Two Matrices for Blakley's Secret Sharing Scheme

Xiali Hei, Xiaojiang Du
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
Email: {xiali.hei, dux}@temple.edu

Binheng Song
Department of Mathematical Sciences
Tsinghua University
Beijing, 100084, China
Email: bsong@math.tsinghua.edu.cn

***Abstract*–The secret sharing scheme was invented by Adi Shamir and George Blakley independently in 1979. In a (k, n)-threshold linear secret sharing scheme, any k-out-of-n participants could recover the shared secret, and any less than k participants could not recover the secret. Shamir's secret sharing scheme is more popular than Blakley's even though the former is more complex than the latter. The reason is that Blakley's scheme lacks determined, general and suitable matrices. In this paper, we present two matrices that can be used for Blakley's secret sharing system. Compared with the Vandermonde matrix used by Shamir's scheme, the elements in these matrices increase slowly. Furthermore, we formulate the optimal matrix problem and find the lower bound of the minimal maximized element for k=2 and upper bound of the minimal maximized element of matrix for given k.**

***Index Terms*–linear threshold cryptography; linear secret sharing; Pascal matrix**

I. Introduction

The secret sharing scheme was invented by Adi Shamir and George Blakley independently in 1979. Until now, well-known secret sharing schemes in the literature include Shamir's [2] based on polynomial interpolation, Blakley's [1] based on hyperplane geometry. Asmuth and Bloom introduced an arithmetic secret sharing scheme [3] based on Chinese Remainder Theorem, which is essentially different from that of Shamir's and Blakely's schemes.

These schemes are called (k, n) threshold secret sharing schemes since the secret is distributed among n participants and only k or more participants can recover the secret. The dealer distributes the secret among n participants. Each participant has his or her own piece of secret called share. The secret is revealed, if any k or more of the shares gather. While Shamir used polynomial-based technique for sharing a secret among n participants, Blakley used a geometric approach. Shamir's technique creates a $(k-1)$ degree polynomial with random coefficients in the range [0, p), where p is a prime number. Secret is the constant term of the polynomial. Lagrange's interpolation technique is used for the reconstruction of the secret from any k or more shares. Blakley's technique assumes that secret is a point in a k-dimensional space. Hyperplanes intersecting at this point are used to construct the shares. Coefficients of n different hyperplanes constitute the corresponding n shares. In these two schemes, a secret is partitioned among n participants. Unless k or more shares are gathered, secret cannot be recovered.

Both Shamir's and Blakley's schemes are linear threshold secret sharing schemes: As Karnin et al. [8] observed, Shamir secret sharing scheme is a subclass of a broader class of linear secret sharing. The polynomial share computation can be represented as a matrix multiplication by using a Vandermonde matrix. Similarly, the secret and the shares of the Blakley's scheme can be represented as a linear system $Cx = y$ where the matrix C and the vector y are obtained from the hyperplane equations. At the same time, Blakley's scheme is the same as Shamir's polynomial system after adding restrictions on which planes are usable as shares, and it is a perfect secrecy system [7].

Shamir's secret sharing scheme is more popular than Blakley's scheme. One reason is that Blakley's scheme lacks actual implementations. In [4], Blakley et al. only provided a guideline on how to design a matrix of linear systems for perfect secrecy, and no actual matrix was given. Recently, researchers began to use Blakley's geometry-based secret sharing approach in the area of secret image sharing [9, 10]. Chen et al. [9] and Tso [10] independently applied Blakley's scheme for secret image sharing. et al. Ulutas et al. [11] proposed an enhanced scheme for secret image sharing, which adopts Blakley's secret sharing method and Steganography together to share the secret and create meaningful shares. As for threshold cryptography, Bozkurt et al [5] proposed the first threshold RSA signature scheme that uses Blakley scheme as the underlying secret sharing scheme. Literature [6] discussed the same scheme in details.

