New Magic Square Problems and Their Applications in Cryptology

Tao Xie

The Center for Soft-Computing and Cryptology, NUDT, Changsha, China

(hamishxie@vip.sina.com)

Abstract: Magic square is a complex and hard permutation problem of combinatorial mathematics with a long history, previous research work primarily focuses on how to construct some magic squares with special mathematic properties, and theoretical investigations on their existence, general construction methods as well as count problem. Not until this paper appears, did there ever exist any technically practical applications of magic squares. A lot of majestic mathematic properties are to be found for magic squares. This paper gives a list of new properties in magic squares we have recently discovered, such as the unfinished magic squares, the modular sum separation of magic squares, the magic square shuffling, and the magic square encryption as well. All problems listed in this paper are new hard problems in combinatorial mathematics, they are both practically and theoretically significant. Probably, these new discoveries in magic square will greatly change the direction of research on magic squares.

Key Word: magic square, unfinished magic square, modular arithmetic, shuffling, permutation, authentication.

1 Introduction

Magic squares have been a popular topic in recreational mathematics for generations. Typically, a magic square is an arrangement of numbers in a square matrix so that the numbers in every column, row and diagonal of the matrix add up to the same value. The standard or normal magic square is defined as an arrangement of the first n^2 consecutive natural numbers into a square matrix so that the sum of the numbers in each column, row and diagonal is the same. This magic constant is determined by n and equal to $n(n^2+1)/2$. The number n, i.e. the number of cells on each side of a magic square, is referred to as the order of the magic square.

Magic squares have a rich history derived from a chinese legend He-Tu and Lo-Shu around 2200 B.C. The oldest magic square is the Lo-Shu, which is purported to be carried by a tortoise from Lo river on its shell. Magic squares in China have been traditionally used in various areas of study, including astrology, divination, and the interpretation of philosophy, natural phenomena, and human behavior. Magic squares also permeated other areas of chinese culture. Magic squares most likely traveled from China to India, then to the Arab countries. From the Arab countries, magic squares journeyed to Europe, then to Japan. Magic squares in India served multiple purposes other than the dissemination of mathematical knowledge. During the seventeenth century, serious consideration was given to the study of magic squares. In 1687-88, a French aristocrat, Antoine de la Loubere, studied the mathematical theory of constructing magic squares. In 1686, Adamas Kochansky extended magic squares to three dimensions. During the latter part of the nineteenth century, mathematicians applied the squares to problems in probability and analysis. For example, attractive patterns may be obtained by connecting consecutive numbers in some special classes of magic squares. Today, magic squares are studied in relation to factor analysis, combinatorial mathematics, matrices, modular arithmetic, and geometry. The magic, however, still remains in magic squares. In the times of information technology, magic squares have found many practical applications in artificial intelligence, graph theory, game theory, experiment designs, industrial arts, electronic circuits and location analytics etc., and probably will extend to more innovated applications.

Though it is an unsolved open problem to exactly count the number of magic squares of order greater than 5, researchers find the number of magic squares is very huge, and increases exponentially with the order. By statistical estimation, the number of magic squares of order 6 reaches around 10^{19}, which is already an astronomic number, thus the magic squares of order 6 will be inexhaustive [1,2]. Although there exist some deterministic and predefined methods of magic square construction, they can not be used to construct magic squares in an arbitrary way[3,4,5]. No researches have been made on the distribution of magic squares in the permutation space ($n^2!$) of the consecutive natural numbers from 1 to n^2. If this distribution is evenly and if we take magic square construction as a search for a solution that meets the mathematic condition of magic square, is there thus a search algorithm which makes evenly random sampling in the whole permutation space ($n^2!$)? The author has given a positive answer to this question, and proposed a fast construction algorithm based on evolutionary computation, which can randomly generate magic squares of any order.

This work is financially supported both by the National Science Foundation of China, under grants 60473011, and the fundamental research project No. JC04-06-015 of National Univ. of Defense Tech., China.

