A New Secure Distributed Storage Scheme for Cloud

- Mathematical Designs and Implementation

Cheong, Hou Teng and Tan, Chih Wei

Pui Ching Middle School, Macau

Supervisor: Mr. Wong, Chan Lam

Abstract

In this report, we have mainly developed an abstract framework of k bit (t,n) secret sharing framework for cloud storage. Such framework is clean and it can be implemented. A successful implementation of the framework would provide users with protection when the system is under the attack on its confidentiality, integrity and reliability. Furthermore, such system has its own encryption by using permutations and tailor made error detection, location and data rescue.

We make use of Lagrange polynomials and take the advantages of the algebraic property "t distinct points on the plane can uniquely determine a polynomial function of degree $t-1$" to design a k bit (t,n) -secret sharing distributed storage. We employ the set with unique factorization property (UFP) so that we simply need to calculate the y intercept of a Lagrange polynomial and then use a look up table to recover a secret. Moreover, the set which has minimum UFP would help us to design storage with smallest containers.

In addition to the algebraic methods, we can utilize the geometric facts that three non collinear points determine a unique circle and four non coplanar points determine a unique sphere to construct k bit $(3,n)$ and k bit $(4,n)$ secret sharing storage respectively. To generalize to arbitrary case, it is straight forward if we have defined the Haar measure on the higher dimensional unit sphere.

The last method is an application of Chinese Remainder Theorem (CRT) and we have designed k bit (t,n) secret sharing distributed storage and one of the designs can produce containers with the half size of the original secret. However, such k is no longer unrestricted and it has to be chosen from an interval.

We have developed a C program for implementing both algebraic, geometric k bit $(3,n)$ secret sharing distributed storages as well as the CRT method. The performances of both algebraic and geometric designs are satisfactory in term of processing time and compressed container size. The container size is even half of the size of the original secret in the CRT case and it is also very speedy.

1. Introduction

The world keeps evolving, so as our life. Alvin Toffler, an America futurist, described in his famous book *The Third Wave* [1], that human progress could be divided into three 'waves': The Agricultural Revolution constitutes the First wave; the Industrial Revolution, the Second Wave and the Third Wave, which is a different world we have just entered, comprises the Information Age based on the revolution brought by Computer Technology. To review from the past, IBM developed the mainframe computer in the 60s of the 20th century; personal computer (PC) become popular in the 90s which followed immediately by information explosion brought by internet. Nowadays, network servers and users exist everywhere in our world, perform various calculation tasks according to different enquires. Now, in the early 21st century, we are living in the Third Wave society which represented as the Cloud Computing Era. Cloud computing has become increasingly important owing to the continuous economic development. Apart from performing calculation and providing storage at supercomputer-level at any time, cloud computing require much lesser cost compared to the supercomputers.

According to SME (Small and medium enterprises) cloud adoption study by Microsoft and Edge Strategies in 2011 (Ref. Figure 1) [2], it indicated that top three workloads addressed by paid cloud service in three years are accounting & payroll, collaboration and File/Storage and back up. Especially, and File/Storage and back up workload is expected to be increased almost double in 2014.

