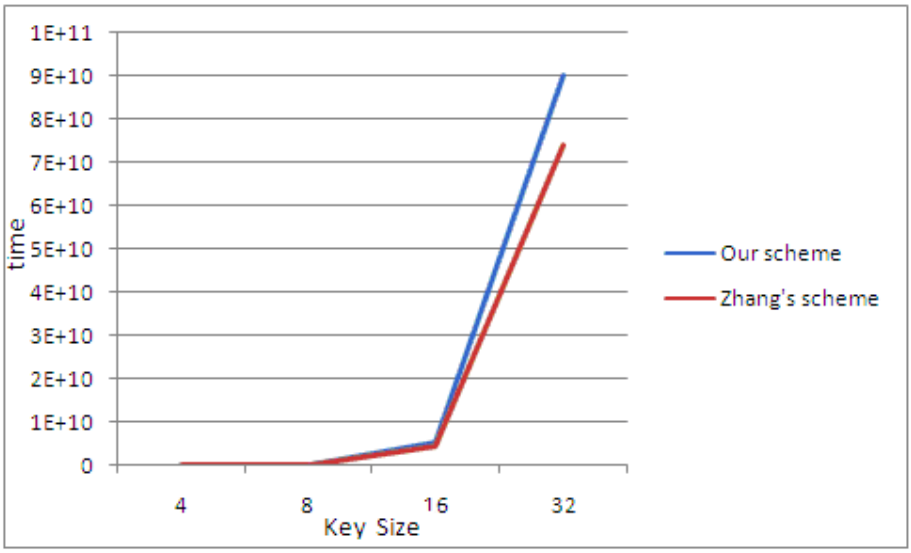

Figure 3.1 Comparison of the overall process with Zhang's scheme

Based on the discussions made, the following conclusions can be drawn.

- Though the system is distributed, the computation time for implementing the identity based proxy blind distributed digital signature scheme is very similar to Zhang's signature scheme; therefore no additional time is required for processing the proposed digital signature scheme.

- This scheme prevents the original signer's forgery by Theorem 3.1.
- It prevents the recipient's universal forgery by Theorem 3.2
- As this scheme delegates the signing authority to a group of systems with a threshold version. The system will output the correct result even if one or few system is busy or corrupted and hence the scheme is robust.