# Cryptosystem for Secret Sharing Scheme with Hierarchical Groups

Atanu Basu[1], Indranil Sengupta[1], and Jamuna Kanta Sing[2]
*(Corresponding author: Atanu Basu)*

Department of Computer Science and Engineering, Indian Institute of Technology[1]
Kharagpur 721302, India.
Department of Computer Science and Engineering, Jadavpur University[2]
Kolkata 700032, India.
(Email: {atanu, isg}@cse.iitkgp.ernet.in, jksing@ieee.org)

*(Received August 09, 2011; revised and accepted Dec. 1, 2011)*

## Abstract

In our proposed scheme, the participants are arranged in a hierarchical structure according to their position or rank and each first level participant as a parent node delegates his power to the lower level hierarchical group members. The group members help to reconstruct the secret shares of their parent nodes in their absence and the secret key is reconstructed even if at least one parent node is present. The secret shares are transmitted between the participants and the trusted dealer through our Elliptic Curve Cryptography (ECC) based signcryption scheme. The formal security analysis shows that our proposed scheme is protected from the adversaries.

*Keywords: Elliptic curve cryptography, hierarchical group, secret sharing, signcyption, unsigncryption*

## 1 Introduction

Shamir [15] and Blakley [5] first proposed secret sharing schemes independently but these schemes [5, 15] cannot survive from malicious participants or adversaries. In a $(t, n)$ threshold secret sharing scheme, a secret key is distributed among $n$ participants, with the property that at least threshold $t$ number of participants or more can reconstruct the secret key. In other words, less than $t$ number of corrupt or malicious participants cannot reconstruct the secret key.

In hierarchical secret sharing schemes [3, 4, 10, 13, 16, 18, 19], the participants of the scheme are arranged in hierarchical levels or multilevels and the number of participants increase down to the bottom level of the hierarchical structure. But, these schemes cannot protect themselves from different types of adversaries and the participants cannot use resource constrained wireless mobile devices.

The motivation of our scheme is to propose a lightweight secured cryptosystem for secret sharing scheme with hierarchical groups where the most powerful members of the hierarchical structure of an organization will initiate the secret key reconstruction process and there should exist some mechanism in which in absence of threshold number of powerful participants, the lower level participants will act on behalf of the absentee or unavailable higher level participants.

In our proposed hierarchical secret sharing scheme, the participants of any organization are arranged into hierarchical groups as shown in Figure 1. Each participant delegates his power to its hierarchical group members where the hierarchical group members are just one level below of that participant. If threshold number of first level participants are unavailable then the corresponding lower hierarchical group members contribute their shares for the construction of secret share of that participant. After collecting and computing secret shares of the unavailable first level participants, the trusted dealer (TD) computes secret key of the participants. All the secret shares are transmitted through our lightweight ECC based signcryption scheme where the digital signature and encryption are done in a single logical step. The verifiable feature of the signcryption scheme helps the recipient to check whether the received share has come from the authorized person or not. The participants of our proposed scheme may use resource constrained wireless mobile devices, e.g. smartphones and the proposed secret sharing scheme protects itself from the active and passive adversaries.

**Organization of the paper.** Section 2 discusses about the related work of the proposed scheme. Section 3 discusses about our signcryption scheme. Section 4 discusses about the proposed secret sharing scheme. The performance analysis and comparison with other schemes have been discussed in Section 5 and Section 6 respectively.

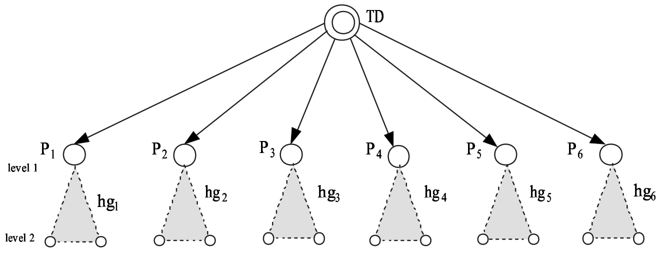

Figure 1: Hierarchical structure of the participants

The security analysis has been done in Section 7 and Section 8 concludes the paper.

# 2 Related Work

Generally, three types of hierarchical secret sharing schemes exist depending on the access structure, i.e. which participants will reconstruct the secret key of the schemes. These are hierarchical secret sharing scheme with weighted access structure, disjunctive access structure and conjunctive access structure.

In the hierarchical secret sharing scheme with weighted access structure [3, 15], the most powerful members possess greater number of shares than the less powerful members. This is the quantitative approach.

In the hierarchical secret sharing scheme with disjunctive access structure [4, 10, 16], the participants are placed in different levels and at least threshold number of participants in a same level can reconstruct the secret key. But, the higher level participants can participate in the process if the lower level participants are less than the threshold number.

In the hierarchical secret sharing scheme with conjunctive access structure [13, 18], the participants are also placed in different levels of the hierarchical structure and secret key reconstruction can start from any level but at least one of the participants from each higher level must be present in the process. In Tassa's [18] scheme which follows qualitative approach with delegation feature, the participants from the higher levels will receive secret shares with lower derivative orders which contain more information than the lower level participants. In this scheme, Birkhoff interpolation is used for the reconstruction of the secret key. Lin et al. [13] proposed an ideal perfect hierarchical secret sharing scheme where separate polynomials are chosen for each hierarchical level. The dealer generates private as well as public shares using the polynomials for each level and distributes those to each participant. Again, the dealer reconstructs the secret key with the help of Lagrange's interpolation using private and public shares of the threshold number of qualified participants.

The multipartite hierarchical secret sharing schemes with both disjunctive and conjunctive access structures were proposed by Tassa et al. [19] and they used bivariate Lagrange's interpolation in their scheme.