Blakley's method [1] uses principles of geometry to share the secret. According to this scheme, secret is a point in a k-dimensional space, which is the intersect point of all the hyperplanes. Affine hyperplanes in this space represent n shares. Blakley secret sharing scheme can be represented as a linear system $Cx \bmod p = y$. The general full rank matrix C is the critical data. In [6], the authors use matrix M_1 (as shown in I-1); in [11], the authors use a matrix M_2 (as shown in I-2). When the secret is too large, say 1024 bit, p should be larger than 2^{1024}, which is a very large number. So, the operation mod p does not affect the result. These matrices have the problem of fast coefficient expansion. That is, as t increases, the coefficients (elements in these matrices) will become very large. If matrix M_1 is used, to get the exact solution to the linear system, $a_1, a_2, a_3, ..., a_t$ should be integer. If we assume that $a_1, a_2, a_3, ..., a_t$ are in nondecreasing order, that means $a_1 < a_2 < a_3 < ... < a_t$. To minimize a_t, let a_1 be 1, then a_t should be t at least, so the last element a_t^{t-1} in matrix M_1 should be not less than t^{t-1}. When t is greater than 20, this

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

integer is very large. It is difficult to compute and store such a large integer. Also, it requires lots of computations to get the $\det(M_1)$. For matrix M_2, there is no general expression of the matrix's element. Also, the coefficient is big when $k = 3$.

$$M_1 = \begin{pmatrix} 1 & a_1 & a_1^2 & . & a_1^{t-1} \\ 1 & a_2 & a_2^2 & . & a_2^{t-1} \\ \cdots & & & & \\ 1 & a_t & a_t^2 & . & a_t^{t-1} \end{pmatrix} \tag{I-1}$$

$$M_2 = \begin{pmatrix} 10 & 12 & 2 \\ 18 & 12 & 18 \\ 24 & 27 & 27 \end{pmatrix} \tag{I-2}$$

In this paper, we propose two matrices for the Blakley secret sharing scheme. In order to obtain the exact solution to the linear system, matrices with float elements cannot be used. Hence, we use matrices with integer elements. Zero elements in the matrix decrease the dimension of the matrix (hyperplane), and may cause information leakage. Hence, there should be no zero elements in the matrix. Coefficient expansion increases the required storage and computation resource. In this paper, we present two full rank positive integer matrices that can avoid fast coefficient expansion and meet all above requirements. The computation and storage overhead of the two matrices are much smaller than the Vandermonde matrix.

We summarize our contributions as follows:

1. We propose two new matrices for the Blakley secret sharing scheme, and they perform much better than the Vandermonde matrix in terms of coefficient expansion, computation and storage overhead.

2. We formulate the optimal matrix problem for linear secret sharing schemes.

3. We find the lower bound for k=2 and upper bound of the minimal maximized element of matrix for any given k.

The rest of the paper is organized as follows. In Section II we introduce the Blakley's secret sharing scheme. In Section III we present a practical implementation of Blakley's scheme using the first matrix. In Section IV we give the second matrix– Pascal matrix. In Section V we formulate the optimal matrix problem and present the related bounds that we found. We conclude the paper in Section VI.

II. The First matrix

Threshold secret sharing scheme is very useful in that not only does it provide secrecy and reliability but also flexibility. Further more, the property of sharing the secret is ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate, for example, in an electronic voting system or a gambling game system.

A. Coefficients of the Linear Equations

Our approach is based on linear equations. Our secret sharing scheme gives a mechanism where n k-dimension hyperplanes $(x_1, x_2, ..., x_k)$ are distributed to n parties and any k shares can reconstruct the secret. Here, the secret s is x_k, and $x_1, x_2, ..., x_{k-1}$ are random numbers. Also note that after reconstruction, the secret becomes known to all parties. A trusted dealer has the secret s. We design the coefficients of the linear equations (which span n k-dimension hyperplanes) according to the following rules:

$a_{pq} = a_{p-1q-1} + a_{p-1q}$ ($p > 1$,$q > 1$), with $a_{1q} = 1$ and $a_{p1} = 1$.