The number of magic squares is extremely huge, even the number of magic squares of order 6 is an astronomic number. Can magic square be used to be a fundamental mathematic problem of cryptology, formulate a new encryption principle and further develop some new encryption methods? As a result, we have found three fundamental mathematic problems, i.e. “how to fill out an unfinished magic square”, “how to separate the modular sum of magic squares”, and “how to restore a shuffled magic square”. No general efficient algorithms exist for these three mathematic problems, and they make compensations for each other and formulate a new mathematic principle for a two-way dynamic identity authentication, which can be widely applied in many fields, such as network identification and authentication, electronic tag, access control, key management and assignment, micro-payment, digital anti-fake technique and etc..

2 The Fast Evolutionary Algorithm for Magic Squares

Researches show that the number of magic squares is too huge to imagine. Even excluding those isomorphic magic squares obtained by rotation and reflection, the number of distinct magic squares of order 1,2,3,4,5 are 1, 0, 1, 880, 275305224, respectively. Pinn and Wieczerkowski (1998) estimated the number of magic squares of order 6 to be （1.7745±0.0016）× 10^{19} using Monte Carlo simulation and methods from statistical mechanics[2], but the exact number is still unknown. Further statistical analysis reveals that the number of magic squares increases exponentially with the order, but this does not mean that, higher the order of magic square is, harder the construction of the magic square will be. By statistic analysis, however, the density of magic squares in the possible permutations of the first n^2 natural numbers decreases by around one million times when the order of magic square increases by one, meaning that the difficulty of searching a magic square increases exponentially with the order. According to this rule, it can be statistically derived that the density and number of magic squares of order 7 is around 10^{-29} and 10^{33}, the density and number of magic squares of order 8 is around 10^{-35} and 10^{54}, the density and number of magic squares of order 9 is around 10^{-41} and 10^{79}, and the density and number of magic squares of order 10 is around 10^{-47} and 10^{110} ……

There exist many conventional construction methods that can generate magic squares only by deterministic rules. Most of the methods including the Siamese method, the “lozenge” method, the “LUX” method and etc., deterministically compose a magic square directly by a specific procedure, without any randomness involved. These deterministic methods cannot construct any other possible magic squares, say nothing of composing special magic squares with additional properties. A successful construction of some special classes of magic squares like panmagic squares, inlaid magic squares and trebly magic squares, is usually a painstaking grinding of someone’s wisdom and willpower, which sometimes use up one’s all life. Fundamentally differing from the deterministic construction methods, it will become a problem of combinatorial optimization if we take the magic square construction process as stochastic search for a solution in the permutation space of the first n^2 consecutive natural numbers. It deserves a deep study on carefully designing a good optimization algorithm so that the “shut” probability for any magic square is statistically the same. The author, xie tao, having worked for years along this direction, finally invented a fast probabilistic construction algorithm based on evolutionary computations[6].

The probabilistic construction algorithm of magic square based on evolutionary computations can quickly and randomly generate magic squares of any order. Evolutionary computations are a population-based stochastic learning method inspired from Darwin’s evolution mechanism of species, i.e. the natural selection theory, which explains the evolution of species as a process of surviving the fittest by eliminating the unfit individuals[7]. Evolutionary computations are often used as an adaptive global search and optimization method, if the initial population is set in a uniformly random way, and if all the genetic operations are in a uniformly random distribution, then it will be a uniformly random sampling process to search the solution in a uniformly distributed space. Using 2.4GHZ/256M Pentium IV PC, this algorithm can generate around 1500 magic squares of order 7 or 1000 magic squares of order 10 in one minute, or other magic squares of high order in decreased speed. The magic square is generated partly depending on the initialized population, different initialized population will produce different magic square, each magic square obtained by the algorithm can be taken as the result of a uniformly random sampling process in the magic square space. Any magic square in the whole space can be shut in statistically the same probability in each stochastic search process, but the magic square actually shut in each process is completely random and can not be predicted. In addition, the probability of shutting the same magic square in two different processes is nearly zero.

3 New Magic Square Problems

Based on the probabilistic evolutionary algorithm of magic square construction, we have discovered some new fundamental problems in magic squares, respectively they are the unfinished magic square problem, the modular sum separation problem of magic squares, the magic square shuffling problem, and the magic square encryption problem as well [8]. These are new hard problems in combinatorics, no polynomial time algorithms exist or can be found in the predictable future.

