A Reusable Multipartite Secret Sharing Scheme Based on Superincreasing Sequence

Putla Harsha, Patibandla Chanakya, and Vadlamudi China Venkaiah
(Corresponding author: Patibandla Chanakya)

School of Computer and Information Sciences & University of Hyderabad
Hyderabad-500046, India
(Email: chanakya.patibandla@gmail.com)
(Received Nov. 1, 2016; revised and accepted Feb. 20, 2017)

Abstract

A new multipartite secret sharing scheme that uses a super increasing sequence is proposed in this paper. Novelty of the scheme is that, apart from being a multipartite scheme, it realizes the level ordered access structure [17]. Also, the proposed scheme is reusable in that the shares of the participants need not be replenished for a new secret after the reconstruction the current secret.

Keywords: Multipartite Secret Sharing; Secret Sharing Schemes; Superincreasing Sequence

1 Introduction

A *Secret Sharing Scheme* consists of two phases: share distribution and secret reconstruction. In share distribution phase, each participant is given a share of the secret. In secret reconstruction phase, authorized subsets of the set of participants collaboratively recover the secret from their shares. A secret sharing scheme is said to be perfect if an unauthorized subset gets no information about the secret and an authorized subset has complete information of the secret in the information theoretic sense. The set of authorized subsets is called an access structure. If the maximal length of the shares and the length of the secret are identical, then such secret sharing scheme is said to be ideal. Several access structures like (t, n)-threshold access structure, Multipartite access structure, Generalized access structure *etc.*, are proposed in the literature.

A $(t, n)-$*Threshold Secret Sharing Scheme* is one in which at least the threshold (t) number of participants participate in secret reconstruction phase to get the secret. In threshold secret sharing scheme, all participants are given equal priority and entrust. Threshold secret sharing schemes were first proposed independently by Adi Shamir [18] and George Blakley [2] in 1979.

In practice, participants may be served with different priorities. So each participant is assigned a level. Based on the level of the participant, the priority, entrust of the participant varies. In general, the participants in the higher level get higher priority and entrust over the participants in the lower level.

In *Multipartite Secret Sharing*, the set of participants are divided into disjoint levels (parts/ compartments) and all participants in a particular level/compartment play the same role as all other participants in that level/compartment. Compartmented threshold secret sharing and multilevel threshold secret sharing are multipartite in nature. Ghodosi *et al.* [9] proposed an ideal and perfect secret sharing schemes for multilevel and compartmented groups. The scheme we are proposing is a multipartite scheme along with a restriction that the low level participants can recover the implicit compartment (level) secret bit assigned to them only if the implicit secret bits assigned to all higher levels are already recovered. The actual secret of our multipartite scheme can be recovered only after completion of the recovery of all the compartment secrets. Compartment secrets can be concurrently reconstructed,but in order to reconstruct the actual secret bit vector/tuple, hierarchy has to be followed. We use Shamir's secret sharing scheme for each compartment to distribute and recover the compartment secret to all the participants of the compartment. Applications of secret sharing can be found in [1, 5, 14, 15, 21] *etc.*

Our scheme supports the property of secret changeability without changing the compartment secret shares. This property makes our scheme secure and robust. All this is possible because the actual secret is not a function of compartment shares alone. That means, the secret can be changed by the dealer any number of times without changing compartment secrets. This property makes our scheme to be reusable in nature and hence avoids communication intense phase of share distribution to the participants whenever the secret is changed. Whenever the secret is changed, the dealer publishes a public value based on the secret and compartment shares. Then with the help of the same compartment secret shares along with the current public value, the compartments together can get the original secret. For each change in secret, the

compartments together has to reconstruct the current secret based on their secret compartment shares and the current public value, since the old secret is no more valid.

2 Related Work

(t, n)-threshold secret sharing schemes proposed by Adi shamir [18] and George Blakley [2] do not support any hierarchy among the participants, but Shamir mentioned that a hierarchical variant of (t, n)-threshold secret sharing scheme can be introduced by assigning larger number of shares to higher level participants. But this will not be ideal. Multipartite secret sharing was introduced by Simmons [3]. Multipartite secret sharing schemes were studied based on different assumptions by different authors *eg* [3, 4, 6–8, 11–13]. Multipartite Secret Sharing schemes by bivariate interpolation are developed by T. Tassa and N. Dyn [19]. Multilevel secret sharing scheme based on the Chinese remainder theorem was discussed in [10]. Hierarchical secret sharing schemes based on MDS codes was proposed in [20].

Secret sharing in multilevel and compartmented groups are discussed in [9]. Our scheme is influenced from the idea of compartmented scheme discussed in [9] in which the global threshold is equal to the sum of all individual compartment thresholds. The scheme, proposed in this paper uses superincreasing sequence, allows concurrency during compartmental share reconstruction. But secret reconstruction is strictly hierarchical in nature, *i.e.*, lower level can get the secret bit(s) information only if all higher levels get their secret bit(s).

