

Group-oriented (k,n) signature schemes on exponentiation of primitive root and Elliptic curve

C. Porkodi¹ and R. Arumuganathan²

Department of Mathematics and Computer Applications
PSG College of Technology, Coimbatore – 641 004. Tamilnadu, India
Email:¹porkodi_c2003@yahoo.co.in, ²ran_psgtech@yahoo.co.in

Abstract: In this paper we construct two group oriented (k,n) signature schemes, one based on exponentiation of primitive root and the other on elliptic curve. The security of schemes depends on difficulty of solving the discrete logarithm problem and integer factorization problem. The message is signed with the assistance of a trusted arbitrator. The trusted arbitrator is the deciding authority for the construction of the group secret key and the individual secret keys of the group members. We discussed how the proposed algorithms withstand various attacks and thus the schemes are well protected in security aspect.

Keywords: Field, primitive root, Euler totient function, elliptic curve, discrete logarithm problem, integer factorization problem, symmetric functions.

1. Introduction:

Confidentiality, Data integrity, authentication and non repudiation are the important goals of many Cryptographic applications. A traditional approach to achieve these requirements is to “encrypt and sign” the messages. Digital signatures provide authentication service. The conventional digital signatures involve a single signer. The security of the conventional digital signature RSA [17] relies on the NP-hard integer factorization problem. The security of Elgamal [4] is based on the discrete logarithm problem. But in today’s business scenario messages are frequently addressed to a group of people. For example, for security purpose a company may have the policy that the cheques must be signed by a group of managing directors instead of a single person. In a bank a safe may be designed in such a way that only a group of senior tellers gain access to the safe by pooling their individual secret keys and no individual teller has access to the safe. Conventional and public key systems are not adopted when messages are intended to a group instead of an individual. In such communication the concept of threshold signatures and group signature schemes are introduced.

A group oriented (k,n) signature has the following properties

- (i) Only k members of the group can sign the messages.
- (ii) The receiver of the signature can verify that it is a valid signature from the group.
- (iii) The receiver of the signature cannot determine the identities of the signers

(iv) in the case of dispute, the signature can be opened to reveal the identity of the signer

Group signatures can be done in two categories

- (i) with the assistance of a trusted arbitrator
- (ii) without the assistance of a trusted arbitrator.

In the first case the trusted arbitrator generates a large pile of public key and private key pairs and gives every member of the group a different list of unique private keys such that no keys on any list are identical. Trusted arbitrator publishes the list of all public keys for the group and keeps a secret record of which keys belong to whom. When the group members wish to sign a document, he chooses a key at random from his personal list. When someone wants to verify that a signature belongs to the group, he looks on the master list for the corresponding public key and verifies the signature. In the event of dispute trusted arbitrator knows which public key corresponds to which group member. At the same time, no one (including the trusted arbitrator) will be able to falsely accuse any other member of the group.

Group signatures allow group members to anonymously sign arbitrary message on behalf of the group. For example, a group signature scheme could be used by an employee of a large company where it is sufficient for a verifier to know a message was signed by an employee, but not the particular employee who signed it.

The basic requirements of a group signature are

Soundness and Completeness: Valid signatures by group members always verified correctly and invalid signatures always fail verification.

Unforgeable: Only group members can create valid group signatures.

Signer ambiguous: Given a message and its signature, the identity of individual signer cannot be determined without revocation.

Unlinkability: Given two messages and their signatures it cannot be identified if the signatures were from the same signer or not.

No framing: Even if all other group members combine together, they cannot forge a signature for a non-participating group member.

The group signature scheme was first introduced by Chaum and Van Heyst [2]. Lee and Chang [11] proposed a linkable signature scheme based on discrete logarithms in which two same pieces of information are included in all group signatures generated by the same group member. Tseng and

Jan proposed an improved group signature scheme [21] which provides the unlinkability and an ID based group signature scheme [22]. He Ge [8] presented an effective method to integrate the revocation mechanism into some group signature schemes based on RSA assumption. Ratna Dutta and Rana Barua [16] proposed a password based authenticated encrypted group agreement protocol immune to dictionary attack under the computation of Diffie-Hellman assumption.