# 3 Signcryption Scheme

The signcryption scheme [2, 12, 14, 20, 21, 22, 23] is a paradigm in public key cryptography which incurs lower computational cost and communication overhead compared to the *signature then encryption* scheme. A signcryption scheme consists of a pair of algorithm, i.e. signcryption algorithm and unsigncryption algorithm. When signcryption algorithm operates on a message $m$ of arbitrary length, it produces a signcrypted text and when unsigncryption algorithm operates on the signcrypted text, it recovers the original message $m$ un-ambiguously. Among the ECC based signcryption schemes Zheng et al. [22], Changgen et al. [14], Zhou [23] schemes does not support all the security features while Hwang et al. [12] scheme supports all the security features including public verifiability feature. In our ECC based signcryption scheme [2] while proposing a secured hierarchical secret sharing scheme, we have tried to improve Hwang et al. [12] scheme by further reducing the computational overhead. We have showed the robustness of the scheme through formal security analysis. The signcryption scheme is presented below.

## 3.1 Signcryption and Unsigncryption Algorithms

It has been considered that Alice sends a message $m$ to Bob through the signcryption scheme. A secure elliptic curve $E_p$ over finite field *GF(p)* [6, 11] is chosen where $p$ is a prime number and its base point is $G$ of order $q$ ($q \geq$ 163 bits). Alice's private key $d_A$ is chosen randomly from $[1, q-1]$ and its corresponding public key is $Q_A$ where $Q_A = d_A.G$ which is a point on $E_p$. Similarly, Bob's private key $d_B$ is chosen randomly from $[1, q-1]$ and its corresponding public key is $Q_B$ where $Q_B = d_B.G$ which is a also point on $E_p$.

### 3.1.1 Signcryption Algorithm

**Step 1.** In each signcryption session, Alice chooses a unique random integer, $k \in [1,\ q-1]$ from *GF(p)* and computes ECC points, $T_1^A = k.G = (x_1,\ y_1)$ and $T_2^A = k.Q_B = (x_2,\ y_2)$ where $x_1$ is the $x$-coordinate, $y_1$ is the $y$-coordinate of the point $T_1^A$ and $x_2$ is the $x$-coordinate, $y_2$ is the $y$-coordinate of the corresponding point $T_2^A$ on the elliptic curve $E_p$.

**Step 2.** Alice computes the ciphertext, $c = m.x_2\ mod\ q$.

**Step 3.** After that, Alice computes a one way hash value $h$ by $h = H(m)$ where $H()$ is a one way collision resistant hash function [17].

**Step 4.** Alice computes a digital signature, $s = d_A - k.h$ mod $q$.

Finally, Alice sends the signcrypted message $(c,\ s, T_1^A)$ to Bob through public channel.

### 3.1.2 Unsigncryption Algorithm

After receiving the signcrypted message $(c,\ s, T_1^A)$, Bob recovers the message $m$. Now, Bob verifies validity of the recovered message $m$ whether to accept it or not.

**Step 1.** Bob computes ECC point, $T_1^B = d_B.T_1^A = (x_3, y_3)$ where $x_3$ is the $x$-coordinate and $y_3$ is the $y$-coordinate of the corresponding point $T_1^B$ on the elliptic curve $E_p$.

**Step 2.** After that, Bob computes $m = c.(x_3)^{-1}$ mod $q$. [As, $T_2^A = T_1^B$].

**Step 3.** Bob computes a one way hash value $h$ where $h = H(m)$, $H()$ is a one way collision resistant hash function [17].

**Step 4.** Finally, Bob computes ECC point, $T_2^B = s.G + h.T_1^A$.

A schematic diagram of the signcryption scheme has been shown in Figure 2 below.

If $T_2^B = Q_A$ where $Q_A$ is the public key of the sender Alice, then Bob accepts the message $m$ otherwise rejects the message. This is termed as **Validity test**.

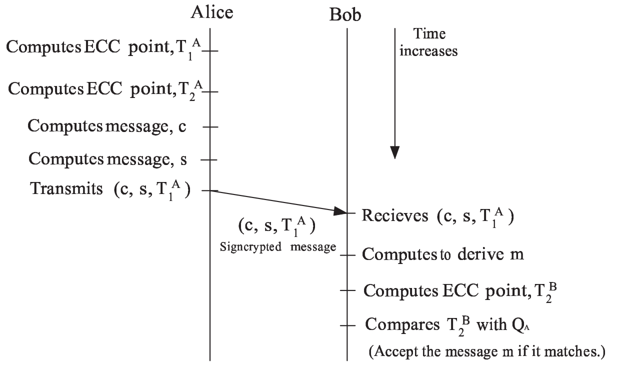

Figure 2: Schematic diagram of the signcryption scheme

**Theorem 1:** *Any receiver accepts the signcrypted message if the unsigncryption of the message passes the validity test.*

**Proof:** It is proved that

$$\begin{aligned} T_1^B &= d_B.T_1^A \\ &= d_B.k.G \\ &= k.(d_B.G) \\ &= k.Q_B \\ &= T_2^A. \end{aligned}$$

Therefore, $T_1^B = T_2^A$.

Again,

$$\begin{aligned} T_2^B &= s.G + h.T_1^A \\ &= (d_A - k.h).G + h.T_1^A \\ &= d_A.G - k.h.G + h.T_1^A \\ &= Q_A - h.(k.G) + h.T_1^A \\ &= Q_A - h.T_1^A + h.T_1^A \\ &= Q_A. \end{aligned}$$

Therefore, $T_2^B = Q_A$.

A comparative study has been shown in Table 1 and Table 2 between our proposed signcryption scheme with other schemes.

## 3.2 The Features of the Signcryption Scheme

The ECPM (elliptic curve point multiplication) operation is the most computational intensive operation among other operations that we have used in our signcryption scheme. The signcryption algorithm uses 2 no. ECPM while the unsigncryption algorithm uses 3 no. ECPM operations. Though in terms of ECC operations, our scheme offers nearly same computational overhead as that of Hwang [12] et al. scheme (Table 2), but our scheme does not use any standard encryption algorithm (DES or AES). This helps to reduce computational overhead further compared to Hwang [12] et al. scheme but at the same time our scheme preserves all the basic security features like authentication, confidentiality, integrity, non-repudiation and forward secrecy efficiently (discussed in Section 7). Our scheme is immune from any plaintext or ciphertext attack as the value $k$ is chosen randomly in each signcryption session. The scheme supports the feaure of public verifiability in case any dispute arises. We have proved through formal security analysis (Section 7) that the signcryption scheme is protected from any type of active and passive attack. So, the signcryption scheme is secured as well as computationally efficient and can be used in resource constrained wireless mobile devices.