The organization of the paper is as follows: In Section 3, the background required like knapsack problem (restricted to our scenario), algorithms related to superincreasing sequence are described and describe concisely the Shamir's secret sharing scheme. In Section 4, multipartite secret sharing scheme based on the superincreasing sequence was proposed. In Section 5, the scheme was explained with an example. In Section 6, the property of secret changeability with an example was explained. In Section 7, a brief note on the security of the scheme in the presence of an intruder and some observations are discussed. Finally concluding remarks are in Section 8.

3 Background

This section reviews the knapsack problem, corresponding instances, algorithms, and the Shamir's threshold secret sharing scheme.

3.1 Knapsack Functions

If the elements inside a knapsack are known, then it is easy to compute the sum of the elements inside the knapsack. But if the sum of the numbers inside the knapsack is only known, it is difficult to list out the numbers inside the knapsack.

Mathematically this observation can be stated as follows:

Let *bagsum* be a function that takes two r-tuples as input and returns an integer value as output. Let $w = [w_1, w_2, w_3, \cdots, w_r]$ and $s = [s_1, s_2, s_3, \cdots, s_r]$ be two r-tuples. The second tuple s contains Boolean elements only *i.e.*, $s_i = 0$ or 1 for $i \in \{1, 2, 3, \cdots, r\}$. $sum = bagsum(w, s) = \sum_{i=1}^{r} w_i s_i = w_1 s_1 + w_2 s_2 + w_3 s_3 + \cdots + w_r s_r$.

Let *bagsum_inverse* be a function that takes the sum value and the first tuple w as input and returns the second Boolean tuple s as output i.e., $s = bagsum\_inverse(sum, w)$.

Given w and s it is easy to find *bagsum*, but given *sum* and w it is difficult to find s. But if the first tuple w contains the superincreasing sequence then it is easy to compute both *bagsum* and *bagsum_inverse* [16].

3.2 Superincreasing Sequence

A sequence is said to be *superincreasing*, if every element (except first) in the sequence is greater than or equal to the sum of all its previous elements. A tuple is said to be superincreasing, if it contains a superincreasing sequence. Thus a tuple $w = [w_1, w_2, w_3, \cdots w_r]$ is superincreasing if and only if $w_j \geq w_1 + w_2 + w_3 + \cdots w_{j-1}$ for $2 \leq j \leq r$.

The pseudo codes of *bagsum* and *bagsum_inverse* for superincreasing sequence are as follows:

Algorithm 1: *bagsum* function

```
1 Function bagsum (w,s);
  Input  : Two tuples: w = [w_1, w_2, w_3, ··· w_r] and
           s = [s_1, s_2, s_3, ··· s_r].
  Output: sum: an integer.
2 sum ← 0;
3 for i = 1 ··· r do
4 |  sum ← sum + w_i × s_i ;
5 end
6 Return sum;
```

The running time of *bagsum* is $\mathcal{O}(r)$. Applied to the superincreasing sequence, in our scheme, the dealer uses this function.

The running time of *bagsum_inverse* applies to the superincreasing sequence (Algorithm 2) is same as *bagsum* *i.e.*, $\mathcal{O}(r)$. In our scheme, this algorithm is implicitly used by compartments during secret reconstruction phase.

Let us define another function *exor*, which takes two binary tuples (bit arrays) of same size as input and performs bit-wise exclusive or of them and outputs the resultant tuple. Let b, b' be two bit tuples of same size, then $c = exor(b, b')$ such that $c_i = b_i \oplus b'_i$, where c_i, b_i, b'_i is the i^{th} bit in the tuples c, b, b' respectively. We use *exor* function to improve security of our scheme by means of hiding the original secret tuple from compartments.

Algorithm 2: *bagsum_inverse* Function for superincreasing tuple

```
Function bagsum_inverse (sum, w);
  Input  : An integer sum and tuple
           w = [w_1, w_2, w_3, ··· w_r].
  Output: Boolean Tuple: s = [s_1, s_2, s_3 ··· s_r].
for i = r ··· 1 do
    if sum ≥ w_i then
        s_i ← 1;
        sum ← sum − w_i;
    else
        s_i ← 0
    end
    Return [s_1, s_2, s_3 ··· s_r];
end
```

3.3 Shamir's Secret Sharing Scheme

Shamir's secret sharing scheme is a threshold secret sharing scheme. It has two phases: share distribution phase and secret reconstruction phase. In share distribution phase the dealer calculates shares and distributes them securely to the participants. In secret reconstruction phase at least threshold number of participants cooperatively participate in interpolation to arrive at the corresponding Lagrange polynomial and recovers the secret. In Shamir's secret sharing scheme dealer uses a polynomial and public identities of the participants to calculate the share values.

