Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 26 (2023), No. 1, pp. 191–205
DOI : 10.1080/09720529.2021.1925447

Secret sharing scheme based on Latin squares

Pithani Ashwini [†]
C. R. Rao Advanced Institute of Mathematics,
Statistics and Computer Science (AIMSCS)
University of Hyderabad Campus
Gachibowli 500046
Telangana
India

V. Ch. Venkaiah [*]
Wilson Naik Bhukya [§]
School of Computer and Information Sciences
University of Hyderabad
Prof. C. R. Rao Road
Gachibowli
Hyderabad 500046
Telangana
India

Abstract

Secret sharing is a method of distributing a secret among a group of participants such that if (i) an authorized set of participants cooperate, then they can reconstruct the secret and (ii) an unauthorized set of participants can't recover the secret and doesn't know anything about the secret. In 1994, Joan Cooper proposed a secret sharing scheme based on the critical set of Latin square[3]. This has some drawbacks. In 2015, Rebecca J. Stones published a secret sharing scheme based on autotopism of Latin square[1]. The scheme solves all the issues associated with the Cooper's scheme. However, we noticed that, her scheme has the following shortcomings: (i) The scheme construction is limited to a particular pattern of the Latin square only, (ii) The scheme works only as k-out-of-k technique and not as a proper threshold scheme, and (iii) the dealer is the central part of the whole process, if he is dishonest then the whole secret is revealed. So, in this paper, we modify Rebecca's scheme so that it works (i) for a couple of more contours with only one autotopism, and (ii) as a t-groups-out-of-k scheme. So to reconstruct the secret anyone of the available t groups

[†]*E-mail:* pithaniashwini@gmail.com
[*]*E-mail:* venkaiah@hotmail.com (Corresponding Author)
[§]*E-mail:* rathore@uohyd.ac.in

can cooperate and recover it. That is, the modified scheme could reduce the key availability problem to some extent. As a byproduct of our modification the security from the brute force attack increases as well.

Subject Classification: *(2010) 94A62, 05B15.*

Keywords: *Latin square, Autotopism, Critical set, Contour, Secret sharing scheme.*

1. Introduction

In many applications, an information system requires more than one participant to access and integrate the information. Based on access rights, users can access or modify the data. Access rights can be provided by employing a key, a password or a token to the secure key management system but the problem is the distribution of the key among the participants. If the key is given to one of the participants in the system, it provides security to the system. But the problem is, the person who has the key may not be available all the time, and he may die or forget the key. This is known as key availability problem. To overcome the key availability problem, the same key may be distributed among all the participants so as to provide reliability to the system. But it leads to integrity problems. Secret sharing scheme is one of the methods that addresses these problems. Several secret sharing schemes are proposed in the literature [1] [3] [5] [6] [11] [12] [13]. One such was proposed by Joan Cooper [3]. It is based on the critical set of a Latin square. It has some drawbacks such as (i) it leaks partial information of the secret through the share of a participant, (ii) the size of the secret space reduces gradually with the increase of number of cooperating participants, (iii) computation of a critical set of a Latin square is difficult, and (iv) the extension of a critical set to its corresponding Latin square is difficult. So, Rebecca published a secret sharing scheme based on the autotopism and the contour of a Latin square. This scheme overcomes all the issues associated with the Cooper's scheme. However, we noticed that, her scheme has the following shortcomings: (i) The scheme construction is limited to a particular pattern of the Latin square only, (ii) The scheme works only as k-out-of-k technique and not as a proper threshold scheme, and (iii) the dealer is the central part of the whole process; if he is dishonest then the whole secret is revealed. Since the scheme is limited to a very few set of contours, it may not be widely applicable in practice. Also, since it is a k-out-of-k scheme, all the participants are required to reconstruct the secret; thereby having a room for the key availability problem. So, in this paper, we modify the Rebecca's

scheme so that it works (i) for a couple of more contours with only one autotopism, and (ii) as a t-groups-out-of-k scheme. In the t-groups-out-of-k scheme, the key is partitioned in t different ways and each partition contains shares such that the product of all its shares gives the key. So to reconstruct the secret anyone of the available t groups can cooperate and recover it. That is, the modified scheme could reduce the key availability problem to some extent. Also the modified scheme increases the flexibility as well as security of the previous scheme.