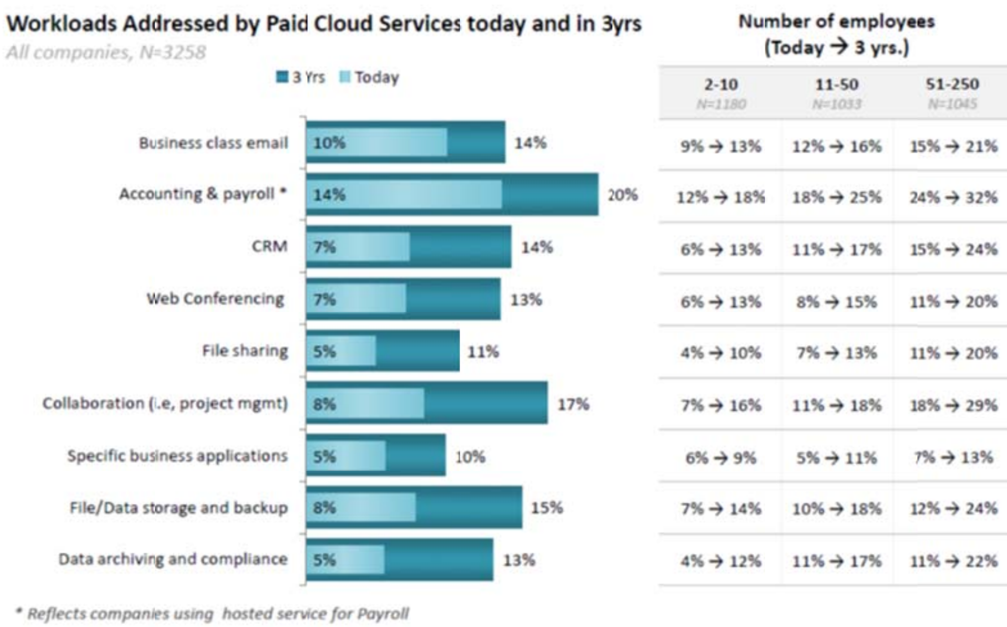

Figure 1. SME cloud adoption study by Microsoft and Edge Strategies in 2011

In spite of the rapid growth of demands of cloud service, people still hesitate to use clouds. A survey by HKPC (Hong Kong Productivity Council) (Ref. Figure 2) [3], we can see that up to 50 % of the samples show their concerns about the security of cloud service which is the major concern in the survey. Also, from an InformationWeek survey 2013 (Ref. Figure 3) [4], security not only occupies the position of the top cloud storage concern but also draws more user attention on the matter.