A (k, n) threshold scheme ($k \leq n$) is a method by which a trusted party computes secret shares S_i , $1 \leq i \leq n$ from an initial secret S and securely distributes S_i to user P_i such that any k or more users can pool their shares and recover S but any group knowing only $k - 1$ or fewer shares may not. Desmedt and Frankel [3] applied a trusted key authentication center to determine the group's secret key and individual group member's secret keys. Li et al [12] showed that Desmedt and Frankel's scheme is not resistible against conspiracy attack, the group key can be recovered by t or more participants by pooling their secret keys. To avoid the conspiracy attack Li et al [13] attached a random number to the sub keys of all participants. Jan et al [10] proposed a threshold signature scheme based on polynomial reconstruction problem using Lagrange interpolation formula. Yu Fang Chung et al [20] proved that the threshold signature scheme proposed by Jan et al fails to resist conspiracy attack. L.Harn et al [6] proposed a threshold cryptosystem with multiple secret policies which depends on a trusted key centre for providing secret keys to the group members and publishing the corresponding public keys. L.Harn presented [7] a group oriented threshold signature schemes based on difficulty of solving discrete logarithm problem. In the group oriented based deniable authentication protocol from the bilinear pairings proposed by Rongxing Lu and Zhenfu Cao [18] the sender is not a single person but the signature is generated by a group of members. Yu Long and Ke-Fei Chen [23] proposed a dynamic (k, n) threshold decryption scheme from pairing, which allows the master key and n -decryption servers secret keys to be renewed or to add/remove a decryption server without secure channels between a trusted private key generator and decryption servers.

Elliptic curve cryptography described in 1985 is a public key algorithm with shorter key lengths that provides improved performance over systems based on integer factorization and discrete logarithms depending upon the operating platform and applications for which they are used.

In the threshold signcryption scheme based on elliptic curves and verifiable secret sharing by Peng Changgen et al [15], the group members are in need of a trusted center for generating parameters. Here a subgroup of t - participants randomly select a person from the group as a designated clerk, who is responsible for collecting and verifying the partial signcryption and group signcryption. Maged H.Ibrahim [14] proposed a robust threshold digital signature based on elliptic curves.

In this paper we propose k participant group signature schemes, one based on exponentiation of primitive root and other on elliptic curve. The group signature is done with the assistance of a trusted arbitrator and each participant is given

a single private and public key instead of a pile of public key and private key pairs. The secret keys used in this algorithm are secured as they are constructed based on the NP hard problems discrete logarithm problem and integer factorization problem.

2. Basic definitions:

2.1 Primitive root

An integer g is called the primitive root mod n if $\phi(n)$ is the smallest positive integer satisfying $g^{\phi(n)} \equiv 1 \pmod{n}$.

2.2 Integer factorization problem

Given a positive integer n , find its prime factorization $n = p_1^{e_1} \cdot p_2^{e_2} \dots p_k^{e_k}$, where p_i 's are pair wise distinct primes and $e_i \geq 1$

2.3 Discrete logarithm problem

Given a finite cyclic group G of order n , a generator α of G and an element $\beta \in G$, find the integer x , $0 \leq x \leq n-1$ such that $\alpha^x \equiv \beta \pmod{n}$.

2.4 Elliptic curve

Let K be a field (either the field $\mathcal{Q}, \mathcal{R}, \mathcal{C}$ or F_p of characteristic $\neq 2, 3$, then an elliptic curve over K is the set of points (x, y) with $x, y \in K$ that satisfy the cubic Diophantine equation $E: y^2 = x^3 + ax + b$, (where the cubic on the right-hand side has no multiple roots, ie, $4a^3 + 27b^2 \neq 0$) together with a single element O_E , called point at infinity.

2.5 Addition of points on elliptic curve

Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ be two points on elliptic curve $E: y^2 = x^3 + ax + b$, then $P_3 = (x_3, y_3) = P_1 + P_2$ on E is computed as

$$P_1 + P_2 = \begin{cases} O_E, & \text{if } x_1 = x_2 \text{ \& } y_1 = -y_2 \\ (x_3, y_3), & \text{otherwise} \end{cases}$$

Where, $(x_3, y_3) = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1)$ and

$$\lambda = \begin{cases} \frac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \\ \frac{y_2 - y_1}{x_2 - x_1} & \text{otherwise} \end{cases}$$

2.6 Elliptic curve discrete logarithm problem

Fix a prime p and an elliptic curve $E(F_p)$, $Q = xP$ represents that the point P on elliptic curve added to itself x times. Then the elliptic curve discrete logarithm problem is to determine x given P and Q . It is relatively easy to calculate Q given x and P , but it is very hard to determine x given Q and P .

(I) Group signature scheme based on exponentiation of primitive root

3. System Initialization

This method requires a trusted arbitrator (TA) in charge of processing parameter setup. Assume that there are n participants in a group represented as $A = \{A_1, A_2, \dots, A_n\}$ where the i th participant is A_i . A message is signed on behalf of the group only if k participants agree to sign the message. These k participants form a subgroup B of A . Whenever these participants wish to sign a message they place request to the trusted arbitrator to provide the keys. For the subset of B with k participants, the trusted arbitrator performs the following procedure to setup the system initialization stage.