3.1 How to fill out an unfinished magic square

magic square M

35	31	8	3	5	29
23	9	27	12	6	34
19	21	11	25	33	2
15	16	13	28	7	32
1	20	30	26	24	10
18	14	22	17	36	4

half magic square M_1

35	31		3		
		27		6	34
19	21	11			2
15		13		7	
1			26	24	10
		22	17		

half magic square M_2

		8		5	29
23	9		12		
			25	33	
	16		28		32
	20	30			
18	14			36	4

Fig1：complementary half magic squares

If a randomly constructed magic square M is randomly and evenly separated into two complementary half magic squares M_1 and M_2, there exists no general efficient (polynomial time) algorithm by which one half M_1 can be derived from the other half M_2, so that the combined matrix be a magic square, and vice versa. We call this the unfinished magic square problem, which is an open mathematic problem, as shown in Fig 1. In general, if one half magic square M_1 is used as a lock, the other half M_2 can be its corresponding key, and vice versa. The key can not be computationally derived from its lock, and the lock can not be computationally derived from its key. As for magic square of order 7, 10^{34} magic square based digital locks can be constructed. For example, when we use the brute force attacks, the time complexity will reach $25!\approx 1.55\times 10^{25}$ even if we know the “lock”; when we use the associated system of equation, around 9 numbers are undetermined, and what is more is that the number of undetermined numbers increases very fast with the order of magic square, thus the computational complexity to fill a half magic square increases fast with the order of magic square.

3.2 How to Separate the Modular Sum of Magic Squares

If two magic squares of order n are summed modulo n^2+1 element by element to get a natural number matrix S within n^2+1, it is computationally hard to decide whether the matrix S is the addition modulo n^2+1 of two magic squares. In general, $k(\geq 2)$ magic squares of order n are summed modulo n^2+1 element by element to get a natural number matrix S within n^2+1, it is computationally hard to decide whether the matrix S is the addition modulo n^2+1 of k magic squares. We call this the modular sum separation problem of magic squares. This principle can be applied in many fields, such as key management, identification, multi-party identification, micro-payment and digital anti-fake technique etc., it is high in security and efficiency, easy to be implemented in integrated logic circuits. As shown in Fig.2, for example, it is computationally hard to separate the matrix in the right into the two magic squares in the left, so that the matrix in the right be equal to the modular sum of the two magic squares in the left.

magic square 1

35	31	8	3	5	29
23	9	27	12	6	34
19	21	11	25	33	2
15	16	13	28	7	32
1	20	30	26	24	10
18	14	22	17	36	4

magic square 2

21	2	33	15	35	5
12	14	25	4	22	34
29	16	10	32	17	7
1	36	19	24	23	8
28	13	6	27	11	26
20	30	18	9	3	31

magic square sum

19	33	4	18	3	34
35	23	15	16	28	31
11	0	21	20	13	9
16	15	32	15	30	3
29	33	36	16	35	36
1	7	3	26	2	35

Fig2：Modular Sum of Magic Squares

Magic square signature can be directly derived from the modular summation principle of magic squares. Construct a magic square as the private signature key, any randomly generated magic square can be summed modulo n^2+1 with the signature magic square to get a matrix, where n is the order of magic squares. Then, the modular difference between the matrix and the signature magic square must be a magic square. By this way, a matrix can be easily proved whether it is produced by the verifier who possesses the signature magic square, since any fabricated matrices without knowing the signature magic square can not pass the modular difference verification. Particularly, the signed matrix (modular sum matrix) is unique, since there are no two different magic squares, whose signed matrixes with the same signature magic square are the same.