2. Preliminaries

This section briefly explains the key concepts required in designing the proposed scheme.

2.1 *Latin square*

A Latin square of order n is a square matrix of order n and each entry of a matrix is chosen from a set of n symbols such that each entry of the matrix occurs exactly once in each row and column. For example latin square of order 3 with symbols {a,b,c} is

$$M = \begin{bmatrix} a & b & c \\ c & a & b \\ b & c & a \end{bmatrix}$$

Number of possible latin squares of order n is (n!).(n-1)!.r, where r is the number of possible reduced latin squares. Reduced Latin Square is a latin square with first row and first column have same order of symbols.

2.2 *Autotopism*

Let θ be a triple order tuple of $S_n \times S_n \times S_n$, where S_n is a symmetric group acting on a set of n positive numbers, Z_n. That is, θ is in the form of $\theta(\alpha, \beta, \gamma)$, here α is a permutation of rows, β is a permutation of columns, and γ is a permutation of symbols. If the result of the application of θ on the partial Latin square C of Latin square L can be extended to a Latin square L' then such a C is called as a contour. If L and L' are equal then θ is called as autotopism otherwise θ is a isotopism.

$$\theta(i, j, l_{i,j}) = (\alpha(i), \beta(j), \gamma(l_{i,j})) \ \forall i, j \in Z_n$$

Table 1

Minimum number of elements in contour with order n

Order	Maximum Cycle length	Minimum number of elements
3	3	3
3	2	4
4	3	6
4	2	8
6	6	6
6	5	8
8	4	16
8	2	32

A critical set of a latin square is a partial latin square that can be uniquely completable. A Contour is different from a critical set. The minimum number of elements in a contour depends on the cyclic structure of the autotopism θ. When cyclic structures of α, β and γ are the same and the maximum cycle length is s, then the minimum number of elements in a contour is $(n^2/s)+(n^2\%s)$, where n is the order of the contour and x%s denotes the remainder when x is divided by s. This can be seen from the Table 1.

3. Proposed Scheme

In this section we propose expressions for couple of more contours by coming up with new pattern and design a scheme based on one autotopism. The proposed scheme is extended to reduce the key availability problem by making it to work as a t-groups-out-of-k secret sharing scheme. In the t-groups-out-of-k scheme, the autotopism θ is split in t different ways and distribute the t partitions among the groups of participants in the system. During reconstruction, any group out of t groups can reconstruct the secret. As usual the proposed scheme consists of a Distribution phase and a Reconstruction phase.

3.1 *Distribution Phase*

Overview of the distribution phase is as follows:

Generate random contour C from the set of equations given in Step 1 of the distribution algorithm and come up with a corresponding random autotopism θ such that its application on C results in a Latin square L,

where L is the secret to be shared, among the participants. Split θ into ℓ pieces as $\lambda_1, \lambda_2, ..., \lambda_\ell$ in t different ways such that the product $\lambda_1 . \lambda_2 ... \lambda_\ell = \theta$ Similarly, partition the k participants into t groups of size ℓ, where ℓ can be different for each group. The dealer then computes C_{public} by applying $\xi = \lambda_\ell . \lambda_{\ell-1} .... \lambda_1$ to the contour C and makes C_{public} public. It is useful at the time of reconstruction. This scheme is applicable for Latin squares of order $n \equiv 2(\text{mod } 4)$.

Detailed description of the distribution phase is given in the following steps:

1. Randomly generates a contour $C_{initial}$ of order n from defined pattern as below.

   $$M_{i,i} = M_{i+(n/2),i+(n/2)} = Z_i$$

   $$M_{(n/2)+i,i} = M_{i,(n/2)+i} = n/2 - Z_i$$

   $$Z_i \in \{0,1,2,3,...n/2\} \qquad i \in \{0,1,2,....n/2 - 1\}$$

   The autotopism $\theta_{initial}$ = $(\tau, \tau^{-1}$, (even elements)(odd elements)), where τ is (1,2,...n/2 -1)(n/2,....n). We compute the initial secret $L_{initial}$ by admitting $\theta_{initial}$ on $C_{initial}$.