Step 3.1: Select two secure prime numbers p and q such

that $p = (2p' + 1)$ and $q = (2q' + 1)$ where p' and q' are also large prime numbers.

Step 3.2: Compute $N = pq$ and the Euler totient

function $v = \phi(N) = 4p'q'$.

Step 3.3: Select the primitive root $g \in Z_N^*$.

Step 3.4: Select the group public key e of A such that $1 < e < v$ and $(e, v) = 1$.

Step 3.5: Select the group private key d of A such that $1 < d < v$ and $ed \equiv 1 \pmod{v}$.

Step 3.6: Select the prime numbers p_1, p_2, \dots, p_n are less than $\min(p, q)$ and compute $l = p_1 \cdot p_2 \cdot \dots \cdot p_n$

Step 3.7: Determine the sub group private key and corresponding public key

$x = (l - 1) \pmod{v}$ and $y = g^x \pmod{N}$

Step 3.8: For each participant A_i of B , mutually trusted party generates the secret key x_i and public key y_i as follows

$$x_i = \left(g^{\left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j + \frac{1}{k} \sum_{j \notin B} a_j \right) \pmod{v}} \right)^d \pmod{N}$$

$$y_i = \left(g^{\left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j + \frac{1}{k} \sum_{j \notin B} a_j \right) \pmod{v}} \right) \pmod{N}$$

Where a_i 's are elementary symmetric functions associated with each participant P_i given by

$$a_1 = \sum_{1 \leq i \leq n} (p_i - 1) \pmod{v}$$

$$a_2 = \sum_{1 \leq i < j \leq n} (p_i - 1)(p_j - 1) \pmod{v}$$

$$a_3 = \sum_{1 \leq i < j < k \leq n} (p_i - 1)(p_j - 1)(p_k - 1) \pmod{v}$$

...

$$a_n = (p_1 - 1)(p_2 - 1)(p_3 - 1) \dots (p_n - 1) \pmod{v}.$$

These a_i are kept secret by the trusted arbitrator.

Step 3.9: Send the key pair (x_i, y_i) to each individual A_i of the subgroup B for $i=1, 2, \dots, k$. Each participant conforms that the key pair is from TA by verifying the equation $y_i = x_i^e$.

Step 3.10: Publish a one way hash function $h(\cdot)$.

Step 3.11: Select a public key $1 < e_1 < \phi(N)$ such that

$$e_1 \neq e \text{ and } (e_1, \phi(N)) = 1.$$

Step 3.12: Choose a secret key $1 < d_1 < \phi(N)$ such

$$\text{that } e_1 d_1 \equiv 1 \pmod{\phi(N)}.$$

4. Signature generation

The sub group of k participants $B = \{A_1, A_2, \dots, A_k\}$ after getting the keys signs a message m on behalf of the group A . Each of the participants signs the message m as follows.

Step 4.1: Select a random integer $k_i \in Z_N^*$ and

$$\text{calculates } r_i = g^{k_i e} \pmod{N}.$$

Step 4.2: Broadcast r_i to other participants of the subgroup B .

Step 4.3: Determines the product $R = \prod_{j \in B} r_j \pmod{N}$.

Step 4.4: Calculate partial signature s_i using the individual private key x_i and the random k_i as

$$s_i = x_i^{h(m, R)} g^{k_i} \pmod{N}.$$

Step 4.5 Send the partial signature (r_i, s_i) to the trusted arbitrator.

After receiving (r_i, s_i) , the trusted arbitrator validates each of the t partial signatures by verifying

$$(s_i)^e = y_i^{h(m, R)} r_i \pmod{N}.$$

Only if the above equation is satisfied, the trusted arbitrator accepts (r_i, s_i) as a valid partial signature by A_i . Once all partial signatures of the participants are validated, the trusted arbitrator generates the group signature (R, S) for the message m on the behalf of A , as $S = \left(\prod_{i \in B} s_i \right) \pmod{N}$.

The trusted arbitrator signs this group signature using his private key d_1 as $S_1 = S^{(d_1)} \pmod{N}$. Now $(m, S(M), S_1(m))$ is taken to be the issued group signature for a message m .

5. Signature Verification:

After receiving the message m sealed with the group signature (R, S) of A , the verifier V first verifies that $S_1(m)$ is the valid signature for $S(m)$ using the public key e_1 and then $S(m)$ is a valid signature for m by verifying the equation $S^e \equiv y^{h(m,R)} R \pmod{N}$ (Theorem 3 of Appendix).