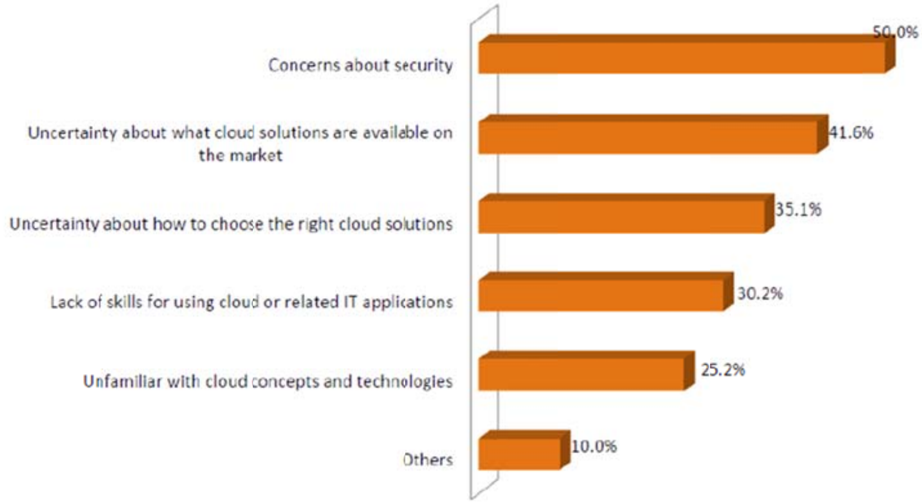

Figure 2. Survey by HKPC on Not Using Cloud in 2012

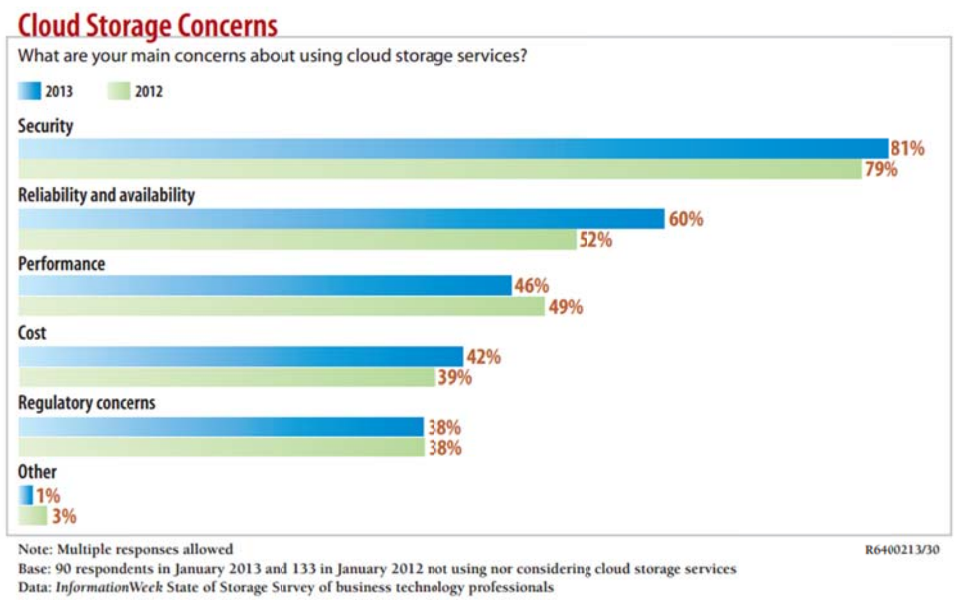

Figure 3. Survey by *InfromationWeek* on Cloud Storage Concern in 2013

Currently, there are many cloud storage providers that let users share and store documents easily online. However, we are lack of control of our documents. For example, when we upload a file to cloud storage, the service provider has not only the total detail of the file but also right to read, duplicate and even transfer the file to anywhere. It will give an opportunity to malicious attackers. Besides, attackers will attack cloud storage server to steal all kinds of documents. Such personal information or documents might be sold or distributed illegally as well as immorally. It hurts benefits of companies, privacy of people and innocent victims themselves.

Usually the cloud storage provider will store files over networks in more than one storage nodes. Such distributed storage will be under three major kinds of attacks. [5]

-The attack on confidentiality reveals stored server contents to attackers;

-The attack on integrity modifies data in victim storage servers without being noticed;

-The attack on reliability makes storage server unavailable to legitimate users.

In this report, we will propose a (t, n) secret sharing secure distributed storage system which can ensure its confidentiality, integrity as well as reliability. Basically, suppose we would like to save a file. What we will do is creating n files called containers which are quite different from the original file. These containers will be store in distinct storages. If we want to restore the file, we need to have at least t containers. These containers will be processed a recovery program. The output of the program is the original file. (ref. Figure 4)