2. Extending a partial Latin square with some regular pattern to the corresponding full pledged Latin square is easier than the contour with random elements. So we disturb the pattern of the contour $C_{initial}$ and convert it to a corresponding contour called intermediate contour C_{im}. C_{im} is computed by taking each entry of $C_{initial}$ and applying $\theta^t_{initial}(i, j, C_{i,j})$ transition, where t is randomly chosen from $\{0,1,2,....n/2-1\}$. This operation must ensure that applying $\theta_{initial}$ on C_{im} generates a latin square, $L_{initial}$.

Algorithm 1 computation of C_{im}

for every element 'e' in $C_{initial}$ **do**

select 't' from 0 to (n/2)-1

add resultant $\theta^t_{initial}(i, j, C_{i,j})$ to C_{im}

end for

---

3. Randomly generate an isotopism ϕ such that the operation of $\theta_{final} = \phi.\theta_{initial}.\phi^{-1}$ must be conjugate to $\theta_{initial}$ i.e cyclic structure of θ_{final} is same as $\theta_{initial}$. Compute the final secret L_{final} by applying ϕ on $L_{initial}$ and the final contour C_{final} by applying ϕ on C_{im}

$$C_{final} = \phi(C_{im})$$

$$L_{final} = \phi(L_{initial})$$

$$\theta_{final} = \phi.\theta_{initial}.\phi^{-1}$$

4. Now the dealer computes the shares by dividing θ_{final} into λ_1, λ_2, $\lambda_3 .... \lambda_l$ in t different ways such that $\lambda_1 . \lambda_2 . \lambda_3 .... \lambda_l = \theta_{final}$. He makes the participants into t different groups of size ℓ, where ℓ can be different for each group. Distributes each type of division of θ_{final} among the groups $G_1, G_2, G_3 ..... G_t$.
5. Finally the dealer computes the public value C_{public} by applying ξ permutation on C_{final}, where ξ is $\lambda_\ell .... \lambda_2 . \lambda_1$. Each group of participants compute the ξ and C_{public}. So, we will get t different types of ξ values and C_{public} values. Make all C_{public} values as public. If we directly make C_{final} as public value then, it reveals the partial information of the secret.

3.2 *Reconstruction Phase*

Any available group out of t groups can reconstruct the secret as follows:

1. Compute θ_{comb} by taking the product of all the shares of the participants of the available group, that is $\theta_{comb} = \lambda_1 . \lambda_2 ... \lambda_\ell$. Then the cyclic structure of θ_{comb} is compared with the original θ cyclic structure n/2. If it is different then the combiner returns false otherwise, returns true and moves to further steps.

$$\theta_{comb} = \lambda_1 . \lambda_2 \cdots \lambda_\ell$$

2. Computes the contour C_{comb} by applying ξ^{-1} to the C_{public} corresponding to the group that is reconstructing the secret. That is compute $C_{comb} = \xi^{-1}(C_{public}) = \lambda_1^{-1} . \lambda_2^{-1} ..... \lambda_\ell^{-1}(C_{public})$.

3. Apply θ_{comb} on C_{comb} and recover the L_{comb}. If L_{comb} is a latin square then combiner returns θ_{comb} to participants otherwise returns false.

4. Example

4.1 *Distribution Phase*

1. Randomly choose

$$C_{initial} = \begin{bmatrix} 1 & . & . & 2 & . & . \\ . & 2 & . & . & 1 & . \\ . & . & 1 & . & . & 2 \\ 2 & . & . & 1 & . & . \\ . & 1 & . & . & 2 & . \\ . & . & 2 & . & . & 1 \end{bmatrix}$$

and $\theta_{initial}$ = (0 1 2)(3 4 5), (0 2 1)(3 5 4), (0 2 4)(1 3 5).

$$L_{initial} = \begin{bmatrix} 1 & 3 & 0 & 2 & 4 & 5 \\ 5 & 2 & 3 & 0 & 1 & 4 \\ 4 & 5 & 1 & 3 & 0 & 2 \\ 2 & 4 & 5 & 1 & 3 & 0 \\ 0 & 1 & 4 & 5 & 2 & 3 \\ 3 & 0 & 2 & 4 & 5 & 1 \end{bmatrix}$$