# 4 Proposed Hierarchical Secret Sharing Scheme

In this section, we describe the detailed working of our scheme. We first discuss about the system and network model in which the proposed scheme works. After that we discuss about the different adversaries which try to acquire, corrupt the shares of the participants and disrupt the operation of the proposed scheme. Finally, we discuss different steps followed in the proposed scheme.

## 4.1 System and Network Model

The nodes or systems of the TD and the participants $P_1, P_2, \ldots, P_n$ are connected through a network. The par-

Table 1: Comparison of different signcryption schemes based on attributes

| **Schemes** | **CON** | **INT** | **UNF** | **NON** | **FOR** | **VER** |
|---|---|---|---|---|---|---|
| **Zheng [21]** | Yes | Yes | Yes | another scheme | No | No |
| **Zheng [22]** | Yes | Yes | Yes | another scheme | No | No |
| **Hwang [12]** | Yes | Yes | Yes | Directly | Yes | Yes |
| **Zhou [23]** | Yes | Yes | Yes | Directly | No | Yes |
| **Our scheme** | Yes | Yes | Yes | Directly | Yes | Yes |

The abbreviated form of the parameters used in the above Table 1 are CON: Confidentiality, INT: Integrity, UNF: Unforgeability, NON: Non-repudiation, FOR: Forward secrecy, VER: Public verifiability.

Table 2: Comparison of different signcryption schemes based on operations

| Schemes | **Participant** | **ECPM** | **ECPA** | **EXP** | **DIV** | **MUL** | **ADD** | **HASH** |
|---|---|---|---|---|---|---|---|---|
| **Zheng [21]** | Sender | - | - | 1 | 1 | - | 1 | 2 |
| | Receiver | - | - | 2 | - | 2 | 1 | 2 |
| **Zheng [22]** | Sender | 1 | - | - | 1 | 1 | 1 | 2 |
| | Receiver | 2 | 1 | - | - | 2 | - | 2 |
| **Hwang [12]** | Sender | 2 | - | - | - | 1 | 1 | 1 |
| | Receiver | 3 | 1 | - | - | - | - | 1 |
| **Zhou [23]** | Sender | 2 | 2 | - | 1 | 2 | 1 | 3 |
| | Receiver | 4 | 4 | - | - | 1 | 1 | 3 |
| **Our scheme** | Sender | 2 | - | - | - | 2 | 1 | 1 |
| | Receiver | 3 | 1 | - | 1 | 1 | 1 | 1 |

The abbreviated form of the parameters used in the above Table 2 are ECPM: The number of elliptic curve point multiplication operation, ECPA: The number of elliptic curve point addition, EXP: The number of modular exponentiation operation, DIV : The number of modular division or inverse operation, MUL: The number of modular multiplication operation, ADD: The number of modular addition operation, HASH: The number of one way or one way keyed hash function operation.

ticipants may use resource constrained wireless mobile devices, e.g. smartphones whereas the TD uses a server. They work in point-to-point basis and do not broadcast any message. The nodes of the proposed scheme are connected through wired or insecure wireless medium. We assume that the TD will remain trusted to the participants and the private keys of the TD as well as the participants will remain secured throughout the operation of the scheme.

## 4.2 Adversary Model

The **active adversary** may take full control of the system of any participant. In this attack, the adversary may corrupt or delete share of any participant. The adversary may submit fake share by using arbitrary private key of the compromised participant and disrupt the activities of the TD as well as the entire system. The active adversary may capture the secret shares from the wireless medium and submit fake shares to the TD through the **replay attack**. The **passive adversary** may eavesdrop transmitted packets from the insecure wireless medium and try to acquire threshold number of shares of the participants to reconstruct the secret key. In the **collusion attack**, a group of malicious participants may collude to reconstruct the secret key.

## 4.3 Details of the Proposed Scheme

The TD forms the hierarchical structure where the most powerful or important participants, e.g. departmental managers are put in the first level of the hierarchical structure or tree as shown in Figure 1 which shows the 2-level hierarchical tree of the participants. Each first level participant of the hierarchical tree forms a hierarchical group where the group members, e.g. departmental staff members exist just in the lower level of that participant as shown in Figure 3 below. In a hierarchical group, the group leader or parent node delegates his power to its lower level group members or children nodes. This helps to reconstruct the secret key when at least first level threshold number of participants are unavailable for reconstruction of the secret key. The **access structure** of the scheme is such that at least one first level participant must contribute his secret share for reconstruction of the secret key.

### 4.3.1 Setup Phase

The following steps are performed by the TD.

**Step 1.** First, the participants register to the TD when they join in the scheme. The unique identification number, $id_i$ is assigned to each participant, $P_i$ $(1 \leq i \leq n)$ by the TD. The hierarchical groups are formed by the TD. The TD chooses threshold value *threshold* and $threshold_{hg}$ for the first level participants and

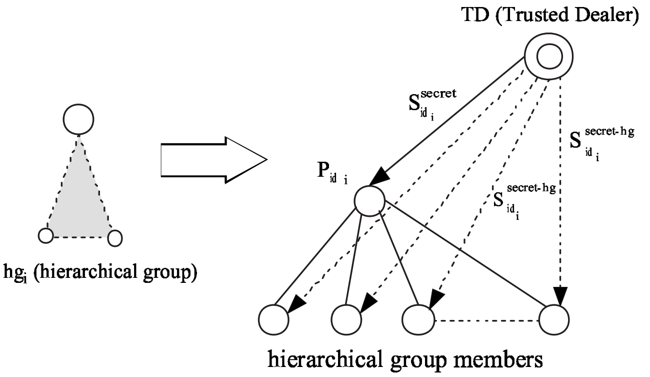

Figure 3: Hierarchical group

the hierarchical groups respectively.