B. Formula for Matrix Elements

We already have: $a_{pq} = a_{p-1q-1} + a_{p-1q}$ ($p > 1$,$q > 1$), with $a_{1q} = 1$ and $a_{p1} = 1$. Hence, $a_{p,q} - a_{1,q} = \sum_{k=1}^{p-1} a_{k,q-1}$. Or $a_{p,q} = 1 + \sum_{k=1}^{p-1} a_{k,q-1}$=1 + $(p-1)$ + $\sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,q-2}$=$p + \sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,q-2}$.

C. The Secret Sharing Scheme

Step 1: The dealer generates $(k-1)$ random numbers $x_1, x_2, ..., x_{k-1}$, and combines them with the secret s (x_k) to get a point: $(x_1, x_2, ..., x_k)$.

Step 2: The server distributes the n k-dimension hyperplanes to the n parties as follows:

The 1st party gets $y_1 = x_1 + x_2 + ... + x_k$

$\cdots$

The kth party gets $y_k = x_1 + kx_2 + \frac{k^2-k+2}{2}x_3 + ... + 2^{k-1}x_k$

The $k+1$th party gets $y_{k+1} = x_1 + (k+1)x_2 + \frac{k^2+k+2}{2}x_3 + ... + (k + 1 + \sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-3})x_{k-1} + (2^k - 1)x_k$

The $k+2$th party gets $y_{k+2} = x_1 + (k+2)x_2 + \frac{k^2+3k+4}{2}x_3 + ... + (k + 2 + \sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-2})x_k$

$\cdots$

Step 3: To reconstruct the secret from any k parties, we only need to solve the linear equations and get the exact solution (the point). If we know the point, we can obtain the secret, which is the last coordinate value of the point.

After we process the coefficient table, we get the general equations II-3.

$$\begin{array}{l} y_1 = x_1 + x_2 + x_3 + ... + x_k \\ \cdots \\ y_k = x_1 + kx_2 + \frac{k^2-k+2}{2}x_3 + ... + 2^{k-1}x_k \\ y_{k+1} = x_1 + (k+1)x_2 + \frac{k^2+k+2}{2}x_3 + ...(k + 1 + \sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-3})x_{k-1} + (2^k - 1)x_k \\ \cdots \end{array} \tag{II-3}$$

We write down the coefficient matrix A as follows:

$$A = \begin{pmatrix} 1 & 1 & 1 & . & . & 1 \\ \cdots & & & & & \\ 1 & k & \frac{k^2-k+2}{2} & . & 2^{k-1}-1 & 2^{k-1} \\ 1 & k+1 & \frac{k^2+k+2}{2} & . & k+1+\sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-3} & 2^k-1 \\ \cdots & & & & & \end{pmatrix} \quad \text{(II-4)}$$

We split matrix A into two submatrices A_1 and A_2. Matrix A_1 is a $k \times k$ square matrix, and matrix A_2 is a $m \times k$ matrix, where $m = n - k$. According to our design rules, in A_2, $a_{p,q} = a_{p-1,q-1} + a_{p-1,q}$ ($p > 1$,$q > 1$), with $a_{1,i} = 1$ and $a_{i,1} = 1$. Hence, the last number in the ith row is not less than the sum of the last numbers from the 1st row to the (i-1)th row, while the first number in the ith row is equal to the first number in all other rows. Thus, the ith row cannot be represented by a linear combination of the first (i-1) rows. Hence, the rank of A_2 is min(k, m). If we assume $m < k$, then the m row vectors in A_2 are linearly independent.

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \quad \text{(II-5)}$$

$$A_1 = \begin{pmatrix} 1 & 1 & 1 & . & 1 \\ \cdots & & & & \\ 1 & k-1 & \frac{k^2-3k+4}{2} & . & 2^{k-2} \\ 1 & k & \frac{k^2-k+2}{2} & . & 2^{k-1} \end{pmatrix} \quad \text{(II-6)}$$

$$A_2 = \begin{pmatrix} 1 & k+1 & \frac{k^2+k+2}{2} & . & k+1+\sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-3} & 2^k-1 \\ 1 & k+2 & \frac{k^2+3k+4}{2} & . & . & k+2+\sum_{k=1}^{p-1}\sum_{w=1}^{k-1} a_{w,k-2} \\ \cdots & & & & & \end{pmatrix} \quad \text{(II-7)}$$

D. Proof of Correctness

The rank of a $k \times n$ matrix is at most $min(k, n)$. A matrix that has a rank as large as possible is said to have full rank. If the rank of the augmented matrix is equal to the rank of the coefficient matrix, the system must have at least one solution. The solution is unique if and only if the rank is equal to the number of variables. Otherwise, the solution has w free parameters where w is the difference between the number of variables and the rank. This theorem is discovered by Rouch' and Capelli [12].