Based on the magic square signature, the unfinished magic square problem can be extended to an unfinished magic square signature problem. If an almost half number of elements are deleted from a signed magic square, it is computationally very hard to find a proper permutation of the deleted elements, so that the unfinished magic square signature problem can be filled in with the deleted elements to restore the signed matrix. The solution to the unfinished magic square signature problem is unique, it can be directly derived from the uniqueness of the signed magic square, since there are no two different permutations of the deleted elements by which the unfinished magic square signature problem can be filled in to form a valid signed magic square, as shown in Fig.3.

signed matrix

19	33	4	18	3	34
35	23	15	16	28	31
11	0	21	20	13	9
16	15	32	15	30	3
29	33	36	16	35	36
1	7	3	26	2	35

signed half matrix 1

19	33		18		
		15		28	31
11	0	21			9
16		32		30	
29			16	35	36
		3	26		

signed half matrix 2

		4		3	34
35	23		16		
			20	13	
	15		15		3
	33	36			
1	7			2	35

Fig3: complementary half signed matrixes

3.3 Magic Square Permutation

Design a matrix permutation list, by which each element in a $n \times n$ original matrix is transferred to a new different position in a new $n \times n$ matrix, so that any two elements in the same column, row or diagonal, do not lie in the same column, row or diagonal after the permutation process is completed This process is called perfect permutation, the permutation list is called perfect permutation list.

It is an unsolved problem whether there is any perfect permutation. To say the least, a perfect permutation list must be composed of two different Latin squares even if the diagonals are not taken into account, but not any two different Latin squares can certainly compose a perfect permutation list.

For an example, the following permutation list is to map each element in a 6x6 matrix to a different position in another new 6x6 matrix.

Permutation List

(5, 1)	(3, 4)	(2, 0)	(5, 2)	(2, 2)	(1, 2)
(0, 3)	(5, 4)	(3, 0)	(4, 2)	(5, 5)	(0, 4)
(1, 3)	(3, 5)	(4, 0)	(1, 1)	(5, 0)	(4, 3)
(4, 5)	(2, 5)	(2, 1)	(1, 5)	(4, 1)	(2, 3)
(0, 5)	(5, 3)	(3, 3)	(1, 0)	(0, 1)	(2, 4)
(0, 2)	(4, 4)	(3, 1)	(0, 0)	(3, 2)	(1, 4)

The dual element (i, j) indicates the new position at row i and column j in the new matrix, where the corresponding element in the original matrix is to be transferred. The upper left element (5,1) in the permutation list is to put the corresponding element at row 5 and column 1, by this way each element is transferred to a new position in the new matrix without duplicates. This list is not a perfect permutation for at least two elements in the same row or column or diagonal are still mapped into the same row, column or diagonal. We call this permutation list an approximately perfect permutation, since each element except the down right one is mapped to a different row, column or diagonal. Being permutated by this way, the mathematical correlation of elements in a magic square will be obliterated.

3.4 Magic Square Encryption

Keep a magic square and a permutation list as the private encryption key, any randomly constructed magic square (plaintext) is firstly summed modulo $n^2 + 1$ with the private magic square to get a signed magic square, this signed magic square is then permutated according to the private permutation list to get a cipher matrix (cryptograph). This process is called a round of magic square encryption, where n is the order of magic squares.

Researches show that, a prime number of rounds of magic square encryption with a specially designed permutation list can protect it from differential attacks, to which one round of magic square encryption may be subject. By using a prime number of rounds of magic square encryption, it is computationally hard to derive the private magic square and permutation list from the magic square and its corresponding cipher matrix, no matter how many you obtain. The principle of magic square encryption can be applied in the fields of network authentication, access control and micro-payment and etc..