2. Randomly choose t from {0,1,2} and compute $C_{im} = C_{im(i,j)} = \theta^t_{initial}(i, j, C_{i,j})$ as follows:

$\theta^0_{initial}(0,0,1) - - > (0,0,1)$ $\theta^1_{initial}(0,3,2) - - > (1,5,4)$

$\theta^0_{initial}(1,1,2) - - > (1,1,2)$ $\theta^1_{initial}(1,4,1) - - > (2,3,3)$

$\theta^0_{initial}(2,2,1) - - > (2,2,1)$ $\theta^1_{initial}(2,5,2) - - > (0,4,4)$

$\theta^1_{initial}(3,0,2) - - > (4,2,4)$ $\theta^0_{initial}(3,3,1) - - > (3,3,1)$

$\theta^1_{initial}(4,1,1) - - > (5,0,3)$ $\theta^1_{initial}(4,4,2) - - > (5,3,4)$

$\theta^2_{initial}(5,2,2) - - > (4,0,0)$ $\theta^0_{initial}(5,5,1) - - > (5,5,1)$

$$C_{im} = \begin{bmatrix} 1 & . & . & . & 4 & . \\ . & 2 & . & . & . & 4 \\ . & . & 1 & 3 & . & . \\ . & . & . & 1 & . & . \\ 0 & . & 4 & . & . & . \\ 3 & . & . & 4 & . & 1 \end{bmatrix}$$

3. Let the randomly generated isotopism ϕ be {(3 4 2), (0 5 2 1), (0 3 4 2 5 1)}. Then

$\theta_{final} = \phi.\theta_{initial}.\phi^{-1}$

$\theta_{final} = \{(254)(130), (105)(243), (104)(523)\}$

$C_{final} = \phi(C_{im})$

$$C_{final} = \begin{bmatrix} . & . & . & . & 2 & 0 \\ 5 & . & 2 & . & . & . \\ . & 2 & . & . & . & 3 \\ . & 0 & . & 4 & . & . \\ . & . & . & 0 & . & . \\ . & . & 0 & 2 & . & 4 \end{bmatrix}$$

$L_{final} = \phi(L_{initial})$

$$L_{final} = \begin{bmatrix} 4 & 3 & 1 & 5 & 2 & 0 \\ 5 & 4 & 2 & 3 & 0 & 1 \\ 0 & 2 & 4 & 1 & 5 & 3 \\ 1 & 0 & 5 & 4 & 3 & 2 \\ 2 & 1 & 3 & 0 & 4 & 5 \\ 3 & 5 & 0 & 2 & 1 & 4 \end{bmatrix}$$

4. Assume that there are nine participants in the system and they are partitioned into three groups as G_1, G_2 and G_3 of size three. Split θ_{final} into pieces as λ_1, λ_2 and λ_3 such that $\lambda_1.\ \lambda_2.\ \lambda_3 = \theta_{final}$ in three different ways and distribute them among the participants.

Group G_1

$\lambda_1 = \{(2\ 4)(1\ 0),\ (1\ 2\ 0\ 5),\ (1\ 5\ 4\ 3)(2\ 0)\}$

$\lambda_2 =$ (4 5 0), (2 3)(1 5), (1 5 3)(4 2)}

$\lambda_3 =$ {(0 4 3), (3 0 4)(5 1),(3 0)}

Group G_2

$\lambda_1 =$ {(2 3 4 1)(0 5),(1 5)(3 4),(1 5 0 4)(2)}

$\lambda_2 =$ (2 3 1)(4 0),(5 0 4),(5 3)(1 4)}

$\lambda_3 =$ {(1 5),(4 5 2),(3 0 2)(4 1)}

Group G_3

$\lambda_1 =$ {(2 1 5 4),(1 2 5 3)(4 0),(1 4)(5 3)}

$\lambda_2 =$ (1 0)(2 4),(1 2 3)(5 4),(4 0 3)}

$\lambda_3 =$ {(0 5 3)(4 2),(3 0),(3 4 2)}

5. Compute C_{public} for each group and publish it.

Group G_1

$\xi = \lambda_3.\lambda_2.\lambda_1 =$ {(0 5 1)(4 3 2),(3 5 1 2)(0 4),(0 5 1 4)(2 3)}