To prove the correctness, the main challenge is to prove that any k dimension row vectors consisted of coefficients of linear equations are linearly independent.

Lemma 1: The $k \times k$ matrix A_1 is full rank.

Proof:
Let A_{1t} be a matrix made up of the first t rows and the first t columns of A_1.

$$A_{1t} = \begin{pmatrix} 1 & 1 & 1 & . & 1 \\ \cdots & & & & \\ 1 & t-1 & \frac{t^2-3t+4}{2} & . & 2^{t-2} \\ 1 & t & \frac{t^2-t+2}{2} & . & 2^{t-1} \end{pmatrix} \quad \text{(II-8)}$$

Step 1: Perform the row transformations on A_{1t}: we let the $(i+1)$th row subtract the (i)th row, (for $i = t-1, ..., 1$), then the A_{1t} turns out to be:

$$A_{1t} = \begin{pmatrix} 1 & d \\ c & A_{1(t-1)} \end{pmatrix} \quad \text{(II-9)}$$

where c is a zero column vector, d is a row vector with nonzero elements, $A_{1(t-1)}$ is a submatrix of A_{1t} by deleting the last row and the last column. The reason is that $a_{i,j} - a_{i-1,j} = a_{i-1,j-1}$, so every element becomes its left and upper neighbor. Since c is a zero column vector and the first element in first row vector is 1, the first row vector cannot be represented by the rows vectors after it. At this time, the problem becomes how to prove that $A_{1(t-1)}$ is full rank.

Step 2: We repeat Step 1 $t-1$ times, eventually we get a matrix with only one element –1. This element does not equal to 0, so it is full rank.

The above two steps prove Lemma 1. This means that the k row vectors of A_1 are linearly independent. If we pick any $t < k$ row vectors in A_1, they are linearly independent, too.

Corollary 1: A_2 is full rank.

Proof:

Without loss of generality, we let x=min(m,k) to simplify the discussion.

Let A_{2x} be a matrix made up by the first x rows and the first x columns of A_2.

By Lemma 1, we know A_{2x} is full rank, i.e., $rank(A_{2x}) = x$, so the first x rows of the first x columns are linearly independent. If we expand these x rows from x columns to k columns, they are still linearly independent. Hence, Corollary 1 is true.

We pick the first k columns from A, and pick the k row vectors from these first k columns to form a full rank matrix B_k.

Lemma 2: B_k is a full rank $k \times k$ matrix consisted of k row vectors in A, if any row vector of it is replaced with one row vector in $A \setminus B_k$, the new matrix is still full rank.

Proof: Here we use mathematical induction on m.

Base case: when $m = 1$, by observation, k dimension row

vectors in B_k has two cases:

Case 1.1: $(k-1)$ row vectors are from A_1 and one row vector is from A_2, which is $(1, (k+1), \frac{(k+1)^2-(k+1)+2}{2}, ..., 2^k-1)$. Because A_1 is full rank matrix, these $(k-1)$ row vectors made up of coefficients are linearly independent. And the k dimension row vector $(1, (k+1), \frac{(k+1)^2-(k+1)+2}{2}, ..., 2^k-1)$ cannot be represented by these $(k-1)$ row vectors because its last element $2^k-1 = 1+2+4+...+2^{k-1}$, the latter is the sum of the last numbers from the 1st row to the (i-1)th row, while the first number in the ith row is equal to the first number in all other rows, it cannot be spanned by the $k-1$ row vectors in A_1. Thus, these updated k row vectors are linearly independent.

Case 1.2: k row vectors are from A_1, because A_1 is $k \times k$ full rank matrix, these k row vectors are linearly independent.