Let n be the number, $\{P_1, P_2, P_3, \cdots P_n\}$ be the participants and $t(\leq n)$ be the threshold. Let id_i be the public identity of the participant P_i for $i \in [1, n]$, where $[1, n]$ denote the set $\{1, 2, 3, \cdots, n\}$. Then the Shamir's secret sharing scheme is as follows:

3.3.1 Share Distribution Phase

Dealer performs the following steps:

1) Selects a prime number $q > n$;
2) Selects a secret $s \in \mathbb{Z}_q^*$;
3) Generates a polynomial $f(x) = s + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ of degree atmost $t-1$ with coefficients in $a_i \in \mathbb{Z}_q^*$ for $1 \leq i \leq t-1$;
4) Calculates $[s]_i = f(id_i) \mod q$ for $1 \leq i \leq n$;
5) Sends $[s]_i$ securely to the participant P_i for $1 \leq i \leq n$.

3.3.2 Secret Reconstruction Phase

- At least t participants are required to cooperate to reconstruct the secret; which can be done using Lagrange polynomial interpolation formula as follows:

$$s = \sum_{i=1}^{t} \left(\prod_{j \neq i} \frac{id_j}{id_j - id_i} \right) [s]_i \mod q.$$

In our proposed scheme, Shamir's secret sharing scheme is used to share the compartment secret and to reconstruct it.

4 Multipartite Secret Sharing Scheme

Our multipartite secret sharing scheme, based on superincreasing sequence, consists of share distribution phase followed by secret reconstruction phase. After presenting pseudo-code, we explain the algorithm with an example. Let $\mathfrak{P}$ be the set of all participants and let the participants be divided into ℓ disjoint levels. $\mathfrak{P}_i$ be the set of participants at level i. n_i be the total number of participants at level i and $t_i \leq n_i$ be its corresponding compartment (level) threshold. Note that the global threshold t for this scheme is the sum of all the individual level thresholds.

4.1 Share Distribution:

Dealer performs the following steps:

1) Selects a secret $s \in \mathbb{Z}_{2^{\ell-1}} - \{0\}$;
2) Converts the secret s into $\ell - 1$ bit binary number $s_{\ell-1}s_{\ell-2}s_{\ell-3} \cdots s_3s_2s_1$ and forms the secret tuple $st = [s_{\ell-1}, s_{\ell-2}, s_{\ell-3} \cdots s_3, s_2, s_1]$;
3) Generates a random bit tuple $'temp'$ of length $\ell - 1, temp = [e_{\ell-1}, e_{\ell-2}, e_{\ell-3} \cdots e_3, e_2, e_1]$;
4) Calculates $st' = exor(st, temp)$, let $st' = [s'_{\ell-1}, s'_{\ell-2}, s'_{\ell-3} \cdots s'_3, s'_2, s'_1]$;
5) Generates a superincreasing tuple w of size $\ell-1, w = [w_{\ell-1}, w_{\ell-2}, w_{\ell-3} \cdots w_3, w_2, w_1]$;
6) Calculates $u = bagsum(st', w)$ and makes u public;
7) Select a prime q, which is greater than the sum $w_1 + w_2 + w_3 + \cdots w_{\ell-1}$ and $max(n_i)$ for $1 \leq i \leq \ell$, and make q public;
8) Distributes shares of w_i to all the participants in level i using Shamir's (t_i, n_i) share distribution for $1 \leq i \leq \ell - 1$ and distribute the shares of decimal equivalent of bit tuple $temp$ to level ℓ by means of Shamir's (t_ℓ, n_ℓ) share distribution;

Note:

- q, u is public. $s, st, temp, st', w$ are not public, but known to dealer.
- In simple and informal language, the share distribution phase can be summarized as: st is the binary tuple for s. $st' = st \oplus temp$.
- Since $temp$ is distributed to the compartment ℓ, eventhough remaining $\ell - 1$ levels gets the associated bit values, they cannot get secret tuple st.