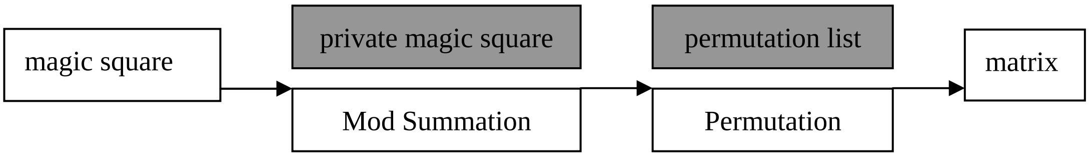


Fig4：one round of magic square encryption

3.5 How to Restore a Shuffled Magic Square

Randomly construct a magic square M of order n, and make a permutation of this magic square according to a fixed approximately perfect permutation list P to get a random matrix X, at the same time, generate an evenly random 0-1 $n \times n$ shuffling matrix S. By the shuffling matrix S, the matrix X is cut into two complementary half matrices X_1 and X_2, which are associated with 1s and 0s in the shuffling matrix S, respectively. Pick up the non-zero elements into a vector and delete the zero elements from the two complementary half matrices by the rule of "from left to right and from top to bottom", a n^2-sized vector V is obtained as in Fig.5. It is very hard to restore the original magic square M from the result vector V even when the permutation list P is known. Equivalently, it is computationally hard to derive the 0-1 shuffling matrix S from both the n^2-sized vector V and permutation list P. We call this a magic square shuffling problem. A reverse process to this magic square shuffling is to recover a magic square from the vector V using its corresponding shuffling matrix S, by which we can verify whether the shuffling matrix S is just what we are going to look for. The vector V and its corresponding shuffling matrix S make up a pair of authentication codes, which can be specially used in digital anti-fake technique and message authentication code hiding.

magic square M

35	31	8	3	5	29
23	9	27	12	6	34
19	21	11	25	33	2
15	16	13	28	7	32
1	20	30	26	24	10
18	14	22	17	36	4

permutation list P

(5,1)	(3,4)	(2,0)	(5,2)	(2,2)	(1,2)
(0,3)	(5,4)	(3,0)	(4,2)	(5,5)	(0,4)
(1,3)	(3,5)	(4,0)	(1,1)	(5,0)	(4,3)
(4,5)	(2,5)	(2,1)	(1,5)	(4,1)	(2,3)
(0,5)	(5,3)	(3,3)	(1,0)	(0,1)	(2,4)
(0,2)	(4,4)	(3,1)	(0,0)	(3,2)	(1,4)

permutated matrix X

17	24	18	23	34	1
26	25	29	19	4	28
8	13	5	32	10	16
27	22	36	30	31	21
11	7	12	2	14	15
33	35	3	20	9	6

Shuffling matrix S

0	1	0	1	0	1
1	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	1	0
0	1	1	0	0	1
0	0	1	1	1	1

half matrix X_1

	24		23		1
26	25	29	19		
8					16
27			30	31	
	7	12			15
		3	20	9	6

half matrix X_2

17		18		34	
				4	28
	13	5	32	10	
	22	36			21
11			2	14	
33	35				

36-elements Vector

24	23	1	26	25	29	19	8	16	27	30	31	7	12	15	3	20	9
6	17	18	34	4	28	13	5	32	10	22	36	21	11	2	14	33	35

Fig.5: an example of magic square shuffling

3.6 How to Restore a Shuffled Complementary Half Matrix

Assume a $n \times n$ matrix M is evenly and randomly cut into two complementary half matrixes M_1 and M_2, let vector V have n^2 elements and be initialized to a 0 vector , the evenly random shuffling matrix $B = \lfloor b_{ij} \rfloor_{n \times n}$, $b_{ij} \in \{0,1\}$.

Firstly, the non-zero elements associated with 1s in B of the half matrix M_2 are transferred into the vector V from the beginning in sequence, then the left non-zero elements associated with 0s in B of the half matrix M_2 are transferred into V at the next consecutive positions in sequence. In this way, V is obtained as the shuffled result.

half matrix M_2

0	10	0	0	16
20	0	0	24	0
21	17	14	0	0
2	0	25	0	22
0	0	0	3	13

shuf matrix B

1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
0	1	0	1	0
0	1	0	1	1

Shuffled result : the vector V

16	20	21	14	3	13	10	24	17	2	25	22	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig6: to shuffle a half matrix M_2

half matrix M_1

15	0	6	18	0
0	11	1	0	9
0	0	0	8	5
0	4	0	12	0
7	23	19	0	0

shuf matrix B

1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
0	1	0	1	0
0	1	0	1	1

matrix M

15	10	6	18	16
20	11	1	24	9
21	17	14	8	5
2	4	25	12	22
7	23	19	3	13

Shuffled result : the vector V

16	20	21	14	3	13	10	24	17	2	25	22	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig7: to restore the matrix M

For example, assume a 5×5 matrix M be cut into two half matrixes M_1 and M_2. Beginning from the upper left element and by the rule of "from left to right and from top to bottom", the non-zero elements in M_2 which are associated with 1s in B are recorded into the vector V in sequence, we get $V = (13,24,4,11,2,8,0,0,0,\cdots,0)$; then, beginning from the upper left element and by the rule of "from left to right and from top to bottom", the non-zero elements associated with the 0s in B of the half matrix M_2 are recorded in sequence into V at the next consecutive positions. Finally, we obtain a shuffled vector V as shown in Fig.6.