**Step 2.** The TD chooses a secure elliptic curve $E_p$ over finite field *GF(p)* [6, 11] where $p$ is a prime number and the base point $G$ of order $q$ (where $q \geq 163$ bits). After that, the TD chooses its private key $d_{TD}$ where $d_{TD} \in [1,\ q-1]$ and corresponding public key $Q_{TD} = d_{TD}.G$. The TD also selects unique private key $d_i$ where $d_i \in [1,\ q-1]$ and the corresponding public key $Q_i = d_i.G$ for the participants $P_i$ where $1 \leq i \leq n$. The private keys of the participants are distributed securely through a secured channel, e.g. postage system. The private key of the TD or participants may be stored in cryptographic coprocessor or secure chip which are attached to the motherboard of the system of the TD or participants [7].

**Step 3.** The *TD* publishes $p$, $q$, $G$, $Q_{TD}$ and $Q_i$ for $P_i$ where $1 \leq i \leq n$.

### 4.3.2 Generation of Secret Shares

The following steps are followed by the TD.

**Step 1.** The TD chooses a secret polynomial for generation of secret shares as follows

$$f(x) = a_0 + \sum_{i=1}^{threshold-1}(a_i.x^i)\ mod\ q$$

where $a_0 = f(0) = S_{secret}$ is the secret key chosen by the TD.

The coefficients $a_1,\ldots,a_{threshold-1}$ are selected randomly from the finite field, *GF(p)*.

**Step 2.** It computes secret share, $S_{id_i}^{share} = f(id_i)$ as the secret share for first level participants, $P_i$ where $1 \leq i \leq n$.

**Step 3.** Now, the TD chooses secret polynomials for generation of secret shares for the hierarchical group members of each first level participants as follows:

$$f^{hg}(x) = S_{id_i}^{share} + \sum_{i=1}^{threshold_{hg}-1}(a_i^{hg}.x^i)\ mod\ q$$

where $a_0^{hg} = f^{hg}(0) = S_{id_i}^{share}$ is the secret share of the parent node (group leader) of the hierarchical group, $id_i$ is the identification number of the group leader or parent node.

The coefficients $a_1^{hg},\ldots,a_{threshold_{hg}-1}^{hg}$ for each hierarchical group are selected randomly from the finite field *GF(p)*.

It computes secret shares for the group members of each hierarchical group, $S_{id_i}^{share-hg} = f^{hg}(id_i)$ where $id_i$ is the unique identification number of the corresponding hierarchical group members. This procedure for generation of a secret share is termed as $SS_{gen}$.

### 4.3.3 Transfer of Signcrypted Messages

**Step 1.** First, the messages $< share, id_i, S_{id_i}^{share}>$ are transferred to the first level participants of the hierarchical tree through the signcryption algorithm (Section 3.1).

**Step 2.** The messages $< share, id_i, gl\_id_i, S_{id_i}^{share-hg}>$ are transmitted to all the participants of each hierarchical group of the hierarchical tree by the TD through the signcryption algorithm (Section 3.1). The identification number $gl\_id_i$ is the identification number of the group leader or parent node of the corresponding group member nodes.

**Step 3.** The secret polynomials, private keys of the participants $d_i$, the secret shares $S_{id_i}^{share}$ and $S_{id_i}^{share-hg}$, the secret key $S_{secret}$ are erased from the TD. The TD retains the hierarchical structure with the unique identification numbers of the participants and threshold values of each level of the hierarchical tree.

### 4.3.4 Secret Key Reconstruction

The secret key is reconstructed on the server of the TD when reconstruction of the secret key is required. It may happen that the TD does not receive at least threshold number of secret shares from the first level participants of the hierarchical tree as some participants may be unavailable at the time of reconstruction of the secret key. In this situation, the TD follows the following tasks or functions as described below:

**Step 1.** The TD instructs the group members of the unavailable hierarchical group leaders to send their secret shares to the TD. These hierarchical group members send the signcrypted messages which contain the message $< share,\ id_i, gl\_id_i,\ S_{id_i}^{share-hg}>$ to the TD. The TD unsigncrypts and verifies the received messages. After that, the TD reconstructs the secret shares for the unavailable parent node of the hierarchical group. This procedure is termed as $SS_{reconst}$.

**Step 2.** The TD completes the availability of at least threshold number of secret shares of the first level participants and computes the secret key $S_{secret}$ using Lagrange's interpolation. This procedure is termed as $SK_{reconst}$.

# 5 Performance Analysis of the Proposed Scheme

The performance analysis of the scheme is performed in terms of computational and communication overhead.

## 5.1 Computational Overhead

The calculations for computational overhead have been done with some basic mathematical operations as described in Section 3. The ECPM, $SS_{gen}$ (Section 4.3.2), $SS_{reconst}$ and $SK_{reconst}$ (Section 4.3.4) are dominant computations than other mathematical operations we have used in our scheme. We consider $t_{ECPM}$, $t_{ECPA}$, $t_{DIV}$, $t_{MUL}$, $t_{ADD}$, $t_{HASH}$, $t_{EXP}$, $t_{Lagrange}$ are the time required for the computation of the operations of ECPM, ECPA, DIV, MUL, ADD, HASH, EXP, Lagrange's interpolation method respectively.

### 5.1.1 Proposed Signcryption Algorithm

The parameter, $t_{SA}$ is the computational overhead due to signcryption of the secret share by any participant of the hierarchical tree. Then,

$$\begin{aligned} t_{SA} &= 2.t_{ECPM} + 2.t_{MUL} + 1.t_{ADD} + 1.t_{HASH} \\ &\approx 2.t_{ECPM}. \\ &\quad (t_{ECPM} >> \ t_{MUL}, t_{ADD}, t_{HASH}) \end{aligned}$$

### 5.1.2 Proposed Unsigncryption Algorithm

The parameter, $t_{UA}$ is the computational overhead due to unsigncryption of the signcrypted secret share by any participant of the hierarchical tree. Then,

$$\begin{aligned} t_{UA} &= 3.t_{ECPM} + 1.t_{ECPA} + 1.t_{DIV} + 1.t_{MUL} \\ &\quad +1.t_{ADD} + 1.t_{HASH} \\ &\approx 3.t_{ECPM}. \end{aligned}$$

### 5.1.3 Secret Share Generation at the System of TD [Section 4.3.2]

The parameter, $t_{SS_{gen}}$ or $t_{f(id_i)}$ is the computational overhead due to one secret share generation.
Then, $t_{f(id_i)} = \mathit{threshold}.t_{ADD} + (\mathit{threshold} - 1).t_{MUL} + (\mathit{threshold} - 1).t_{EXP}$. Then, total number of secret share generation in TD is $[n.(1 + A)\ .t_{f(id_i)}.2.t_{ECPM}]$ where $A$ is the average no. of participants in a hierarchical group.