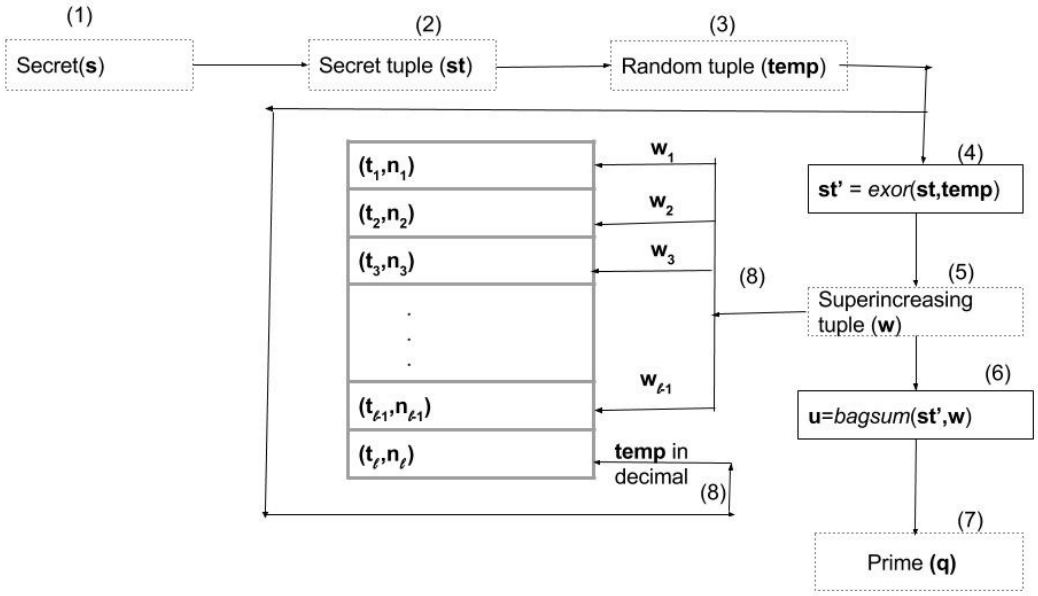

Figure 1: Stepwise pictorial representation of share distribution

4.2 Secret Reconstruction

1) At least t_i participants of level i perform (t_i, n_i) Shamir's secret reconstruction and gets the level share w_i for $1 \leq i \leq \ell - 1$;

2) Level 1 checks whether $u \geq w_1$. If true, then Level 1 outputs bit 1 and sends $u'_{1,2} = u - w_1$ to level 2. Else it outputs bit 0 and sends $u'_{1,2} = u$ to level 2 and appends its output bit (which is s'_1) to an empty tuple (say st'');

3) For $2 \leq i \leq \ell - 1$, Level i checks whether $u'_{i-1,i} \geq w_i$. If true, then level i outputs bit 1 and sends $u'_{i,i+1} = u'_{i-1,i} - w_i$ to level $i+1$. Else outputs bit 0 and sends $u'_{i,i+1} = u'_{i-1,i}$ and appends the output bit (which is s'_i) to the starting index of the tuple st'';

4) Level ℓ performs (t_ℓ, n_ℓ) Shamir's secret reconstruction and converts the result into $\ell - 1$ bit tuple, which is $temp = [e_{\ell-1}, e_{\ell-2}, e_{\ell-3} \cdots e_3, e_2, e_1]$. (Observe that $st' = st''$ and st'' is not public). level ℓ performs $exor(temp, st')$ which results in st;

5) Finally the secret binary tuple st is obtained, which can be converted to decimal to obtain the secret s.

Note: Steps 2 and 3 corresponds to the *bagsum_inverse* algorithm discussed earlier.

5 Explanation with an Example

Let the number of levels (ℓ) be 6. Also let the threshold values and total number of participants for each level be as follows: $(t_1, n_1) = (2, 5)$, $(t_2, n_2) = (3, 5)$, $(t_3, n_3) = (4, 5)$, $(t_4, n_4) = (5, 6)$, $(t_5, n_5) = (3, 6)$, $(t_6, n_6) = (5, 7)$. Let $1, 2, 3 \cdots n_i$ be the public identities of the participants in compartment i.

5.1 Share Distribution:

Dealer performs the following steps:

1) Selects a secret $s \in \mathbb{Z}_{2^{\ell-1}} - \{0\}$;

$$\textbf{Since } \boldsymbol{\ell = 6}\textbf{, } \boldsymbol{\mathbb{Z}_{2^{\ell-1}} - \{0\} = \{1, 2, 3 \cdots 31\}}\textbf{. Let } \boldsymbol{s = 21}\textbf{.}$$

2) Converts the secret s into $\ell - 1$ bit binary number $s_{\ell-1}s_{\ell-2}s_{\ell-3} \cdots s_3 s_2 s_1$ and forms secret tuple $st = [s_{\ell-1}, s_{\ell-2}, s_{\ell-3} \cdots s_3, s_2, s_1]$;

$$\boldsymbol{st = [1, 0, 1, 0, 1]}$$

3) Generates a random bit array(tuple) $'temp'$ of length $\ell - 1, temp = [e_{\ell-1}, e_{\ell-2}, e_{\ell-3} \cdots e_3, e_2, e_1]$;

$$\boldsymbol{temp = [0, 1, 1, 1, 0]}$$

4) Calculates $st' = exor(st, temp)$, let $st' = [s'_{\ell-1}, s'_{\ell-2}, s'_{\ell-3} \cdots s'_3, s'_2, s'_1]$;

$$\boldsymbol{st' = exor(st, temp) =}$$
$$\boldsymbol{exor([1, 0, 1, 0, 1], [0, 1, 1, 1, 0]) = [1, 1, 0, 1, 1]}$$