It is a reverse process to restore the original half matrix M_2 from the shuffled vector V, in which the complementary half matrix M_1 is needed and can be taken as the key of the reverse operation. M_2 is restored using the shuffled result V, the shuffling matrix B and the corresponding complementary half matrix M_1 together, as a result, the original matrix M is also restored. The restoration process is represented as $V+B\xrightarrow{M_1}M$.

The restoration process is separated into two procedures. In the first procedure, the elements in matrix M with respect to 1s in B are restored. Beginning from the upper left element of the matrix M_1, and proceeding by the rule of “from left to right and from top to bottom”, the element in matrix M is set to the corresponding element in M_1 if the element of M_1 associated with 1s in B is not zero, otherwise, the element in the vector V at the current position, and the position of vector V steps forward by one element. In the second procedure, the elements in matrix M with respect to 0s in B are restored. In the same way, beginning from the upper left element of the matrix M_1, and proceeding by the rule of “from left to right and from top to bottom”, the element in matrix M is set to the corresponding element in M_1 if the element of M_1 associated with 0s in B is not zero, otherwise, the element in the vector V at the current position, and the position of vector V steps forward by one element. By these two procedures, the matrix M is completely restored as shown in Fig.7.

When the matrix M is a random magic square, two complementary half magic squares M_1 and M_2 can be used as a technique of dynamic authentication.

4 A Paradigm of Magic Square Application in Identity Authentication

Identity authentication is a pre-requisite technique in network information security, it is also the foundation of electronic business and government affairs. A general principle of identity authentication is to make a comparison between the specific information or some special computation capability provided by the party to be verified and that possessed by the verifier. The technique of challenge-response is a type of dynamic secure identity authentication method. In a challenge-response way, a client possesses a special function, which is usually embedded in an identification card, and a copy of the corresponding function is kept in the authentication server. When a client is logging in a distant web server, the web server asks the authentication server for the way to identify the client, and the authentication server responses with a challenge-response type of authentication and a corresponding random challenge number to the web server. The web server sends this challenge number to the client, the client makes use of the special function to compute a response number with this challenge number and sends it back to the web server. The web server sends the response number to the authentication server, and the authentication server makes a comparison between the response number and the number obtained by its special function using the challenge number as input. The customer can be validated if these two numbers are equal. Usually, one-way hash functions and some encryption algorithms like DES and RSA are used as special functions in the challenge-response type of identity authentication.

A new type of two-way challenge-response identity authentication can be designed if the three principles are applied in a composite way[8,9], including the unfinished magic square problem, the modular sum separation problem of magic squares and the magic square shuffling problem. Assume a two-way identity authentication is necessary between the web server and the client called Alice. Alice’s identification information can be derived from the magic square $M(Alice)$, which is generated specifically for Alice. $M(Alice)$ is randomly separated into two complementary half magic squares $M_1(Alice)$ and $M_2(Alice)$, $M_1(Alice)$ is firstly signed by a fixed magic square and then permuted by a fixed permutation list to obtain an encrypted matrix $W_1(Alice)$, where the fixed magic square and permutation list is taken as the private encryption key. $W_1(Alice)$ is kept in the data bank of the web server as Alice’s registered information, $M_2(Alice)$ is directly stored in the identification card as Alice’s private identification information. The identification card is also protected with a personal identity number in case of peculation. The verification card in the web server is integrated with the encryption key (the signature magic square and permutation list) as well as their corresponding processing logic, both the verification and identification cards are embedded with the shuffling and restoring algorithms as well as the magic square verification algorithms, and a soft module is kept both in the web server and the client side, which is used to generate evenly random shuffling matrixes. The two-way identity authentication process involves two independent procedures, one is to verify Alice’s identity by the web server’s verification card, the other is to verify the web server by Alice’s identification card.