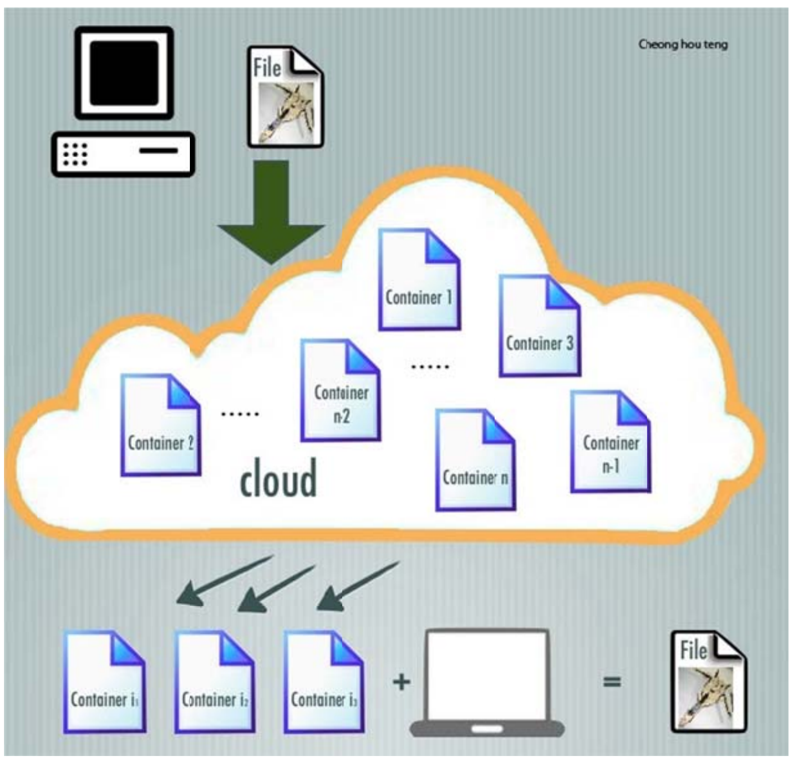

Figure 4. The idea of the proposed system

Besides, we will be expected to have more features such as unauthorized modification detection, location and correction.

Reference

1. Toffler, A. *"The third wave."* Bantam Books, 1989.

2. http://www.microsoft.com , http://www.edgestrategies.com/

3. HKPC, "*Cloud guide book for HK SME*", 《香港中小企业云端方案应指南》 ISBN 9789881-200617, HKPC, 2012.

4. http://reports.informationweek.com/abstract/24/9898/Storage

5. Lan, T., Lee, R.B., Chiang, M. *Reliable and Secure Distributed Storage of Critical Information*, Princeton University Department of Electrical Engineering Technical Report CE-L2008-017, 2008.

2. Literature Revision of (t,n)-Secret Sharing

Secret sharing [1] is currently a very popular topic which is a method of distributing a secret, among a group of users, requiring a cooperative effort to determine the secret. Secret sharing schemes are designed with specific parameters that determine the number of shares needed to uncover the secret, and the overall number of shares in the scheme. The ultimate goal of the scheme is to divide the secret being hidden into n shares, but any subset of t shares can be used together to solve for the value of the secret. Additionally, any subset of $t-1$ shares will prevent the secret from being reconstructed [2]. This is defined as a (t,n) threshold scheme, meaning that the secret is dispersed into n overall pieces, with any t pieces being able to recreate the original secret. We are interested in $(3,n)$ -threshold scheme in this research.

As in [3], we would like to consider the following problem:

4 scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if 3 or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?

It is not hard to show that the minimal solution for this $(3,4)$ problem uses 4 locks and 2 keys per scientist. However, if we increase the number of scientist to 11 and at least 6 of the scientists have to present in order to open the cabinet. We can show that the minimal solution 462 locks and 252 keys per scientist. It is clearly not practical. Moreover, the numbers will increase exponentially as the number of scientist increases. Therefore, other schemes which are more innovative have been proposed.

2.1 Shamir's Scheme

In the Shamir's scheme [3], [4], any t out of n shares may be used to recover the secret. The method is based on the fact that you can fit a unique polynomial of degree $(t-1)$ to any set of t points that lie on the polynomial. The method is to create a polynomial of degree $t-1$ with the secret as the first coefficient and the remaining coefficients picked at random. Next find n points on the curve of the polynomial and give one to each of the shares. When at least t out of the n shares reveal their points, there is sufficient information to fit a $(t-1)$ th degree polynomial to them, the first coefficient being the secret.

Example 2.1.1 (A (3,6)-Shamir's scheme)

Suppose that our secret $S = 1234$. We obtain 2 numbers 166 and 94 randomly. Define a quadratic function

$$f(x) = S + 166x + 94x^2 = 1234 + 166x + 94x^2.$$

We construct 6 points form f as below:

$$(1,1494), (2,1942), (3,2578), (4,3402), (5,4414), (6,5614).$$

In order to reconstruct the secret S, any three of the above points will be enough. Let us consider $(x_0, y_0) = (2,1942), (x_1, y_1) = (4,3402)$ and $(x_2, y_2) = (5,4414)$. We will compute Lagrange basis polynomials:

$$\begin{aligned}
L_0 &= \frac{x - x_1}{x_0 - x_1} \frac{x - x_1}{x_0 - x_1} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3} \\
L_1 &= \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} = \frac{-1}{2}x^2 + \frac{7}{2}x - 5. \\
L_2 &= \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1} = \frac{1}{3}x^2 - 2x + \frac{8}{3}
\end{aligned}$$

Therefore, $f(x) = \sum_{j=0}^{2} y_j L_j(x) = 1234 + 166x + 94x^2$. Hence $S = f(0)$.