Combined these two cases, when $m = 1$, the Lemma 2 is correct.

Then, assume that for all positive integers $m = s$, any k row vectors in A are linearly independent.

Suppose these k row vectors in A constitute a new matrix B_k. We also assume i row vectors in B_k are from matrix A_1, the other $(k-i)$ row vectors are from matrix A_2.

When $m = s+1$, if we replace one row vector in B_k with the new row vector $(1, (k+s+1), \frac{(k+s+1)^2-(k+s+1)+2}{2}, ..., k+s+1+\sum_{k=1}^{k+s}\sum_{w=1}^{k-1} a_{w,k-2})$, we called the new matrix is B_{s+1}.

Case s.1: The replaced row vector is from A_1.

Because the remaining $(i-1)$ row vectors in B_{s+1} are from matrix A_1, so they are linearly independent. The new vector $(1, (k+s+1), \frac{(k+s+1)^2-(k+s+1)+2}{2}, ..., k+s+1+\sum_{k=1}^{k+s}\sum_{w=1}^{k-1} a_{w,k-2})$ cannot be represented by these $(k-1)$ row vectors because its last element $k+s+1+\sum_{k=1}^{k+s}\sum_{w=1}^{k-1} a_{w,k-2}) > 2^k-1 = 1+2+4+...+2^{k-1}$, the latter is the sum of the last numbers from the 1st row to the (i-1)th row, while the first number in the ith row is equal to the first number in from 1st row to (i-1)th row, it cannot be spanned by the $i-1$ row vectors in A_1. Thus, these updated i row vectors are linearly independent and the new vector is linearly independent to these remaining $(i-1)$ row vectors from matrix A_1 in B_{s+1}. By Corollary 1, the new vector is linearly independent with the other $(k-i)$ row vectors are from matrix A_2 in B_{s+1}. According to the induction hypothesis, the remaining $(k-1)$ vectors are mutually linearly independent because the all k vectors are mutually linearly independent. So the the rank of new matrix B_{s+1} is still k. That means, the new k vectors are linearly independent.

Case s.2: The replaced row vector is from A_2.

Because the remaining i row vectors in B_{k+1} are from matrix A_1, so they are linearly independent. The new vector $(1, (k+s+1), \frac{(k+s+1)^2-(k+s+1)+2}{2}, ..., k+s+1+\sum_{k=1}^{k+s}\sum_{w=1}^{k-1} a_{w,k-2}$ cannot be represented by these (i) row vectors because its last element $k+s+1+\sum_{k=1}^{k+s}\sum_{w=1}^{k-1} a_{w,k-2}) > 2^k-1 = 1+2+4+...+2^{k-1}$, the latter is the sum of the last numbers from the 1st row to the (i-1)th row, while the first number in the ith row is equal to the first number in from 1st row to (i-1)th row. Thus, the new vector is linearly independent to these remaining i row vectors from matrix A_1 in B_{k+1}. By Corollary 1, the new vector is linearly independent with the other $(k-i-1)$ row vectors are from matrix A_2 in B_{k+1}. According to the induction hypothesis, the remaining $(k-1)$ vectors are mutually linearly independent because the all n vectors are mutually linearly independent. So the the rank of new matrix B_{k+1} is still k. That means, the new k vectors are linearly independent.

Hence, by the second principle of induction, Lemma 2 is true.

Theorem 1: Any k row vectors in matrix A are linearly independent.

Proof:
Initially, we pick k row vectors from A_1. They are row vectors from a full rank matrix. Then, we replace one row with a row vector from A_2, this is case 1.1, so it is correct. After that, we replace one row in the updated k dimension row vectors with one row vector from A_2, and it is correct according Lemma 2. Next, we delete the picked row vector in A_2.

We iteratively do the above until there are no more rows left in A_2, the k row vectors are still linearly independent. Hence, any k row vectors in matrix A are linearly independent. Done.

E. Properties of the First Matrix

1) Computation Complexity and Storage Requirement: The first matrix has similar properties as the Vandemonde matrix. However, it also has some special properties. For example, for any first k participants, the determinant of the corresponding matrix is 1. This reduces the complexity of computing the inverse matrix over the field $GF(p)$.