$$C_{public} = \xi(C_{final}) = \begin{bmatrix} . & . & . & 3 & 1 & . \\ . & 0 & . & 5 & . & 3 \\ . & . & 5 & . & . & 0 \\ . & . & . & . & . & 5 \\ . & 2 & 3 & . & . & . \\ 3 & 5 & . & . & . & . \end{bmatrix}$$

Group G_2

$\xi = \lambda_3.\lambda_2.\lambda_1 =$ {(0 1)(2 4 5 3), (0 3 4)(1 5 2), (0 2)(1 5 3 4)}

$$C_{public} = \xi(C_{final}) = \begin{bmatrix} . & 0 & . & 3 & . & . \\ 0 & . & 2 & . & . & . \\ . & . & . & . & 1 & 2 \\ . & 2 & 1 & . & 0 & . \\ . & . & 4 & . & . & 0 \\ . & . & . & . & 2 & . \end{bmatrix}$$

Group G_3

$\xi = \lambda_3.\lambda_2.\lambda_1 = \{(0\ 4\ 2\ 1)(3\ 5),\ (0\ 2\ 1\ 5)(3\ 4),\ (0\ 5\ 3)(1\ 4\ 2)\}$

$$C_{public} = \xi(C_{final}) = \begin{bmatrix} . & 1 & 3 & . & . & . \\ 0 & . & . & . & . & 1 \\ . & . & . & . & 5 & . \\ 2 & 5 & . & . & 1 & . \\ 5 & . & . & 1 & . & . \\ . & . & . & . & 2 & 5 \end{bmatrix}$$

4.2 *Reconstruction Phase*

Assume that all the participants of G_2 are available at the time of secret reconstruction. Then all G_2 participants share their share to the combiner as $\lambda_1 = \{(2\ 3\ 4\ 1)(0\ 5), (1\ 5)(3\ 4), (1\ 5\ 0\ 4)(2\ 0)\}$, $\lambda_2 = (2\ 3\ 1)(4\ 0), (5\ 0\ 4), (5\ 3)(1\ 4)$ and $\lambda_3 = \{(1\ 5), (4\ 5\ 2), (3\ 0\ 2)(4\ 1)\}$. Suppose, all participants of the G_2 honestly shared their share to the combiner then the resultant $\theta_{comb} = \{(2\ 5\ 4)(1\ 3\ 0), (1\ 0\ 5)(2\ 4\ 3), (1\ 0\ 4)(5\ 2\ 3)\}$ and ξ^{-1} of G_2 is $\{(0\ 1)(3\ 5\ 4\ 2), (4\ 3\ 0)(2\ 5\ 1), (0\ 2)(4\ 3\ 5\ 1)\}$.

$$C_{comb} = \xi^{-1}(tC_{public}) = \begin{bmatrix} . & . & . & . & 2 & 0 \\ 5 & . & 2 & . & . & . \\ . & 2 & . & . & . & 3 \\ . & 0 & . & 4 & . & . \\ . & . & . & 0 & . & . \\ . & . & 0 & 2 & . & 4 \end{bmatrix}$$

Apply θ_{comb} on C_{comb} and recover the L_{final}.

5. Security issues

5.1 *Brute force attack*

Brute force attack, to determine the secret, requires to compute all possible latin squares associated with each isotopism θ. This is because the adversary doesn't have any knowledge of θ. That is the adversary is required to explore all possible θ values and all possible latin squares corresponding to each θ. The amount of work required to do so, for even small orders, can be visualized from the numbers given in the Table 2 [7]. Entries in the third column of Table 2, that is, the number of isotopisms can be determined as follows: θ is an element of $S_n \times S_n \times S_n$ and the cyclic structure of S_n is n/2. So the number of possible permutations is $(n!/(2!(n/2)^2))^3 = (2(n-1)!/n)^3$. That is, the number of permutations grows exponentially. It can be seen from the table that not only the permutations but also the number of latin squares associated with each permutation grows quite rapidly. Therefore determining the secret using brute force requires exponential time complexity and hence not feasible. For example, even for order six, the number of latin squares to be computed by brute force attack will be more than $648 \times (6 \times 10^4) = 3888 \times 10^4$.