### 5.1.4 Secret key reconstruction process when at least threshold number of participants from the first level of the hierarchical tree send secret shares to the TD

If threshold number of first level participants contribute their shares for secret key reconstruction then the computational overhead due to secret key reconstruction by the TD,
$t_{SK_{reconst}} = \mathit{threshold}.$Unsigncryption of the secret shares by the TD + Secret key reconstruction process
$= \mathit{threshold}.t_{UA}$ + time required to $SK_{reconst}$
$\approx \mathit{threshold}.3.t_{ECPM} + t_{Lagrange}.$

### 5.1.5 Secret key reconstruction process when [*threshold* − 1] first level participants are absent

In this worst case scenario, we consider only one first level participant is present (access structure) and $[\mathit{threshold} - 1]$ second level hierarchical groups reconstruct the secret shares on behalf of the unavailable $[\mathit{threshold} - 1]$ number first level participants for the reconstruction of the secret key. Then, the computational overhead in TD due to reconstruction of secret key when $[\mathit{threshold} - 1]$ first level participants are absent,

$$\begin{aligned} t_{SK_{reconst}} &= (\mathit{threshold} - 1)(\mathit{threshold}_{hg}.t_{UA} \\ &\quad + \text{time required to compute } SS_{reconst}) \\ &\quad + \text{time required to compute } SK_{reconst} \\ &= (\mathit{threshold} - 1)(\mathit{threshold}_{hg}.3.t_{ECPM} \\ &\quad + t_{Lagrange}) + t_{Lagrange} \\ &= (\mathit{threshold} - 1)(\mathit{threshold}_{hg}.3.t_{ECPM}) \\ &\quad + \mathit{threshold}.t_{Lagrange} \\ &= (\mathit{threshold} - 1)(\mathit{threshold}.3.t_{ECPM}) \\ &\quad + \mathit{threshold}.t_{Lagrange}. \\ &\quad (\text{if } \mathit{threshold} = \mathit{threshold}_{hg}) \end{aligned}$$

### 5.1.1 Communication Overhead

The communication overhead has been considered in terms of number of transmissions between the TD and the participants.

### 5.2.1 Secret share transfer to the participants

The communication overhead due to secret share transfer for our proposed scheme is $[n.(1 + A)]$.

### 5.2.2 Secret key reconstruction when at least [*threshold* − 1] hierarchical groups transfer their shares to the TD

In this worst case, the communication overhead for secret key reconstruction for our proposed scheme -
$[(\mathit{threshold} - 1).A + 1]$.
The computational and communication overhead for any participant and the TD has been arranged in Table 3 below.

# 6 Comparison with Other Schemes

We compare our scheme with Tassa's [18] scheme as both the schemes follow the conjunctive access structure and delegate power to the lower level participants qualitatively. As Tassa's scheme does not include any security feature, we compare only delegated secret share generation and secret key reconstruction part of our scheme with that of Tassa's scheme. The comparison has been done for hierarchical structure as shown in Figure 1.

Table 3: Computational & communication overhead of the scheme

| | **TD** | **Participant** |
|---|---|---|
| **Comp. overhead: Total $SS_{gen}$** | $n.(1 + A)\ .t_{f(id_i)}.2.t_{ECPM}$ | $3.t_{ECPM}$ |
| $SK_{reconst}$ | $[threshold.3.t_{ECPM} + t_{Lagrange}]$<br>(Average case) | $2.t_{ECPM}$ |
| | $[(threshold$ - $1)(threshold.3.t_{ECPM}) + threshold.t_{Lagrange}]$<br>(Worst case) | |
| **Comm. overhead:**<br>**Secret Share Transfer** | $[n.(1 + A)]$ | |
| **Secret Key Reconst.** | Average case : $[n.(1 + A)]$<br>Worst case : $[(threshold - 1).A + 1]$ | |

## 6.1 Computational Overhead

### 6.1.1 Computational overhead due to delegated secret share generation of our proposed scheme without security feature and Tassa's scheme

If $A$ is the average number of participants in a hierarchical group of any first level participant, then computational overhead due to secret share generation of first level participants and their hierarchical groups,

Total $t_{SS_{gen}}$ $(Proposed\ scheme) = n.t_{f(id_i)} + n.A.t_{f(id_i)}$ $= n.[t_{f(id_i)} + A.t_{f(id_i)}]$ , where $t_{f(id_i)}$ is the time required for the generation of a secret share through polynomial computation.

In Tassa's scheme polynomial derivative has been considered. Then, sum of the computational overhead due to secret share generation,

Total $t_{SS_{gen}}$ $(Tassa's\ scheme)$ = computational overhead due to $f(id_i)$ for first level participants + computational overhead due to *threshold*-th derivative to $f(id_i)$, i.e. $f^{threshold}(id_i)$ for second level participants.

$= n.[t_{f(id_i)} + A.f^{threshold}(id_i)\ ]$.

### 6.1.2 Delegated secret key reconstruction process of our proposed scheme without security feature and Tassa's scheme

The computational overhead due to secret key reconstruction process when at most $[threshold - 1]$ number of first level participants are unavailable,

$t_{SK_{reconst}}(Proposed\ scheme)$ = Computational overhead due to $[threshold - 1]$ number of secret share reconstruction + secret share reconstruction of first level participants

$= (threshold - 1).t_{Lagrange} + t_{Lagrange}$.

$= threshold.t_{Lagrange}$.

The overhead due to secret key reconstruction process for Tassa's scheme,

$t_{SK_{reconst}}(Tassa's\ scheme)$= time required to compute Birkhoff interpolation method.