5) Generates a superincreasing tuple w of size $\ell - 1, w = [w_{\ell-1}, w_{\ell-2}, w_{\ell-3} \cdots w_3, w_2, w_1]$;

$$\textbf{Let } \boldsymbol{w = [10, 11, 30, 60, 130]}$$

Table 1: Shamir's share distribution at compartments

Level number	$(\mathbf{t_i}, \mathbf{n_i})$	Dealer polynomial	Participants share list (Public identity, Share value)
1	(2,5)	$130 + x$	$(1, 131), (2, 132), (3, 133), (4, 134), (5, 135)$
2	(3,5)	$60 + 2x + 3x^2$	$(1, 65), (2, 76), (3, 93), (4, 116), (5, 145)$
3	(4,5)	$30 + 4x + x^3$	$(1, 35), (2, 46), (3, 69), (4, 110), (5, 175)$
4	(5,6)	$11 + 2x + 4x^3 + 3x^4$	$(1, 20), (2, 95), (3, 368), (4, 502), (5, 232), (6, 447)$
5	(3,6)	$10 + x^2$	$(1, 11), (2, 14), (3, 19), (4, 26), (5, 35), (6, 46)$
6	(5,7)	$14 + x + x^2 + x^3 + 16x^4$	$(1, 33), (2, 284), (3, 267), (4, 407), (5, 431), (6, 450), (7, 418)$

6) Calculates $u = bagsum(st', w)$ and makes u public;

$$\boldsymbol{u = bagsum(st', w) = bagsum([1, 1, 0, 1, 1], [10, 11, 30, 60, 130]) = 10 + 11 + 60 + 130 = 211}$$

7) Select a prime q, which is greater than the sum $w_1 + w_2 + w_3 + \cdots w_{\ell-1}$ and $max(n_i)$ for $1 \leq i \leq \ell$ make u as public;

$$\boldsymbol{w_1 + w_2 + w_3 + w_4 + w_5 = 10 + 11 + 30 + 60 + 130 = 241,\ max(n_i) = 7,\ \text{let } q = 541}$$

8) Distributes shares of w_i to all the participants in level i using Shamir's (t_i, n_i) share distribution for $\ell - 1 \leq i \leq 1$ and distribute the shares of decimal equivalent of bit tuple $temp$ to level ℓ by means of Shamir's (t_ℓ, n_ℓ) share distribution;

We have the compartmental shares given in Table 1.

5.2 Secret Reconstruction

1) Performing (t_i, n_i), $1 \leq i \leq \ell - 1$ Shamir's secret reconstruction, we have the compartmental shares given in Table 2.

2) Carrying out the step 2 of secret reconstruction, we have the results given in Table 3.

3) Results of the step 3 of the secret reconstruction are given in Table 4.

4) Step 4 of the reconstruction gives the results given in Table 5. Also

$$\boldsymbol{(14)_{10} = (01110)_2 \implies temp = [0, 1, 1, 1, 0]}$$

$$\boldsymbol{exor(temp, st') = exor([0, 1, 1, 1, 0], [1, 1, 0, 1, 1]) = [1, 0, 1, 0, 1] = st}$$

5) Converting the binary tuple st to decimal, we have $s = 21$.

6 Secret Changeability with an Example

6.1 Secret Changeability

In our scheme, the shares are reusable, That is, the shares of the participants can remain same even for a new secret. The following algorithm is to renew the secret by the dealer.

6.1.1 Secret Renewal

Dealer performs the following steps:

1) Selects a new secret $ns \in \mathbb{Z}_{2^{\ell-1}} - \{0\}$;

2) Converts the secret ns into $\ell - 1$ bit binary number $ns_{\ell-1}ns_{\ell-2}ns_{\ell-3}\cdots ns_3ns_2ns_1$ and forms new secret tuple $nst = [ns_{\ell-1}, ns_{\ell-2}, ns_{\ell-3}\cdots ns_3, ns_2, ns_1]$;

3) Calculates $nst' = exor(nst, temp)$, let $nst' = [ns'_{\ell-1}, ns'_{\ell-2}, ns'_{\ell-3}\cdots ns'_3, ns'_2, ns'_1]$;

4) Calculates $nu = bagsum(nst', w)$ and makes nu public.

Table 6 shows the parameters that change after the secret renewal algorithm.

There is a change in the secret, secret tuple, $exor$ tuple and public value only, all other parameters are intact. Since w contains the compartment secret shares for levels from 1 to $\ell - 1$ and decimal equivalent of $temp$ is the compartment secret share for level ℓ, the compartment shares doesn't change after changing the secret by the dealer. Since there is no change in compartment shares, there is no need for distribution phase after the secret update. There is no change in the secret reconstruction phase. Based on the new public value nu, the compartment can get the actual secret using their corresponding secret shares. We explain the algorithm in detail with an example in next subsection. Pinnacle step for running time of the above algorithm is 4. So the running time of the secret renewal algorithm is $\mathcal{O}(\ell)$.