6.0 Security Analysis:

(1) Verifiable secret key distribution: In the secret key split key space each participant can check whether his secret

key is from the trusted arbitrator by verifying $y_i = x_i^e$.

Hence this group signature scheme can prevent the impersonation of some other intruder as a trusted arbitrator.

(2) Unforgeability of partial signature: In the partial signature phase it is impossible for any attacker to create partial signature for forged message because he has to

create signature s_i satisfying $(s_i)^e = y_i^{h(m,R)} r_i \pmod{N}$. For

that he must create a secret key x_i satisfying $y_i = x_i^e$ for $i = 1, 2, \dots, k$.

(3) Against conspiracy attack: As the group signature S is again signed by the trusted arbitrator with his secret key the proposed scheme prevents any set of participants, from sending the group signature for forged message without the knowledge of trusted arbitrator. The secret key of the arbitrator d_1 is constructed based on the NP hard integer factorization problem and hence it is computationally infeasible for the conspirators to get this secret key.

Even though it is possible for the conspirators to generate

the group signature $S = y^{dh(m,R)} g^k$ with $R = g^{ke}$

satisfying $S^e = y^{h(m,R)} R$ (By Theorem 2 of appendix, with the available secret keys x_i for $i = 1, 2, \dots, k$ the conspirators

can compute y^d), they cannot send the group signature to the verifier without the help of trusted arbitrator.

(4) Confidentiality of private keys: If any attacker wants to recover the secret keys x, x_i, k_i and k from the available public information y, y_i, r_i and R , he has to solve the discrete problem.

7. Numerical illustration: (3, 5) group oriented system

Step 7.1: The trusted arbitrator (TA) choose $p = 467$,

$q = 503$, computes $N = pq = 234901$ and

$v = \phi(N) = (p-1)(q-1) = 233932$

Step 7.2: The TA select the primitive root as $g = 912$.

Step 7.3: The TA chooses the public keys as $e = 1453$,

$e_1 = 281$ and private key as $d = 161, d_1 = 281$

such that $ed \equiv 1 \pmod{v}$ and $e_1 d_1 \equiv 1 \pmod{v}$.

Step 7.4: The TA chooses the prime numbers $p_1 = 47$,

$p_2 = 79, p_3 = 97, p_4 = 137, p_5 = 101$ and

computes the symmetric functions $a_1 = 456$,

$a_2 = 81012, a_3 = 208532, a_4 = 33792, a_5 = 4500$

and $l = 94361$. Group private key is $x = 94360$ and public key is $y = 224064$

Step 7.5: Suppose the participants P_2, P_4, P_5 wish to sign a

message, the TA constructs the individual secret

keys as $x_2 = 233404, x_4 = 129065, x_5 = 234357$

and public keys as $y_2 = 58035, y_4 = 153686$,

$y_5 = 187284$.

Step 7.6: These participants choose secret $k_2 = 1052$,

$k_4 = 2100, k_5 = 198$ respectively, computes

$r_i = g^{k_i e} \pmod{N}$ and broadcast among each other

$r_2 = 44418, r_4 = 87918, r_5 = 72665$

Each participant calculates $R = \prod_i r_i = 24672$ and

the hash value of the message m

Step 7.7: Suppose the hash value of the

message m is $h(m, R) = 512$, the participants

generates the signatures as

$s_2 = 199959, s_4 = 170857, s_5 = 67796$ and send

to TA.

Step 7.8: TA verifies these signatures by checking

$s_2^e = y_2^e r_2 = 158708$,

$s_4^e = y_4^e r_4 = 215942$ and

$s_5^e = y_5^e r_5 = 78238$ and calculates the group

signature as $S = \prod_i s_i = 201861$

Step 7.9: TA signs the group signature S with his secret

key d_1 as $S_1 = 128220$ and sends the triplet

(S, R, S_1) .

Step 7.10: The verifier conforms that the signature is from

TA by verifying $S_1^{e_1} = S = 201861$

(II) Group signature scheme based on Elliptic curves

8. System initialization stage

For the subset of B with k participants, the trusted arbitrator performs the following procedure to setup the system initialization stage.

Step 8.1: Select a prime field F_p and the elliptic curve $E(F_p)$.

Step 8.2: Compute the base point P and the order q of P .

Step 8.3: Select the group public key e of A such that $1 < e < q$ and $(e, q) = 1$.

Step 8.4: Select the group private key d of A such that $1 < d < q$ and $ed \equiv 1 \pmod{q}$.

Step 8.5: Select $l = p_1 \cdot p_2 \dots p_n$ where the prime numbers p_1, p_2, \dots, p_n are less than p .

Step 8.6: Determine the sub group private key and corresponding public key $x = (l-1) \pmod{q}$ and $y = xP$.