2.2 Blakley's Scheme

Blakley's secret sharing scheme [5] is geometric in nature. For a (t, n) secret sharing, we use the fact that any t nonparallel $(t-1)$-dimensional hyperplanes intersect at a specific point. So suppose the secret S is a point in the t dimensional space. Just create n nonparallel $(t-1)$-dimensional hyperplanes as keys. Then any t of them will uniquely determine a point which is the secret point.

Example 2.2.1 (A (2,6) Blakley's secret sharing scheme)

Let S=(0,0). We can create a (2,6) Blakley's secret sharing. Make 6 keys for each share as below:

$$K_1 = x, K_2 = 2x, K_3 = 3x, K_4 = 4x, K_5 = 5x, K_6 = 6x$$

For any K_i and K_j, we are able to solve the intercept point out which is the secret point $(0,0)$.

2.3 Using the Chinese Remainder Theorem

We would like to only illustrate the idea by an example. It is based on the Asmuth-Bloom's Scheme [6]. Let $t=3$ and $n=4$. Let $m_0=3, m_1=11, m_2=13, m_3=17$ and $m_4=19$. Let the secret $S=2$. Pick $\alpha=51$ according to Asmuth-Bloom's Scheme. Then $2+51\times 3=155$. Assign $\left(m_i, 155(\operatorname{mod} m_i)\right)$ to the i th share for $i=1,2,3,4$. To recover the secret, we have one possible 3 of the shares for example, $(11,1),(13,12)$ and $(17,2)$. Then apply to solve the system of equations

$$\begin{aligned} x &= 1 \quad (\operatorname{mod} 11) \\ x &= 12 \quad (\operatorname{mod} 13). \\ x &= 2 \quad (\operatorname{mod} 17) \end{aligned}$$

[i.e. Consider the solutions of the following systems of equations

$$\begin{matrix} x=1 \quad (\operatorname{mod} 11) & x=0 \quad (\operatorname{mod} 11) & x=0 \quad (\operatorname{mod} 11) \\ x=0 \quad (\operatorname{mod} 13), & x=1 \quad (\operatorname{mod} 13), & x=0 \quad (\operatorname{mod} 13) \\ x=0 \quad (\operatorname{mod} 17) & x=0 \quad (\operatorname{mod} 17) & x=1 \quad (\operatorname{mod} 17) \end{matrix}$$

which are 221, 1496 and 715 respectively]. So the solution is
$1\times 221+12\times 1496+2\times 715=19603$ and $19603=155(\operatorname{mod} 11\times 13\times 17)$. Finally, $S=2=155(\operatorname{mod} 3)$.

It is worthy to note that the above method normally cannot be applied directly to practical problems. Further development, modification and design are needed to be made so that a real world problem can be solved.

Reference

1. Martin, R., *Introduction to secret sharing*, [Online]. Available: http://www.cs.rit.edu/~rfm6038/Paper.pdf

2. Stinson, D. R. , "*Cryptography: Theory and Practice*", 2nd ed. CRC Press, 2006.

3. Shamir, Adi , "*How to share a secret*", *Communications of the* ACM 22 (11): 612–613, 1979

4. Wikipedia: Shamir's Secret Sharing

5. Blakley, G. R. , "*Safeguarding cryptographic keys*". *Proceedings of the National Computer Conference* **48**: 313–317. 1979

6. Wikipedia: Secret sharing using the Chinese remainder theorem

3. Framework of *k* bit *t* Secret Sharing Distributed Storage

In this Chapter, we will introduce the various notions related to proposed framework. Besides, we will give simplified examples in order to illustrate the ideas. The algorithms of generating containers and recovering the original secret are also given. A tailor made encryption for container is given by the end of the chapter.