6.2 Example for Secret Changeability

We use the example explained in section 5 to explain share renewal algorithm. Now, total number of levels (ℓ)=6. Threshold values and total number of participants for

Table 2: Shamir's secret reconstruction at compartments

Level number	$(\mathbf{t_i}, \mathbf{n_i})$	**Interested participants with shares**	**Compartmental share (lagrange interpolation)**
1	(2,5)	$(1, 131), (3, 133)$	$\frac{3}{2}(131) + \frac{-1}{2}(133) = \mathbf{130}$
2	(3,5)	(1,65), (3,93), (5,145)	$\frac{15}{8}(65) + \frac{-5}{4}(93) + \frac{3}{8}(145) = \mathbf{60}$
3	(4,5)	(1,35), (3,69), (4,110), (5,175)	$\frac{5}{2}(35) + (-5)(69) + 5(110) + \frac{-3}{2}(175) = \mathbf{30}$
4	(5,6)	(1,20), (2,95), (3,368), (4,502), (5,232)	$5(20) + (-10)(95) + 10(368) + (-5)(502) + 1(232) = 552 \equiv \mathbf{11} \mod 541$
5	(3,6)	(1,11), (3,19), (6,46)	$\frac{9}{5}(11) + (-1)(19) + \frac{1}{5}(46) = \mathbf{10}$
6	(5,7)	(1,33), (2,284), (3,267), (4,407), (5,431)	$5(33) + (-10)(284) + 10(267) + (-5)(407) + 1(431) = -1609 \equiv \mathbf{14} \mod 541$

Table 3: Output bit from Level 1

Level number	**u**	$\mathbf{w_1}$	$\mathbf{u \geq w_1}$	**Output bit**	$\mathbf{u'_{1,2}}$	$\mathbf{st''}$
1	211	130	True	1	211-130=81	[1]

Table 4: Output bits from Level 2 to Level 5

Level number	$\mathbf{u'_{i-1,i}}$	$\mathbf{w_i}$	$\mathbf{u'_{i-1,i} \geq w_i}$	**Output bit**	$\mathbf{u'_{i,i+1}}$	$\mathbf{st''}$
2	81	60	True	1	81-60=21	[1,1]
3	21	30	False	0	21	[0,1,1]
4	21	11	True	1	21-11=10	[1,0,1,1]
5	10	10	True	1	10-10=0	[1,1,0,1,1]

Table 5: Shamir's secret reconstruction at last compartment

Level number	$(\mathbf{t_i}, \mathbf{n_i})$	**Interested participants with shares**	**Compartmental share (Lagrange interpolation)**
6	(5,7)	(1,33), (2,284), (3,267), (4,407), (5,431)	$5(33) + (-10)(284) + 10(267) + (-5)(407) + 1(431) = -1609 \equiv \mathbf{14} \mod 541$

Table 6: Table showing change in values

Parameter	**Previous value**	**New value**	**Change in parameter**
Secret	s	ns	Yes
Secret tuple	st	nst	Yes
exor tuple	st'	nst'	Yes
Random bit tuple	$temp$	$temp$	No
Super increasing tuple	w	w	No
Public value(u)	u	nu	Yes
prime number	q	q	No

each level are as follows: $(t_1, n_1) = (2, 5)$, $(t_2, n_2) = (3, 5)$, $(t_3, n_3) = (4, 5)$, $(t_4, n_4) = (5, 6)$, $(t_5, n_5) = (3, 6)$, $(t_6, n_6) = (5, 7)$.

Now dealer executes the following algorithm to change secret value from $s = 21$ to $ns = 25$.

6.2.1 Secret Renewal

Dealer performs the following steps:

1) Selects a new secret $ns \in \mathbb{Z}_{2^{\ell-1}} - \{0\}$;

$$\boldsymbol{ns = 25, \text{ Since } \ell = 6,}$$
$$\boldsymbol{\mathbb{Z}_{2^{\ell-1}} - \{0\} = \{1, 2, 3 \cdots 31\}.}$$

2) Converts the secret ns into $\ell - 1$ bit binary number $ns_{\ell-1}ns_{\ell-2}ns_{\ell-3} \cdots ns_3ns_2ns_1$ and forms new secret tuple $nst = [ns_{\ell-1}, ns_{\ell-2}, ns_{\ell-3} \cdots ns_3, ns_2, ns_1]$;

$$\boldsymbol{nst = [1, 1, 0, 0, 1]}$$

3) Calculates $nst' = exor(nst, temp)$, let $nst' = [ns'_{\ell-1}, ns'_{\ell-2}, ns'_{\ell-3} \cdots ns'_3, ns'_2, ns'_1]$;

$$\boldsymbol{nst' = exor(nst, temp) =}$$
$$\boldsymbol{exor([1, 1, 0, 0, 1], [0, 1, 1, 1, 0]) = [1, 0, 1, 1, 1]}$$

4) Calculates $nu = bagsum(nst', w)$ and makes nu public;

$$\boldsymbol{nu = bagsum(nst', w) =}$$
$$\boldsymbol{bagsum([1, 0, 1, 1, 1], [10, 11, 30, 60, 130]) =}$$
$$\boldsymbol{10 + 30 + 60 + 130 = 230}$$

Now, we explain secret reconstruction algorithm with the new values. Instead of repeating compartment share reconstruction, we assume that all applied Lagrange interpolation and obtained their shares as shown in share reconstruction of Section 5.