Step 8.7: For each participant A_i of B , trusted arbitrator generates the secret key pair (x_i, y_i) as follows

$$x_i = \left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j + \frac{1}{k} \sum_{j \notin B} a_j \right) dP$$

$$y_i = \left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j + \frac{1}{k} \sum_{j \notin B} a_j \right) P$$

Where a_i 's are elementary symmetric functions associated with each P_i given by

$$a_1 = \sum_{1 \leq i \leq n} (p_i - 1) \pmod{q}$$

$$a_2 = \sum_{1 \leq i < j \leq n} (p_i - 1)(p_j - 1) \pmod{q}$$

$$a_3 = \sum_{1 \leq i < j < k \leq n} (p_i - 1)(p_j - 1)(p_k - 1) \pmod{q}$$

...

$$a_n = (p_1 - 1)(p_2 - 1)(p_3 - 1) \dots (p_n - 1) \pmod{q}.$$

These a_i are kept secret by the trusted arbitrator.

Step 8.8: Send the key pair (x_i, y_i) to each individual A_i for $i=1, 2, k$, the k members of the subgroup B .

Each A_i conforms that the key pair is from TA by examining $y_i = ex_i$.

Step 8.9: Publish a one way hash function $h(\cdot)$.

Step 8.10: Select a public key $1 < e_1 < q$ such that $e_1 \neq e$ and $(e_1, q) = 1$.

Step 8.11: Choose a secret key $1 < d_1 < q$ such that

$$e_1 d_1 \equiv 1 \pmod{q}.$$

9. Signature generation

The sub group of k participants $B = \{A_1, A_2, \dots, A_k\}$ after getting the keys signs a message m on behalf of the group A . Each of the participants signs the message m as follows.

Step 9.1: Select a random integer $k_i \in F_p$ and

$$\text{calculates } r_i = k_i(eP).$$

Step 9.2: Broadcast r_i to other participants of the subgroup B .

Step 9.3: Determines the sum $R = \sum_j r_j$ and R_x be

X coordinate of R .

Step 9.4: Calculate partial signature s_i using the individual private key x_i and the random k_i as

$$s_i = h(m, R_x) x_i + k_i P$$

Step 9.5: Send the partial signature (r_i, s_i) to the trusted arbitrator.

After receiving (r_i, s_i) , the trusted arbitrator validates each of the k partial signatures as

$$es_i = h(m, R_x) y_i + r_i \text{ for } i = 1, 2, \dots, k.$$

Only if the above equation is satisfied, the trusted arbitrator accepts (r_i, s_i) as a valid partial signature by A_i . Once all partial signatures of the participants are validated, the trusted arbitrator generates the group signature (R, S) for the message m on the behalf of A as $S = \sum_i s_i$.

The trusted arbitrator signs this group signature using his private key d_1 as $S_1 = d_1 S$. Now $(m, S(M), S_1(m))$ is taken to be the issued group signature for a message m .

10. Signature Verification:

After receiving the message m sealed with the group signature (R, S) of A , the verifier V first verifies that $S_1(m)$ is the valid signature for $S(m)$ using the public key e_1 and then $S(m)$ is a valid signature for m by verifying the equation $eS = h(m, R_x)y + R$ (Theorem 5 of Appendix).

11. Security Analysis:

(1) **Verifiable secret key distribution:** In the secret key split key space each participant can check whether his secret key is from the trusted arbitrator by verifying $y_i = ex_i$.

Hence this group signature scheme can prevent the impersonation of some other intruder as a trusted arbitrator.

(2) **Unforgeability of partial signature:** In the partial signature phase, it is impossible for any attacker Eve to

create partial signature for forged message because he has to generate signature s_i satisfying $es_i = h(m, R_x)y_i + r_i$ to forge the receiver. To create such a partial signature for forged message he has to construct a secret key x_i

satisfying $y_i = ex_i$ for $i = 1, 2, \dots, k$. It is computationally infeasible.

(3) Against conspiracy attack: As the group signature S is again signed by the trusted arbitrator with his secret key the proposed scheme prevents any set of participants, from sending the group signature for forged message without the knowledge of trusted arbitrator. The secret key of the arbitrator d is constructed based on the NP hard integer

factorization problem and hence it is computationally infeasible for the conspirators to get this secret key.

Even though it is possible for the conspirators to generate the group signature $S = h(m, R_x)dy + kP$ satisfying

$$eS = h(m, R_x)y + R \text{ (By Theorem 4 of Appendix, with the}$$

available secret keys x_i for $i = 1, 2, \dots, k$ the conspirators can compute dy), they cannot send the group signature to the verifier without the help of trusted arbitrator.