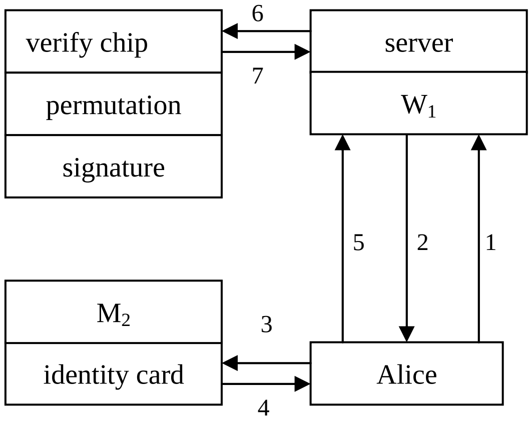
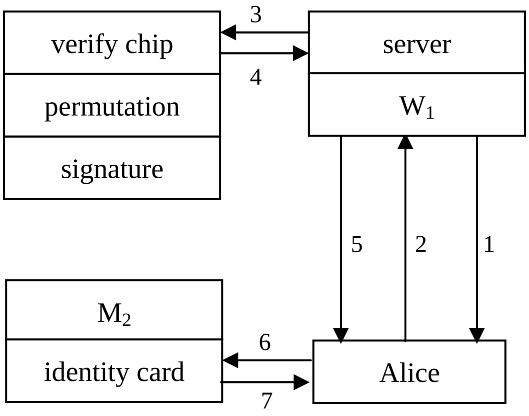


Fig8：the server verifies Alice.



The

Fig9：Alice verifies the server

The procedure to verify Alice's identity is as follows:

1) Alice is logging in the web server;
2) An evenly random shuffling matrix $B(Web,t)$ is generated by the web server and sent to Alice;
3) Alice sends the shuffling matrix $B(Web,t)$ to her identification card;
4) According to the shuffling matrix $B(Web,t)$, the half magic square $M_2(\mathrm{Alice})$ is shuffled into an authentication vector $V(Alice,t)$ by the shuffling algorithm (see 3.6 for details) embedded in the identification card, and sent back to Alice;
5) Alice sends the vector $V(Alice,t)$ to the web server;
6) The web server sends $W_1(Alice),V(Alice,t)$ and $B(Web,t)$ together to its verification card;
7) By the permutation list and signature magic square, $W_1(Alice)$ is decrypted into the half magic square $M_1(Alice)$, $M_1(Alice)$ and $V(Alice,t)$ are used in a combined way with $B(Web,t)$ to restore a matrix M , a verification is made to ensure if M is a magic square, and the verification result is sent to the web server.

The procedure to verify the web server is as follows:

1) The web server sends a signal to Alice for its authentication, Alice is required to generate an evenly random shuffling matrix $B(Alice,t)$ and sends it to the web server;
2) Alice generates an evenly random shuffling matrix $B(Alice,t)$ and sends it to the web server as an response;
3) The web server sends Alice's registered information $W_1(Alice)$ and the shuffling matrix $B(Alice,t)$ to its verification card;
4) In the verification card, the $W_1(Alice)$ is decrypted into the half magic square $M_1(Alice)$ by using the signature magic square and the permutation list, and $M_1(Alice)$ is shuffled into a vector $V(Web,t)$ according to the shuffling matrix $B(Alice,t)$, then $V(Web,t)$ is sent back to the web server;
5) The web server sends the shuffled vector $V(Web,t)$ to Alice;
6) Alice sends the vector $V(Web,t)$ and the shuffling matrix $B(Alice,t)$ into her identification card;
7) In the identification card, $M_2(\mathrm{Alice})$ and $V(Web,t)$ are used in a combined way with the shuffling matrix $B(Alice,t)$ to restore a matrix M , and a verification is made to ensure if M is a magic square, then the verification result is sent to Alice.