6.2.2 Secret Reconstruction

1) At least t_i participants of level i performs (t_i, n_i) Shamir's secret reconstruction and gets the compartment/level share w_i for $1 \le i \le \ell - 1$;

Table 7: Compartment secrets

Level number	Compartmental share (Lagrange interpolation)
1	**130**
2	**60**
3	**30**
4	$552 \equiv \mathbf{11} \mod 541$
5	**10**
6	$-1609 \equiv \mathbf{14} \mod 541$

2) Carrying out the step 2 of secret reconstruction, we have the results given in Table 8;

3) Step 3 of the reconstruction gives the results given in Table 9;

4) Level ℓ performs (t_ℓ, n_ℓ) Shamir's secret reconstruction and converts the result in to $\ell - 1$ bit tuple, which is $temp = [e_{\ell-1}, e_{\ell-2}, e_{\ell-3} \cdots e_3, e_2, e_1]$. (Observe that $st' = st''$ and st'' is public). level ℓ performs $exor(temp, st')$ which results in st;

$$\boldsymbol{(14)_{10} = (01110)_2 \implies temp = [0, 1, 1, 1, 0]}$$
$$\boldsymbol{exor(temp, nst') =}$$
$$\boldsymbol{exor([0, 1, 1, 1, 0], [1, 0, 1, 1, 1]) =}$$
$$\boldsymbol{[1, 1, 0, 0, 1] = nst}$$

5) Finally the secret binary tuple nst is obtained, which can be converted to decimal to obtain secret s.

$$\boldsymbol{nst = [1, 1, 0, 0, 1] \implies ns = (11001)_2 = 25.}$$

Thus the new secret can be reconstructed by the compartments using the same shares. Note that for the sake of completeness, we repeated Steps 1 to 4 in the above algorithm, which is not needed since the compartments already have calculated their shares earlier.

7 Brief Note on Security and Observations

7.1 Brief Note on Security

- *Public*: u, q, ℓ;
- *Private to dealer*: $s, st, temp, st', w$, polynomials (use to generate shares);
- *Private to compartment i participant*: Participant share of the compartment secret w_i.

Note that with public values u, q, an intruder (either inside or outside $\mathfrak{P}$) cannot get about st and s because s, st are private to dealer only and w is needed to calculate s along with u. All levels has to intervene to get the secret, because $w, temp$ are available only after the secret share reconstruction of each and every compartment. In our scheme, the compartment with lower level number has highest priority because, other levels cannot get the bit information without the value passed by the higher level.

Assuming that the dealer is honest, it is infeasible to obtain s from u, q, ℓ. Hence our scheme is secure with respect to trusty dealer.

7.2 Observations

The multipartite secret sharing scheme proposed deals with the $\ell - 1$ bit secret, i.e., $1 \le s \le 2^{\ell-1} - 1$. So, in step 1 of share distribution phase, dealer selects secret $s \in \mathbb{Z}_{2^{\ell-1}} - \{0\}$. The selection range of the secret can be increased. Suppose the secret is of $\ell'(> \ell - 1)$ bits, then the following two trivial methods can be employed to handle this issue:

Table 8: Output bit from Level 1

Level number	**nu**	$\mathbf{w_1}$	$\mathbf{nu \geq w_1}$	**Output bit**	$\mathbf{nu'_{1,2}}$	$\mathbf{nst''}$
1	230	130	True	1	230-130=100	[1]

Table 9: Output bits from Level 2 to Level 5

Level number	$\mathbf{nu'_{i-1,i}}$	$\mathbf{w_i}$	$\mathbf{nu'_{i-1,i} \geq w_i}$	**Output bit**	$\mathbf{nu'_{i,i+1}}$	$\mathbf{nst''}$
2	100	60	True	1	100-60=40	[1,1]
3	40	30	True	1	40-30=10	[1,1,1]
4	10	11	False	0	10	[0,1,1,1]
5	10	10	True	1	10-10=0	[1,0,1,1,1]

Table 10: Shamir's secret reconstruction at last compartment

Level number	**Compartmental share (Lagrange interpolation)**
6	$-1609 \equiv \mathbf{14} \mod 541$

Method 1: one of the levels can be used to handle the extra bits. The extra bits can be shared to a compartment as compartment share (may be after *exor* with *temp* tuple) and after reconstruction phase, these bits get pre-appended to the tuple st''.

Method 2: Each level may be given multiple compartment shares. Thus, each level can reconstruct multiple compartment secrets and hence multiple bits related to secret.

The pinnacle step for running time is share distribution by dealer to all the participants using polynomial, which is $\mathcal{O}(\sum_{i=1}^{\ell}(n_i t_i))$. Since compartments can concurrently get their shares during secret reconstruction phase, the running time is $\mathcal{O}((max(n_i))^2 + \ell)$. Note that $\ell - 1$ comparisons and one *exor* is needed in order to obtain the secret.