(4) Confidentiality of private keys: If any attacker wants to recover the secret keys x, x_i, k_i and k from the available public information y, y_i, r_i and R , he has to solve the Elliptic curve discrete logarithm problem.

12. Numerical illustration: (3,5)group oriented signature scheme.

Step 12.1: The trusted arbitrator choose the elliptic curve $y^2 = x^3 - 4 \pmod{211}$ over the prime field F_{211} .

The base point of the elliptic curve $P = (94, 57)$ has the order $q = 241$.

Step 12.2: The arbitrator selects $e = 16, d = 226$ such

$$\text{that } ed \equiv 1 \pmod{241} \text{ and } e_1 = 77, d_1 = 72 \text{ such}$$

that $e_1 d_1 \equiv 1 \pmod{241}$. The values e, e_1 are kept public and d, d_1 are kept secret.

Step 12.3: The trusted arbitrator choose the prime numbers $p_1 = 11, p_2 = 17, p_3 = 31, p_4 = 47,$

$p_5 = 101$ and computes the secret key of the group as

$$x = l - 1 = ((p_1 p_2 p_3 p_4 p_5) - 1) \pmod{q} = 14 \text{ and}$$

public key of the group as

$$y = xP = 14P = (116, 114).$$

Step 12.4: Suppose the subgroup B participants $\{P_1, P_2, P_5\}$

wish to sign a message, the private and public keys are distributed by the trusted arbitrator to

these participants as follows.

The trusted arbitrator calculates the symmetric functions

$$a_1 = 202, a_2 = 220, a_3 = 62, a_4 = 191, a_5 = 62.$$

Private key	Public key
$x_1 = (58, 12)$	$y_1 = (185, 199)$
$x_2 = (5, 200)$	$y_2 = (136, 11)$
$x_3 = (13, 111)$	$y_3 = (16, 100)$

Step 12.5: To sign a message m , the participants of group B choose random secret integers

$$k_1 = 38, k_2 = 23, k_5 = 32,$$

Step 12.6: The signers compute $r_1 = k_1 eP = (78, 3),$

$$r_2 = k_2 eP = (146, 84), r_5 = k_5 eP = (48, 119)$$

and broadcast these values among them. These participants calculate

$$R = r_1 + r_2 + r_5 = (60, 96) \text{ and suppose the}$$

hash value of the message m is $h(m, R_x) = 58$

$$\text{with } R_x = 60.$$

Step 12.7: The participants generate the signatures as

$$s_1 = (190, 21), s_2 = (174, 163), s_3 = (124, 92)$$

and send the pair (s_i, r_i) to the trusted arbitrator.

Step 12.8: The trusted arbitrator conform these signatures by checking the equations

$$es_1 = h(m, R_x)y_1 + r_1 = (104, 190)$$

$$es_2 = h(m, R_x)y_2 + r_2 = (133, 48)$$

$$es_3 = h(m, R_x)y_3 + r_3 = (39, 119)$$

Step 12.9: After verifying these signatures the trusted arbitrator generate the group signature as

$$S = (175, 155) \text{ and signs this signature using its private key } d_1 \text{ as } S_1 = (20, 20)$$

13. Conclusion

We proposed two group oriented signature schemes involving symmetric functions. One is based on exponentiation and other on elliptic curves. The group signature is done with the help of a trusted arbitrator. The algorithm is constructed in such a way that every time the k participants depend on the trusted arbitrator to sign a message and the group signature is constructed by the trusted arbitrator. So there is a possibility for the arbitrator himself to forge the verifier and hence he must be a reliable one. We discussed the security analysis related to the both schemes. In future our aim is to construct secured group oriented threshold signature schemes such that the group signature is generated with the assistance of trusted arbitrator.

Appendix

Theorem 1: If $l = p_1 p_2 p_3 \dots p_n$ where $p_1, p_2, p_3, \dots, p_n$ are

prime numbers then $l - 1 = a_1 + a_2 + \dots + a_n$,

where $a_1 = \sum_{i=1}^n (p_i - 1), a_2 = \sum_{1 \leq i < j \leq n} (p_i - 1)(p_j - 1),$

$\dots a_n = (p_1 - 1)(p_2 - 1) \dots (p_n - 1)$

Proof:

Let $l = p_1 p_2 p_3 \dots p_n$

We know that $l = \sum_{d|l} \phi(d)$

The divisors of l are

$p_1, p_2, p_3, \dots, p_n, (p_1 p_2), (p_1 p_3), (p_2 p_3), \dots, (p_{n-1} p_n), \dots, (p_1 p_2 \dots p_n)$