3.1 Secret Shareable Pairs

Let $\mathbb{Z}_k$ be the set of all binary strings with length k, i.e.

$$\mathbb{Z}_k = \{\epsilon_1\epsilon_2\cdots\epsilon_k : \epsilon_i = 0 \text{ or } 1, i = 1, 2, \cdots, k\}$$

Definition 3.1.1: Let $t \geq 3$ be an integer and let Λ be a family of subsets of $\mathbb{R}^d$ such that if

a) $C_1, C_2 \in \Lambda$ and $|C_1 \cap C_2| \geq t$, then $C_1 = C_2$;
b) t is the smallest integer which has property a).
c) Let $p_1, p_2, \cdots, p_t \in C$ and $C \in \Lambda$. There is a method Ψ (or an algorithm) enable us to obtain C from the given t points.

Then Λ is t-secret shareable.

Notation: We would like to denote the set of all the graphs of polynomials of degree $t-1$ by Λ_t. Hence, Λ_3 is the set of all the graphs of quadratic functions.

Example 3.1.2: Recall that a classical algebra result, two quadratic functions agree with each other at three distinct points if and only if they are equal. Then Λ_3 has properties a) and b) in the definition 3.1.1. Assume that (x_1, y_1), (x_2, y_2) and (x_3, y_3). Let

$$L(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} y_3 .$$

Then $L(x_1) = y_1, L(x_2) = y_2$ and $L(x_3) = y_3$. Hence, (x_1, y_1), (x_2, y_2) and (x_3, y_3) belong to the graph of $L(x)$ and Λ_3 is 3-secret shareable.

Let $\Phi : \mathbb{Z}_k \to \Lambda$ be an one to one onto function from $\mathbb{Z}_k$ to Λ. Then it is called an encoding function and the ordered pair (Λ, Φ) is called a *k* bit *t*-secret shareable pair.

Example 3.1.3: For any $\epsilon_1\epsilon_2 \in \mathbb{Z}_2$, we define $\Phi_{\Lambda_3}(\epsilon_1\epsilon_2)$ to be the graph of the quadratic function $(x+2^{\epsilon_1})(x+3^{\epsilon_2})$. Then $(\Lambda_3, \Phi_{\Lambda_3})$ is a 2 bit 3 secret shareable pair.

We would like to give another example of secret shareable pair.

Example 3.1.4: Let Λ^3 be the set of all circles in $\mathbb{R}^2$. Then by the elementary Geometry theorem, three points which are not collinear determine a unique circle as in the below figure.

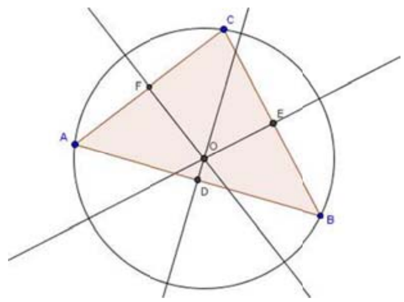

Figure 5: Three non-collinear points determine a unique circle.

Hence, Λ^3 is 3 secret shareable. Now we define a function Φ from $\mathbb{Z}_2$ to Λ^3 by

$$\Phi(00) = C_{00}$$
$$\Phi(01) = C_{01}$$
$$\Phi(10) = C_{10}$$
$$\Phi(11) = C_{11}$$

where C_{xy} is a circle such that its center is (x,y) and its radius is $1/3$. (Ref. Figure 6)

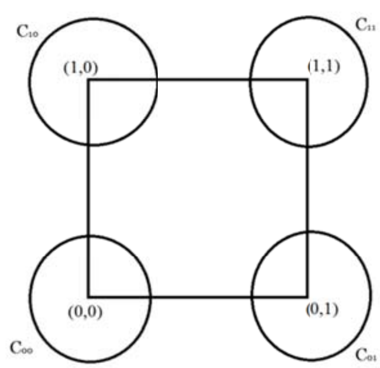

Figure 6: The function Φ maps two bits ε_1 ε_2 to a circle C $_{\varepsilon_1 \varepsilon_2}$.