$= t_{Birkhoff}$.

It is evident that $t_{Birkhoff} > t_{Lagrange}$. The Birkhoff interpolation method may have no solution or the solution may be not unique. So, some precautions must be taken while assigning parameters for the participants from the field.

## 6.2 Communication Overhead

The communication overhead due to secret share transfer and secret key reconstruction are same for our proposed scheme and Tassa's scheme.

A comparison of computational overhead of delegated secret share generation and secret key reconstruction without security feature has been shown in Table 4 below:

# 7 Security Analysis

For the security analysis of our proposed hierarchical secret sharing scheme, first we explain how the basic security features like authentication, confidentiality, integrity, unforgeability, non-repudiation and forward secrecy of the proposed signcryption scheme are preserved through formal security analysis.

We define formally the elliptic curve discrete logarithm problem ($ECDLP$) similar to the discrete logarithm problem ($DLP$) [9].

**Definition 1**: We consider, a secure elliptic curve $E_p$ defined over the finite field $GF(p)$ where $p$ is a prime number and $q$ ($\geq$ 163 bits) is the order of the base point $G$ with a point $\mathcal{O}$ (point at infinity or zero point). Let the two points $P$ and $Q$ ($Q = k.P$ )$\in E_p$ where $k \leftarrow {}_R GF(p)$ which signifies $k$ is chosen randomly from $GF(p)$.

*Instance* : ($P$, $Q$, $l$) for some $k$, $l \leftarrow {}_R GF(p)$.

*Output* : ***Yes*,** if $Q = l.P$, i.e. $k = l$, and output ***No***, otherwise.

We consider, the following two distributions -

$D_{real} = \{k \leftarrow {}_R GF(p),\ X = P,\ Y = Q\ (= k.P),\ W = k\text{: } (X,\ Y,\ W)\}$,

$D_{rand} = \{l,\ k \leftarrow {}_R GF(p),\ X = P,\ Y = Q\ (= k.P),\ W = l\text{: } (X,\ Y,\ W)\}$.

The advantage of any probabilistic, polynomial-time, 0/1-valued distinguisher $D$ in solving $ECDLP$ on $E_p$ is de-

Table 4: Comparison between Tassa's scheme and our scheme

| | **Tassa's Scheme** | **Proposed Scheme** |
|---|---|---|
| **Secret Share Gen. at TD (Computational overhead)** | $n.[t_{f(id_i)} + A.f^{threshold}(id_i)]$ | $n.[t_{f(id_i)} + A.t_{f(id_i)}]$ |
| **Secret Key Reconst. at TD (Computational overhead)** | $t_{Birkhoff}$ | $threshold.t_{Lagrange}$ |
| **Secret Share transfer at TD (Communication overhead)** | $n.(1 + A)$ | $n.(1 + A)$ |
| **Secret Key Reconst. at TD (Communication overhead)** | $(threshold$ - $1).A + 1$ | $(threshold$ - $1).A + 1$ |

fined as $Adv_{D,E_p}^{ECDLP} = |Pr\ [(X,\ Y,\ W) \leftarrow D_{real}\text{: } D(X,\ Y,\ W) = 1] - Pr\ [(X,\ Y,\ W) \leftarrow D_{rand}\text{: } D(X,\ Y,\ W) = 1]|$, where the probability $Pr(.)$ is taken over the random choices of $k$ and $l$. The parameter $D$ is termed to be a ($t$, $\epsilon$)-$ECDLP$ distinguisher for the $E_p$ if $D$ runs at most in time $t$ such that $Adv_{D,E_p}^{ECDLP}(t) \geq \epsilon$.

***ECDLP* assumption:** For every probabilistic, polynomial-time 0/1-valued distinguisher $D$, we must have $Adv_{D,E_p}^{ECDLP}(t) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. Therefore, there exists no ($t$, $\epsilon$)-$ECDLP$ distinguisher for the $E_p$.

Now, we declare two theorems (Theorem 2 and Theorem 3) below:

**Theorem 2:** *Under the elliptic curve discrete logarithm problem (ECDLP) assumption, the proposed signcryption scheme is provably secure against an adversary.*

**Theorem 3:** *As the security of our proposed secret sharing scheme mainly depends on our signcryption scheme, then if the signcryption scheme is secured against the adversaries, our proposed secret sharing scheme is also secured.*

**Proof of Theorem 2:**

We use the method of *proof by contradiction* as proposed by Chuang et al. [8]. We assume that an adversary can solve the $ECDLP$ to find the value $k$ from the points $P$ and $Q$ ($Q = k.P$) $\in E_p$. Now, we define the following oracle - ***Reveal*:** This outputs the value $k$ through the solution of $ECDLP$ by using the points $P$, $Q$ where $Q = k.P$ and other elliptic curve public parameters.

The adversary $A$ executes two algorithms, say $Trial1_{SC,A}^{ECDLP}$ (Algorithm 1) and $Trial2_{SC,A}^{ECDLP}$ (Algorithm 2) for the proposed signcryption scheme $SC$. We define

$$Succ1_{SC,A}^{ECDLP} = Pr[Trial1_{SC,A}^{ECDLP} = 1] - 1$$

as defined by Baek et al. [1]. Then the advantage function for $Trial1_{SC,A}^{ECDLP}$ is defined as

$$Adv1_{SC,A}^{ECDLP}(t, q_R) = max_A\{Succ1_{SC,A}^{ECDLP}\},$$

where the maximum is taken over all $A$ with execution time $t$ and $q_R$ is the number of queries to the *Reveal* oracle. We say that the proposed signcryption scheme provides confidentiality, if $Adv1_{SC,A}^{ECDLP}(t, q_R) \leq \epsilon$, for any sufficiently small $\epsilon > 0$.