The shuffling matrix $B=\lfloor b_{ij}\rfloor_{n\times n}$ used in the dynamic identity authentication is generated by a 0-1 cellular automata machine or pseudo-random number generator in such an evenly random way that the global state of $B=\lfloor b_{ij}\rfloor_{n\times n}$ is actually not repeated and can not be predicted, and the number of 1s in $B=\lfloor b_{ij}\rfloor_{n\times n}$ is statistically equal to that of 0s. Most of cellular automata machines are irreversible, i.e., it can not go backwards to its original global state step by step[10]. To prevent the half magic square $M_2(\text{Alice})$ from the chosen text attacks, a specially designed cellular automata machine can be embedded in the verification card and Alice's identification card so that the shuffling matrix for $M_2(\text{Alice})$ can not be pre-designed for cryptanalysis.

When verifying magic square within both the authentication and verification cards, the consecutive natural numbers from 1 to n^2 in matrix M must be checked for their uniqueness before each row, column and diagonal are verified for their sum, respectively.

5 Conclusion and Some Discussion

Many fantastic properties exist in natural numbers, scientists from different disciplines make every endeavors to discover these pretty mathematical properties and try to make uses of them. Magic square is the earliest mathematical problem found by China which indicates the great law of consecutive natural numbers, but no more discoveries has been made in thousands of years that is practical and applicable till this paper appears. The discoveries made in this paper, including the unfinished magic square problem, the modular sum separation problem of magic squares, the magic square shuffling problem as well as the magic square encryption problem, are multi-discipline research results in intelligent computations, combinatorics and cryptography, their practical applications can be widely found in many fields, such as network authentication, electronic tag, access control, key management and assignment, micro-payment, digital anti-fake technique and etc.. Based on these new magic square problems, several invention patents have been applied and some of them have been put into use, among which a two-way magic square based dynamic identity authentication protocol has been depicted in this paper.

The unfinished magic square problem, the modular sum separation problem of magic squares, the magic square shuffling problem as well as the magic square encryption problem, are new combinatorial problems brought forward in this paper with respect to random magic squares. Do there exist any polynomial time algorithms that solve these problems, or whether these problems are computationally decidable[11]? It is theoretically significant both in the theory of computation and combinatorics, thus deserving further study and discussions.

References

1. Abe G. Unsolved Problems on Magic Squares. Disc. Math. 127, 1994,3-13
2. Pinn K. and Wieczerkowski C. Number of Magic Squares from Parallel Tempering Monte Carlo. Int. J. Mod. Phys. C 9, 1998,541-547
3. Madachy, L. S.. Magic and Antimagic Squares. Ch.4 in Madachy's Mathematical Recreations. New York: Dover, 1979, 85~113
4. Kraitchik M. Magic Squares. Ch.7 in Mathematical Recreations. New York: Norton, 1942,142-192
5. Berlekamp, E. R., Convay, J. H., Guy, R. K.. winning ways for your mathematical plays, Vol.2: Games in particular. London: Academic Press, 1982,778~783
6. Tao Xie, Lishan Kang. An Evolutionary Algorithm for Magic Squares. In 2003 Congress on Evolutionary Computation[C], Canberra, Australia, Dec., 2003, 2:906-913
7. Back T, Hoffmeister F, Schwefel H. P. A Survey of Evolution Strategies. In: Below R. K. and Booker L. B. eds. Proc. of the 4th ICGA, San Diego, 1991. San Mateo: Morgan Kauffman Publishers,1991,2~9
8. Tao Xie. A magic square based signing method for identification and authentication, State Intellectual Property Office of China, Patent No:200410046922.2, 2004
9. Tao Xie, Chen HuoWang, Kang Lishan. A unified method of magic square-based two-way certificate authentication and key transferring, State Intellectual Property Office of China, Patent No:02114288.2, 2002.
10. Jarkko Kari. Reversibility of 2D Cellular Automata is Undecidable. Phisica D 45,1990,379~385.
11. Harry R. Lewis, Christos H. Papadimitriou. Elements of the Theory of Computation, 1998, Second Edition. Prentice Hall, Inc.