8 Conclusions

In this paper, we proposed our new multipartite secret sharing scheme, which is based on superincreasing sequence. The property of secret changeability is explained along with an example, listed various observations and discussed briefly about the security of the scheme in the case of trustworthy dealer. Work is underway to extend these ideas to arrive at similar schemes for other access structures.

References

[1] F. Alsolami and T. E. Boult, "Cloudstash: Using secret-sharing scheme to secure data, not keys, in multi-clouds," in *11th International Conference on Information Technology: New Generations (ITNG'14)*, pp. 315–320, 2014.

[2] G. R. Blakley, "Safeguarding cryptographic keys," *Proceeding of the National Computer Conference1979*, vol. 48, pp. 313–317, 1979.

[3] E. F. Brickell, "Some ideal secret sharing schemes," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 468–475, 1989.

[4] E. Dawson and D. Donovan, "The breadth of shamir's secret-sharing scheme," *Computers & Security*, vol. 13, no. 1, pp. 69–78, 1994.

[5] N. A. Ebri, J. Baek and C. Y. Yeun, "Study on secret sharing schemes (SSS) and their applications," in *International Conference for Internet Technology and Secured Transactions (ICITST'11)*, pp. 40–45, 2011.

[6] O. Farràs, J. Martí-Farré, and C. Padró, "Ideal multipartite secret sharing schemes," *Journal of Cryptology*, vol. 25, no. 3, pp. 434–463, 2012.

[7] O. Farras, C. Padró, C. Xing, and A. Yang, "Natural generalizations of threshold secret sharing," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1652–1664, 2014.

[8] P. M. Fathimal and P. A. J. Rani, "Threshold secret sharing scheme for compartmented access structures," *International Journal of Information Security and Privacy*, vol. 10, no. 3, pp. 1–9, 2016.

[9] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini, "Secret sharing in multilevel and compartmented groups," in *Australasian Conference on Information Security and Privacy*, pp. 367–378, 1998.

[10] L. Harn and M. Fuyou, "Multilevel threshold secret sharing based on the chinese remainder theorem," *Information Processing Letters*, vol. 114, no. 9, pp. 504–509, 2014.

[11] C. F. Hsu and L. Harn, "Multipartite secret sharing based on crt," *Wireless Personal Communications*, vol. 78, no. 1, pp. 271–282, 2014.

[12] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 72, no. 9, pp. 56–64, 1989.

[13] S. C. Kothari, "Generalized linear threshold scheme," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 231–241, 1984.

[14] P. S. Kumar, R. R. Kurra, A. N. Tentu, and G. Padmavathi, "Multi-level secret sharing scheme for mobile ad-hoc networks," *International Journal of Advanced Networking and Applications*, vol. 6, no. 2, pp. 2253, 2014.

[15] M. Larson, C. Hu, R. Li, W. Li, and X. Cheng, "Secure auctions without an auctioneer via verifiable secret sharing," in *Proceedings of the Workshop on Privacy-Aware Mobile Computing*, pp. 1–6, 2015.

[16] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 525–530, 1978.

[17] D. K. Pattipati, A. N. Tentu, C. V. Vadlamudi and A. A. Rao, "Sequential secret sharing scheme based on level ordered access structure.," *International Journal of Network Security*, vol. 18, no. 5, pp. 874–881, 2016.

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

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

[20] A. N. Tentu, P. Paul and C. V. Vadlamudi, "Conjunctive hierarchical secret sharing scheme based on mds codes," in *International Workshop on Combinatorial Algorithms*, pp. 463–467, 2013.

[21] Y. Wang and Y. Desmedt, "Efficient secret sharing schemes achieving optimal information rate," in *IEEE Information Theory Workshop (ITW'14)*, pp. 516–520, 2014.

Biography

Putla Harsha received M.Tech from University of Hyderabad, Hyderabad and she did Bachelors degree in computer science. Her research interests include Cryptography, Algorithms, and Computational number theory.

Patibandla Chanakya received M.Tech from University of Hyderabad, Hyderabad. Currently he is pursuing his PhD in Computer Science from University of Hyderabad. His research interests include Computational number theory, Algorithms, and Cryptography.

Vadlamudi China Venkaiah obtained his PhD in 1988 from the Indian Institute of Science (IISc), Bangalore in the area of scientific computing. He worked for several organisations including the Central Research Laboratory of Bharat Electronics, Tata Elxsi India Pvt. Ltd., Motorola India Electronics Limited, all in Bangalore. He then moved onto academics and served IIT, Delhi, IIIT, Hyderabad, and C R Rao Advanced Institute of Mathematics, Statistics, and Computer Science. He is currently serving the Hyderabad Central University. He is a avid researcher. He designed algorithms for linear programming, subspace rotation and direction of arrival estimation, graph colouring, matrix symmetriser, integer factorisation, cryptography, knapsack problem, *etc.*