Now, we define

$$Succ2_{SC,A}^{ECDLP} = Pr[Trial2_{SC,A}^{ECDLP} = 1] - 1$$

as defined by Baek et al. [1]. Then the advantage function for $Trial2_{SC,A}^{ECDLP}$ is defined as

$$Adv2_{SC,A}^{ECDLP}(t, q_R) = max_A\{Succ2_{SC,A}^{ECDLP}\},$$

where the maximum is taken over all $A$ with execution time $t$ and $q_R$ is the number of queries to the *Reveal* oracle. We say that the proposed signcryption scheme preserves security features like authentication, integrity (replay or man-in-the-middle attack), unforgeability, non-repudiation as well as forward secrecy, if $Adv2_{SC,A}^{ECDLP}(t, q_R) \leq \epsilon$, for any sufficiently small $\epsilon > 0$.

**Algorithm 1** : $Trial1_{SC,A}^{ECDLP}$

```
Capture the signcrypted message ($c$, $s$, $T_1^A$).
Call Reveal oracle. Let $k \leftarrow$ Reveal($E_p$, $G$, $T_1^A$).
Using the value $k$, compute $(x_2, y_2) = k.Q_B$.
Retrieve the original message $m$ as $m = c.x_2^{-1}\ mod\ q$.
```

**Algorithm 2** : $Trial2_{SC,A}^{ECDLP}$

```
Capture the signcrypted message ($c$, s, $T_1^A$).
Call Reveal oracle. Let $k \leftarrow$ Reveal($E_p$, $G$, $T_1^A$).
Using the value $k$, compute $(x_2, y_2) = k.Q_B$ and retrieve the original message $m$ as $m = x_2^{-1}\ mod\ q$.
Change $m$ to $m'$ and compute $h' = H(m')$. Compute $c' = m'.x_2\ mod\ q$.
Call Reveal oracle. Let $d_A \leftarrow$ Reveal($E_p$, $G$, $Q_A$).
Choose a random integer $k' \in [1, q - 1]$.
Compute $s' = d_A - k'.h' mod\ q$ and $T_1^{A'} = k'.G$.
Send ($c'$, $s'$, $T_1^{A'}$) to the verifier.
Verifier checks if $T_2^B = s'.G + h'.T_1^{A'} = Q_A$, where $h' = H(m')$ and $m'$ is computed by the verifier.
If the verification satisfies then
    return 1
else
    return 0
end if
```

According to $Trial1_{SC,A}^{ECDLP}$, the adversary is able to compute $k$ from $T_1^A$ and thus compute the original message. However, it is a contradiction due to the computational difficulty of the $ECDLP$. Thus, $Adv1_{SC,A}^{ECDLP}(t, q_R) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. Hence, if any attacker captures the signcrypted message ($c$, $s$, $T_1^A$ ), he cannot compute the parameter $k$ from $T_1^A = k.G$ (Step 1 of Section 3.1.1) due to the computational difficulty of the $ECDLP$. Therefore, the signcryption scheme provides **confidentiality** feature.

According to $Trial2_{SC,A}^{ECDLP}$, the adversary is able to compute $k$ and $d_A$. The adversary has thus the ability to change the original message $m$ and the value of $s$ so that the verifier does not have any ability to verify whether the message is fake or not. In this case, the verifier always be able to verify the condition $T_2^B = Q_A$. This proves that the message has come from the right person which proves **authentication** feature. However, it is again a contradiction due to the computational difficulty of the $ECDLP$. Thus, $Adv2_{SC,A}^{ECDLP}(t, q_R) \leq \epsilon$, for any sufficiently small $\epsilon > 0$. Since the attacker does not have any ability to change the original message $m$, the values $s$ and $d_A$, the adversary is not able to perform replay or main-in-the-middle attack. For unforgeability, after eavesdropping of the signcrypted message ($c$, s, $T_1^A$ ), if any attacker wants to forge the message ($c$, $s$, $T_1^A$ ) to ($c'$, $s'$, $T_1^{A'}$ ), he needs to get the private key $d_A$ of the sender, Alice and the randomly chosen value $k$. However, these are not possible due to difficulty of $ECDLP$. As a result, the scheme provides **unforgeability** feature.

For non-repudiation, if the sender Alice denies that she has not sent the signcrypted message ($c$, $s$, $T_1^A$ ) to receiver Bob, then any third party can compute the verification condition $T_2^B = Q_A$ using the public key $Q_A$ of the sender Alice. However, if the condition $T_2^B = Q_A$ satisfies, that ensures that the message has indeed come from the sender Alice and later she cannot deny that she has not sent the message. Thus, the signcryption scheme provides **non-repudiation**.

Finally, for forward secrecy, even if the adversary possesses the private key $d_A$ of Alice at later stage, he cannot recover the previously sent signcrypted messages because he has to get the value $k$ and retrieving the value $k$ is difficult due to $ECDLP$. As a result, the adversary is not able to recover the previous original messages and thus, the **forward secrecy** feature of the scheme is preserved.

Therefore, it is proved that our signcryption scheme preserves the basic security features like authentication, confidentiality, integrity, unforgeability, non-repudiation and forward secrecy.

**Proof of Theorem 3:** It has been proved in **Theorem 2** that the signcryption scheme preserves all the basic security features. As the security of our proposed secret sharing scheme depends on the signcryption scheme, so it is proved that our proposed secret sharing scheme is also secured against the adversaries.

The proposed secret sharing scheme is protected from any **plaintext** or **ciphertext** attack as in each signcryption session the value $k$ is chosen randomly (Section 3.1) where the key to signcrypt a secret share is derived from $k$.

It is also not possible practically to reconstruct the threshold number of secret shares of first level participants to reconstruct the secret key by the hierarchical group members of second level participants as collusion from each threshold number of group members from each group is not possible. Thus, any type of **collusion attack** by the malicious participants is prevented in this scheme.

So, it can be concluded that our proposed hierarchical secret sharing scheme is secured from any type of attack from the adversaries.

## 8 Conclusion

Shamir's [15] secret sharing scheme is a special case of our proposed secret sharing scheme with security features when threshold number of first level participants are present for secret key reconstruction. Our proposed scheme discusses about the secret sharing scheme in hierarchical groups where the parent node or group leader delegates his power to his hierarchical group members for reconstructing his secret share when he will be unavailable. Our proposed scheme also discusses about how the scheme is protected from different types of adversaries using the lightweight ECC based signcryption scheme through formal security analysis. At least one participation from the most powerful members, e.g. vice president or manager of the first level of the hierarchical tree is mandatory for the secret key reconstruction. This is useful in bank scenario where the powerful members or mixture of at least one powerful member and lower level members can reconstruct the secret key to sign or authorize the electronic fund transfers.