5.2 *Completion of* C_{public}

The proposed scheme is immune to leakage of the partial information of the secret. However, C_{public} can give some information about the secret. Completion of the C_{public} can give the isotopism class of Latin square L and L is one of the values in the class. The probability of guessing original L from the isotopism class is 1/(size of isotopism class). We can find the size of the isotopism class of the Latin square by using the "Orbit-Stabilizer" theorem. According to "Orbit-Stabilizer" theorem, size of isotopism class, size(isot) = $n!^3 / autp(L)$, where autp(L) is the size of the autotopism

Table 2

Number of possible Latin squares with θ and possible isotopisms with order n

n	No. of Latin squares with θ	No. of possible isotopisms
6	648	6×10^4
10	20,820,000	3×10^{14}
14	?	7×10^{26}
18	?	6×10^{40}

Table 3
Minimum size of the isotopism class

n	Minimum size of size(isot)
6	2×10^5
10	4×10^{14}
14	1×10^{27}
18	7×10^{39}

group of L. The maximum size of autp(L) of the Latin square with order n is $n^{2+\lfloor log_2^n \rfloor}$ [8], which gives the minimum size of isotopism class size. Size of isotopism class increases with the order. This can be clearly seen in the Table 3. So, probability of finding original L from the isotopism class decreases. Hence, finding original L from the completion of the C_{public} fails as the order increases.

5.3 *Participant Attack*

Attack can be from the participants of the scheme as well. During reconstruction, θ value is computed by taking the product of all shares of any one of the groups. Each participant has his share with him and don't have any knowledge about others share. If either the first or last participant in the group is dishonest then the dishonest participant can mislead the group in such a way that the group is denied the secret; whereas the dishonest member gets the secret. For example, if the first participant in the group is dishonest and he shares an erroneous share, λ_{1e} to the combiner, then the Combiner computes θ value as θ_e with the erroneous share of the first participant. First participant can then compute the correct θ value from the operation $\lambda_1 . \lambda_{1e}^{-1} . \theta_e$. Similarly, the last participant can mislead the group by sharing an erroneous share, say, $\lambda_{\ell e}$. So the combiner computes the θ value as θ_e with the last participant's erroneous share. Whereas the last participant can compute the correct θ by performing the operation $\theta_e . \lambda_{\ell e}^{-1} . \lambda_\ell$. This attack will not be successful because the θ_{comb} may not satisfy the structural requirements if it is not the correct one. In fact, the probability of θ_e satisfying the structural requirements $= \left(\frac{2(n-1)/n}{n!/2}\right)^3 = \frac{64}{n^6}$. Also the generated L_{comb} may not be a Latin square if θ_{comb} is not the correct one. That is, even if the dishonest participant succeeds with the probability , the probability of generating a Latin square with random θ is very less. This probability is computed in [1] for orders 6 and 10. This is in the range 3.13×10^{-5} to 4.5×10^{-5} for order 6 and $1.04 \times$

Table 4

Number of Possible contours with the pattern

order	6	10	14	**18**	22	26	32	38	44
Previous scheme	4	4	4	**4**	4	4	4	4	4
modified scheme	8	12	16	**20**	24	28	32	36	46

10^{-14} to 2×10^{-11} for order 10. Therefore the attack fails for sufficiently large orders of Latin squares.