When k approaches n, the maximum element in the Vandemonde matrix is n^n, and the maximum element in our new matrix is only 2^n. Our matrix saves a lot of computations and storage when n is large. Also, our matrix does not require multiplication operation over the field $GF(p)$. The size of each sharing does not exceed the size of the secret.

2) Extensible and Dynamic: When k is fixed, secret pieces can be dynamically added or deleted without affecting other pieces. Security can be easily enhanced without changing the secret, but by changing the coefficients occasionally and constructing new shares to the participants.

3) Flexibility: In organizations with hierarchy, we can provide each participant with different number of pieces according to his role with the organization. For instance, the CEO can obtain the secret alone, whereas 3 secretaries are required to do so.

III. The Second Matrix–Pascal Matrix

Here, we design another coefficient matrix for above equations according to the following rules:

$a_{pq} = a_{pq-1} + a_{p-1q}$ ($p > 1$,$q > 1$), with $a_{1i} = 1$ and $a_{i1} = 1$ for i is integer. It is a Pascal matrix A_3.

$$A_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & . & 1 \\ 1 & 2 & 3 & 4 & . & k \\ \cdots & & & & & \\ 1 & n & C_{n+1}^2 & C_{n+2}^3 & . & C_{k+n-2}^{n-1} \end{pmatrix} \quad \text{(III-10)}$$

The proof of correctness is similar to the proof in VI.D and our paper [13] uses this matrix in Reed-Solomon code. The elements of A_3 increase much slowly than M_1. When n or k is small, A_3 is also practical.

IV. The Optimal Matrix Problem

A. Formulation

If one wants to find a matrix for the linear secret sharing schemes, the matrix should have the following property:

Property A: For some k, any k (k = min(n, m)) row vectors among the first k column vectors in $n \times m$ matrix O are linearly independent and any $k-1$ row vectors among the first $k-1$ column vectors in matrix O are linearly independent.

The optimal matrix O is formulated as follows:

$$min(max(\ all\ elements\ in\ \mathrm{O})) \quad \text{(IV-11)}$$

B. Lower Bound of the Minimal Maximized Element in O for $k = 2$

When k=2, we assume the largest number in O is u,

n= # of (the relative prime pairs in [1, ..., u])$\times$2 -1 =$[u(u+1)/2 - \sum_{i=1}^{u}(u-i)/(i+1)] \times 2$-1 $\simeq u(u+1) - 2u * logu + 2\sum i/(i+1)$-1= $u(u+1) - 2u \times logu + 2u - 2logu - 1$=$u^2 + (3 - 2logu)u - 2logu - 1 < u^2$ (when $3 < 2logu$, and it means $u \geq 3$).

Hence, $n < u^2$ implies $u > n^{1/2}$, this is the lower bound.

C. Upper Bound of the Minimal Maximized Element in O for Any Given k

Here we relax the condition to be any given k. Suppose that O is one of the optimal matrices for the given k.

$$O_{nk} = \begin{pmatrix} a_{11} & . & . & a_{1k} \\ a_{21} & . & . & a_{2k} \\ \cdots & & & \\ a_{(n-1)1} & . & . & a_{(n-1)k} \\ a_{n1} & . & . & a_{nk} \end{pmatrix} \quad \text{(IV-12)}$$

$$O_{(k-1)(k-1)} = \begin{pmatrix} a_{11} & . & . & a_{1(k-1)} \\ a_{21} & . & . & a_{2(k-1)} \\ \cdots & & & \\ a_{(k-1)1} & . & . & a_{(k-1)(k-1)} \end{pmatrix} \quad \text{(IV-13)}$$

After we delete any one column and any one row, O turns out to be $O_{(n-1)(k-1)}$. If we pick any $k-1$ row vectors of $O_{(n-1)(k-1)}$, we can get matrix $O_{(k-1)(k-1)}$. Because O_{nk} is full rank, using property A we can get $O_{(k-1)(k-1)}$ is still full rank, that means $r(O_{(k-1)(k-1)}) = k-1$.