## References

[1] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption," *Journal of Cryptology*, vol. 20, no. 2, pp. 203–235, 2007.

[2] A. Basu, I. Sengupta, and J. K. Sing, "Secured hierarchical secret sharing using ecc based signcryption," *Security and Communication Networks, Wiley, DOI: 10.1002/sec.370*, 2011.

[3] A. Beimel, T. Tassa, and E. Weinreb, "Characterizing ideal weighted threshold secret sharing," in *Proceedings of The Second Theory of Cryptography Conference, (TCC 2005), LNCS*, pp. 600–619, Cambridge, MA, USA, February 2005.

[4] M. Belenkiy. "Disjunctive multi-level secret sharing,". tech. rep., Brown University, January 2008.

[5] G. Blakley, "Safeguarding cryptographic keys," in *Proceedings of The National Computer Conference (AFIPS 1979)*, pp. 313–317, 1979.

[6] D. R. L. Brown. "Sec1: Elliptic curve cryptography,". Tech. Rep. 2.0, Certicom Research, May 2009.

[7] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, p. 398, 2002.

[8] Y. H. Chuang and Y. M. Tseng, "An efficient dynamic group key agreement protocol for imbalanced wireless networks," *International Journal of Network Management, Wiley*, vol. 20, no. 4, pp. 167–180, 2010.

[9] R. Dutta and R. Barua, "Provably secure constant round contributory group key agreement," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2007–2025, May 2008.

[10] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini, "Secret sharing in multilevel and compartmented groups," in *Proceedings of The Third Australasian Conference on Information Security and Privacy (ACISP'98)*, pp. 367–378, Brisbane, Australia, July 1998.

[11] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography.* Springer, 2004.

[12] R. J. Hwang, C. H. Lai, and F. F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve," *Applied Mathematics and Computation, Elsevier Inc.*, vol. 167, no. 2, pp. 870–881, 2005.

[13] C. Lin, L. Harn, and D. Ye, "Ideal perfect multilevel threshold secret sharing scheme," in *Proceedings of The Fifth International Conference on Information Assurance and Security (IAS 2009), IEEE*, pp. 118–121, Xian, China, Aug. 2009.

[14] C. Peng and X. Li, "Threshold signcryption scheme based on elliptic curve cryptosystem and verifiable secret sharing," in *Proceedings of The International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1182–1185, Sept. 2005.

[15] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[16] G. J. Simmons, "How to (really) share a secret," in *Proceedings of The 8th Annual International Cryptology Conference (Crypto'88)*, pp. 390–448, California, USA, August 1988.

[17] W. Stallings, *Cryptography and Network Security : Principles and Practices (4th Edition).* Pearson Prentice Hall, 2006.

[18] T. Tassa, "Hierarchical threshold secret sharing," *Journal of Cryptology*, vol. 20, no. 2, pp. 237–264, 2007.

[19] T. Tassa and N. Dyn, "Multipartite secret sharing by bivariate interpolation," *Journal of Cryptology*, vol. 22, no. 2, pp. 227–258, 2009.

[20] M. Toorani and A. A. B. Shirazi, "Cryptanalysis of an efficient signcryption scheme with forward secrecy based on elliptic curve," *International Journal of Network Security*, vol. 10, no. 1, pp. 51–56, 2010.

[21] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption)," in *Proceedings of The 17th Annual International Cryptology Conference (The Advances in Cryptology - CRYPTO '97), Springer*, pp. 165–179, Santa Barbara, California, USA, August 1997.

[22] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Information Processing Letters*, vol. 68, no. 5, pp. 227–233, 1998.

[23] X. Zhou, "Improved signcryption scheme with public verifiability," in *Proceedings of The Pacific-Asia Conference on Knowledge Engineering and Software Engineering, IEEE*, pp. 178–181, Shenzhen, China, December 2009.

**Atanu Basu** is currently pursuing Ph.D. in Computer Sc. & Engg. at Jadavpur University, Kolkata (India). He received Masters in Science (M.S.) in Computer Sc. & Engg. from the Dept. of Computer Sc. & Engg., Indian Institute of Technology, Kharagpur (India). He is currently working as Technical Superintendent in the Dept. of Computer Sc. & Engg., Indian Institute of Technology, Kharagpur (India). His main research interest includes Network Security and Cryptography, Cloud computing, Sensor and Mobile Ad Hoc networks.

**Indranil Sengupta** is a professor of Department of Computer Science & Engg. in the Indian Institute of Technology, Kharagpur (India). He received his Bachelors, Masters and Doctorate degrees in Computer Science from University of Calcutta, India. His research interests are primarily in the field of Information Assurance, Cryptography & Network Security, Testing & fault diagnosis and CAD for VLSI. He has published more than 90 research articles in leading journals, conference proceedings and books including ACM Transactions, IEEE, JCC and JSA. He serves in editorial boards of several International Journals and has served in program committees of several international conferences. He has two decades of rich experience in teaching and research. He has been continuously steering research projects of national importance.

**Jamuna Kanta Sing** is working as Reader in the Dept. of Computer Sc. & Engg., Jadavpur University, Kolkata (India). He received his Ph.D. in Computer Sc. & Engg. from the Jadavpur University, Kolkata (India). He has receiced M.Tech. in computer sc. & engg. from IIT Kharagpur (India) and his bachelor degree from the Dept. of Computer Sc. & Engg., Jadavpur University, Kolkata (India). He has done his post doctoral work in Dept. of Electrical and Computer Engineering, University of Iowa, Iowa City and Medical Image Processing Group, Dept. of Radiology, University of Pennsylvania, Philadelphia, USA. His main research interest is in human Face Detection, recognition and tracking. He is also working in medical image analysis. He has published many research articles in leading journals, conference proceedings. He has completed and continuing many research projects.