6. Comparison with previous work

This section discusses the comparative analysis of the proposed scheme with that of the Rebecca's scheme. As claimed previously, the proposed scheme improves on that of the Rebecca's scheme in three aspects: (a) increase in the number of contours, (b) reduction in the key availability problem to some extent, and (c) increase in the number of permutations.

a. *Increase in the number of contours*

For constructing the scheme, we need a contour and an autotopism such that both together can generate a Latin square. By taking some random values of contour and autotopism, the contour can't be extended to a Latin square. We need some set of predefined contours along with autotopism so that the scheme can be easily constructed. In the Rebecca's scheme, a pattern for the contour to be used in the scheme was given. This pattern can generate four different contours for any order of Latin square. In our scheme, we modified the expressions so that they generate more number of contours. This can be seen from the Table 4. So there is an increase in the flexibility of the scheme while the security remains same.

b. *Key availability*

Secret sharing scheme provides key availability and security. Rebecca's scheme is successful in providing security to the secret but the key availability problem remains. Probability of availability of the key, when the key is with any one of the n participants is 1/n. Note that, this probability remains same with Rebecca's scheme; whereas in our scheme it is t/n, where t is the number of groups to which the participants are

partitioned. That is, our scheme reduces the key availability problem to some extent.

c. *Increase in the number of permutations*

In the Rebecca's scheme, all α, β and γ of the autotopism are same. Then the number of possible permutations with the previous θ is n!/(2! $(n/2)^2$) = 2(n-1)!/n. In our scheme, autotopism can have different α, β and γ values. So the number of possible permutations in our scheme is $(2(n-1)!/n)^3$ and hence more protection from the brute force attack.

References

[1] Stones R.J., Su, M., Liu, X. et al. A Latin square autotopism secret sharing scheme. Des. Codes Cryptogr. 80, 635–650 (2016). https://doi.org/10.1007/s10623-015-0123-1

[2] Chum, Chi Zhang, Xiaowen. (2009). Improved Latin Square based Secret Sharing Scheme. Computing Research Repository - CORR. 10.1090/conm/582/11562.

[3] Cooper, J. Donovan, Diane Seberry, Jennifer. (1994). Secret sharing schemes arising from Latin squares. Bulletin of the Institute of Combinatorics and its Applications. 12.

[4] Howse . Adell. (1998).Minimal critical sets for some small Latin squares. T*he Australasian Journal of Combinatorics* [electronic only. 17.

[5] Shamir, Adi. (1979) . How to Share a Secret. Commun. ACM . 10.1145/359168.359176 . 612-613 . 22

[6] G. R. BLAKLEY, Safeguarding cryptographic keys , 1979 . International Workshop on Managing Requirements Knowledge (MARK), New York, NY, USA, 1979, pp. 313-318, doi: 10.1109/MARK.1979.8817296.

[7] Stones, Douglas. (2010). The parity of the number of quasigroups. *Discrete Mathematics.* 310. 3033-3039. 10.1016/j.disc.2010.06.027.

[8] Browning, Josh Stones, Douglas Wanless, Ian. (2013). Bounds on the number of autotopisms and subsquares of a Latin square. Combinatorica. 33. 10.1007/s00493-013-2809-1.

[9] W. Ford and B. S. Kaliski, "Server-assisted generation of a strong secret from a password," Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises

(WET ICE 2000), Gaithersburg, MD, USA, 2000, pp. 176-180, doi: 10.1109/ENABL.2000.883724.

[10] Pal, S.K. Kapoor, Shivam Arora, Alka Chaudhary, Reshu Khurana, Jatin. (2010). Design of strong cryptographic schemes based on Latin Squares. *Journal of Discrete Mathematical Sciences Cryptography*. 3. 10.1 080/09720529.2010.10698290.

[11] Abhishek Mishra Ashutosh Gupta (2018) Multi secret sharing scheme using iterative method, *Journal of Information and Optimization Sciences,* 39:3, 631-641, DOI: 10.1080/02522667.2017.1385161.

[12] Priyanka Jaiswal Sachin Tripathi (2018) Cryptanalysis of olimid's group key transfer protocol based on secret sharing, *Journal of Information and Optimization Sciences,* 39:5, 1129-1137, DOI: 10.1080/02522667.2017.1292655.

[13] Carolina Mejia J. Andrés Montoya (2018) On the information rates of homomorphic secret sharing schemes, *Journal of Information and Optimization Sciences*, 39:7, 1463-1482, DOI: 10.1080/02522667.2017.1367513

Received October, 2020

Revised February, 2021