We assume all coefficients belong to {1,..., u } for some u. Then we have u^{k-1} possible $(k-1)$-vectors. Choose any $(k-2)$ $(k-1)$-vectors. And they can span a $(k-2)$ (or lower) dimensional space. There should be around u^{k-2} vectors in this space. Do this $\binom{n-1}{k-2}$ times. There should be around $u^{k-2}\binom{n-1}{k-2}$ or less vectors in all those spaces.

If $u^{k-2}\binom{n-1}{k-2} < u^{k-1}$, then there is a vector that is not in any of those spaces. Let $u = \binom{n-1}{k-2}$ +1, and it is the upper bound of the minimal maximized element in O for given k, n.

D. Extension

Apart from the above methods, to get a suitable coefficient matrix, one can generate a random $n \times k$ positive integer matrix, and then use a computer program to check if any k row vectors are linearly independent or not. This method may get a better matrix. However, it is very time-consuming. For example, for $n = 40, k = 4$, it need several hours to get a matrix that satisfies the conditions with small coefficients.

V. Conclusion

In this paper, we proposed two new matrices for practical implementation of Blakley's secret sharing scheme. The determinant of the corresponding square matrix is 1, so it is easy to compute its inverse matrix over the field $GF(p)$. And the generation of these matrices does not need multiplication operation under mod p field. Compared with the Vandermonde matrix, our matrices require less computation and storage. Furthermore, we formulated the optimal matrix problem and obtained the lower bound for $k = 2$ and the upper bound for any given k.

VI. Acknowledgment

This research was supported in part by the US National Science Foundation (NSF) under grants CNS-0963578, CNS-1002974, CNS-1022552, and CNS-1065444, as well as the US Army Research Office under grant W911NF-08-1-0334.

References

[1] G. R. Blakley. Safeguarding cryptographic keys. *In proc. of the National Computer Conference*, vol. 48, pp: 313C317, 1979.

[2] A. Shamir. How to share a secret. Communications of the ACM, vol. 22, No. 11, pp: 612-613, 1979.

[3] C. Asmuth and J. Bloom. A modular approach to key safeguarding. IEEE Trans. Information Theory, vol. 29, No. 2, pp: 208-210, 1983.

[4] G. R. Blakley and G. A. Kabatianski. Linear algebra approach to secret sharing schemes. In *proc. of Error Control, Cryptology, and Speech Compression, Lecture Notes in Computer Science*, vol. 829, pp: 33-40, 1994.

[5] I. N. Bozkurt, K. Kaya and A. A. Selcuk. Threshold Cryptography Based on Blakley Secret Sharing. In *Proc. of Information Security and Cryptology 2008*, Ankara, Turkey, Dec. 2008.

[6] I. N. Bozkurt, K. Kaya and A. A. Selçuk Practical Threshold Signatures with Linear Secret Sharing Schemes LNCS vol. 5580 (Progress in cryptology-AFRICACRYPT 2009) pp: 167-178, 2009.

[7] G. J. Simmons, editor, Contemporary Cryptology - The Science of Information Integrity, IEEE Press, 1992.

[8] E. D. Karnin, J. W. Greene and M. E. Hellman. On secret sharing systems. IEEE Transactions on Information Theory, vol. 29, pp: 35-41, 1983.

[9] C. C. Chen and W. Y. Fu. A geometry-based secret image sharing approach. Journal of Information Science and Engineering, vol. 24, no. 5, pp: 1567-1577, 2008.

[10] H. K. Tso. Sharing secret images using Blakley's concept. Optical Engineering, vol. 47, no. 7, pp: 21-23, 2008.

[11] M. Ulutas, V. V. Nabiyev and G. Ulutas. Improvements in Geometry-Based Secret Image Sharing Approach with Steganography. Mathematical Problems in Engineering, vol. 53, pp: 101-110, 2009.

[12] Rouche'–Capelli Theorem, http://en.wikipedia.org/wiki/Rouche'–Capelli–theorem

[13] X. Hei, X. Du and B. Song. A Distributed Login Framework for Semi-structured Peer-to-Peer Networks. In proc. of IEEE ICC 2012, Ottawa, June, 2012.