Also $\phi(p) = p - 1$

$\phi(p_1 p_2 p_3 \dots p_n) = (p_1 - 1)(p_2 - 1)(p_3 - 1) \dots (p_n - 1)$

$l = 1 + \sum_{i=1}^n \phi(p_i) + \sum_{1 \leq i < j \leq n} \phi(p_i p_j) + \dots + \phi(p_1 p_2 \dots p_n)$

$= 1 + \sum_{i=1}^n (p_i - 1) + \sum_{1 \leq i < j \leq n} (p_i - 1)(p_j - 1) + \dots + (p_1 - 1)(p_2 - 1) \dots (p_n - 1)$

$= 1 + a_1 + a_2 + \dots + a_n$

Where

$a_1 = \sum_{i=1}^n (p_i - 1), a_2 = \sum_{1 \leq i < j \leq n} (p_i - 1)(p_j - 1), \dots, a_n = (p_1 - 1)(p_2 - 1) \dots (p_n - 1)$

Theorem 2: $\prod_{i \in B} x_i \text{ mod } N = y^{d \text{ mod } v} \text{ mod } N$

Proof: Consider $\prod_{i \in B} x_i \text{ mod } N$

$$\begin{aligned}
 & \left(g^{\left(\sum_{\substack{j \in B \\ j \neq i}} \left(\frac{a_j}{k-1} + \frac{a_j}{k} \right) \right) d \text{ mod } v} \right) \text{ mod } N \\
 &= \prod_{i \in B} \left(g^{\left(\sum_{\substack{j \in B \\ j \neq i}} \left(\frac{a_j}{k-1} + \frac{a_j}{k} \right) \right) d \text{ mod } v} \right) \text{ mod } N \\
 &= \left(g^{\left((k-1) \sum_{j \in B} \frac{a_j}{k-1} + \sum_{j \in B} \frac{ka_j}{k} \right) d \text{ mod } v} \right) \text{ mod } N \\
 &= \left(g^{\left(\sum_{j \in B} a_j + \sum_{j \in B} a_j \right) d \text{ mod } v} \right) \text{ mod } N \\
 &= \left(g^{(a_1 + a_2 + \dots + a_n) d \text{ mod } v} \right) \text{ mod } N \\
 &= g^{((l-1) d \text{ mod } v)} \text{ mod } N
 \end{aligned}$$

$$\begin{aligned}
 &= g^{(xd \text{ mod } v)} \text{ mod } N \\
 &= y^{d \text{ mod } v} \text{ mod } N
 \end{aligned}$$

Theorem 3: $\prod_{i \in B} S_i^e \text{ mod } N = y^{h(m,R)} R \text{ mod } N$

Proof: $\prod_{i \in B} S_i^e \text{ mod } N$
 $= \prod_{i \in B} x_i^{h(m,R)} g^{k_i e} \text{ mod } N$
 $= g^{(l-1)h(m,R)} R \text{ mod } N$
 $= y^{h(m,R)} R \text{ mod } N$

Theorem 4: $\sum_{i \in B} x_i = dy$

Proof: Consider

$$\begin{aligned}
 \sum_{i \in B} x_i &= \sum_{i \in B} \left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j + \frac{1}{k} \sum_{j \in B} a_j \right) dP \\
 &= \sum_{i \in B} \left(\frac{1}{(k-1)} \sum_{j \in B, j \neq i} a_j dP + \frac{1}{k} \sum_{j \in B} a_j dP \right) \\
 &= \left(\sum_{j \in B, j \neq i} a_j dP + \sum_{j \in B} a_j dP \right) \\
 &= (a_1 + a_2 + \dots + a_n) dP \\
 &= (l-1) dP \\
 &= xdP \\
 &= dy
 \end{aligned}$$

Theorem 5: $\sum_{i \in B} es_i = h(m, R_x) y + R$

Proof: $\sum_{i \in B} es_i$
 $= \sum_{i \in B} h(m, R_x) x_i + k eP$
 $= (l-1) h(m, R_x) P + R$
 $= h(m, R_x) xP + R$
 $= h(m, R_x) y + R$

13. References

- [1] Boyd.C, "Digital multisignature", Proceedings of Conference on coding and Cryptography, Cirencester, pp.15-17, 1986.
- [2] Chaum.D and Van Heyst.E, "Group signature", in Advances in Cryptology", Proceedings of Eurocrypt91, pp.257-265, April 1991.
- [3] Desmedt.Y and Frankel.Y, "Shared generation of authenticators and signatures", in 'Advances in Cryptology' Crypto'91, Lecture Notes in Computer Science, pp.457- 469, Springer-Verlag ,1992
- [4] Elgamal. T, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information theory, Vol.31, pp. 469-472, 1985.

- [5] Frankel.Y, "A practical protocol for large group oriented networks" in "Advances in Cryptology", Proceedings of *Auscrypt 92*, Dec. 1992.
- [6] L.Harn, H.Y.Lin and S.Yang,"Threshold cryptosystem with multiple secret sharing policies", *IEEE Proceedings of Computers and Digital techniques*, Vol 141, Issue 2, pp.142-144, March1994.
- [7] L.Harn, "Group oriented (t, n) threshold digital signature scheme and digital multi signatures", *IEEE Proceedings of Computers and Digital techniques*, Vol141, Issue 5, pp.307-313, Sep 1994.
- [8] He Ge, "An effective method to implement group signature with revocation", *International journal of Network security*, Vol.5, No.2, pp.134-139, Sep. 2007.
- [9] Hwang.T. "Cryptosystem for group oriented cryptography", in "Advances in Cryptology", Proceedings of *Crypto 90*, pp.352-360, April 1990.
- [10] Jinn-Ke Jan, Yuh-Min Tseng and Hung-Yu Chien,"Threshold signature schemes withstanding the conspiracy attack", *Communications of Institute of Information and Computing Machinery*, Vol. 2 , No. 3, 1999, pp.31-37.
- [11] Lee. Chang, "An efficient group signature scheme based on the discrete logarithm", *IEEE Proc.comp Digital Techniques*, Vol145 (1), pp.15-18, 1998.
- [12] Li.C, Hwang.T, Lee.N, "Remark on the Threshold RSA signature scheme", "Advances in Cryptology", Proceedings of *Crypto'93*, Lecture Notes in computer Science, Springer Verlag, pp 413-420, 1993.
- [13] Li.C, Hwang.T, Lee.N, "Threshold-multi signature schemes where suspected forgery implies traceability of adversarial Shareholders", *Advances in Cryptology Eurocrypt'94*, Springer, Berlin, pp.194-204, 1994.
- [14] Maged H.Ibrahim, I.A.Ali, I.I.Ibrahim and A.H.El-sawi, "A robust threshold elliptic cuve digital signature providing a new verifiable secret sharing scheme", Proceedings of the 46th *IEEE international Symposium on Circuits and Systems*, Vol11, pp.276-280, Dec 2003.
- [15] Peng Changgen, Li Xiang, "Threshold signcryption scheme based on elliptic curve cryptosystem and verifiable secret sharing", Proceedings of *International conference on Wireless communication, Networking and Mobile computing*, Vol.2,pp.1182-1185, Sep2005.
- [16] Ratna Dutta and Rana Barua,"Password-based encrypted group key agreement", *International Journal of Network Security*, Vol.3, No.1, pp.23-34, July 2006.
- [17] Rivest.R.L, Shamir.A and Adelman.L, "A method for obtaining digital signatures and public key cryptosystem", *Commun of ACM*, 21, (2), pp.120-126, 1978.
- [18] Rongxing Lu and Zhenfu Cao,"Group Oriented based Deniable Authentication Protocol from the Bilinear Pairings", *International Journal of Network Security*, Vol5, No.3, pp.283-287, Nov 2007.
- [19] Shamir.A, "How to share a secret", *Comm. ACM*, 22, pp.612-613, 1979.
- [20] Yu Fang Chung Chia-hui Liu Feipei Lai and Tzer-Shyong Chen, "Threshold signature scheme resistable for conspiracy attack", Proceedings of the *seventh international conference on parallel and distributed computing, Applications and Technologies (PDCAT'06)*, pp.479-483, Dec 2006.
- [21] Y.M.Tseng, J.K.Jan., "Improved group signature based on discrete logarithm", *Electronics Letters*, 35(1) ,pp.37-38, 1999.
- [22] Y.M.Tseng, J.K.Jan," A novel ID-based group signature, International Computer Symposium, Workshop on *Cryptology and Information Security*, Tainan, pp. 159-164, 1998.
- [23] Yu Long and Ke-Fei Chen, "Construction of Dynamic Threshold Decryption scheme from pairing" *International journal of Network Security*, Vol.2,No.2, pp. 111-113, Mar.2006.

Author Biographies

Ms.C.Porkodi is a Senior Lecturer in Department of Mathematics and Computer Applications, PSG College of Technology, Coimbatore, Tamilnadu, India. Currently she is doing Ph.D in Cryptography.

Dr.R.Arumuganathan is a Professor in Department of Mathematics and Computer Applications, PSG College of Technology, Coimbatore, Tamilnadu, India. His research interests are Queueing theory, Stochastic processes, Cryptography and Wavelet theory.