Then (Λ^3, Φ) is a 2 bit 3 secret shareable pair as well.

3.2 k bit (t,n) Secret Sharing Distributed Storage framework

Assume that the ordered pair (Λ,Φ) is a k bit t-secret shareable pair and $n \geq t$ is an integer.

Definition 3.2.1: Let $\pi_1,\pi_2,\cdots,\pi_n$ be a sequence of functions from Λ to $\mathbb{R}^d$ such that for $j, j' = 1,2,\cdots,n$ and $C \in \Lambda$,

1. $\pi_j(C) \in C$;
2. $\pi_j(C) = \pi_{j'}(C)$ if and only if $j = j'$.

Then $\pi_1,\pi_2,\cdots,\pi_n$ is said to be a sequence of choice functions of Λ.

Example 3.2.2: Consider Λ_k and $C \in \Lambda$ such that C is a graph of a polynomial $P(x)$. We define

$$\pi_j(C) = (j, P(j))$$

for $j = 1,2,\cdots,n$.

Let $\ell = Mk$ for some positive integer M. Then we call s is a secret if $s = \epsilon_1\epsilon_2\cdots\epsilon_\ell \in \mathbb{Z}_\ell$. Therefore, s can also be written as

$$s = s_1 s_2 \cdots s_M$$

where $s_i = \epsilon_{(i-1)k+1}\epsilon_{(i-1)k+2}\cdots\epsilon_{ik}$ and $i = 1,2,\cdots,M$.

Note that it is sometime useful if we index s_i as below

$$s_i = \epsilon_1^i\epsilon_2^i\cdots\epsilon_k^i.$$

and so

$$s = \epsilon_1^1\epsilon_2^1\cdots\epsilon_k^1\epsilon_1^2\epsilon_2^2\cdots\epsilon_k^2\cdots\epsilon_1^i\epsilon_2^i\cdots\epsilon_k^i\cdots\epsilon_1^M\epsilon_2^M\cdots\epsilon_k^M.$$

3.2.1 Creating Containers from a Secret

Given a secret $s = s_1 s_2 \cdots s_M \in \mathbb{Z}_\ell$, a container array $[s]$ is an M by n array of points in $\mathbb{R}^d$ such that

$$[s]_{ij} = \pi_j(\Phi(s_i))$$

for $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, n$. The j th container $[s]_j$ of the secret s is the j column of the container array $[s]$ where $j = 1, 2, \cdots, n$.

Example 3.2.3: if $s = 100111$, then $s_1 = 10, s_2 = 01, s_3 = 11$. Then

$$\pi_j\left(\Phi\left(s_1\right)\right) = \pi_j\left(\Phi\left(10\right)\right) = \left(j, (j-2)(j-1)\right)$$
$$\pi_j\left(\Phi\left(s_2\right)\right) = \pi_j\left(\Phi\left(01\right)\right) = \left(j, (j-1)(j-3)\right)$$
$$\pi_j\left(\Phi\left(s_3\right)\right) = \pi_j\left(\Phi\left(11\right)\right) = \left(j, (j-2)(j-3)\right)$$

where $j = 1, 2, 3$. So the container array of the secret s is

$$[s] = \begin{bmatrix} (1,0) & (2,0) & (3,2) & (4,6) & (5,12) \\ (1,0) & (2,3) & (3,0) & (4,3) & (5,8) \\ (1,2) & (2,0) & (3,0) & (4,2) & (5,6) \end{bmatrix}.$$

Finally, we can obtain 5 containers

$$[s]_1 = \begin{bmatrix} (1,0) \\ (1,0) \\ (1,2) \end{bmatrix}, [s]_2 = \begin{bmatrix} (2,0) \\ (2,3) \\ (2,0) \end{bmatrix}, [s]_3 = \begin{bmatrix} (3,2) \\ (3,0) \\ (3,0) \end{bmatrix}, [s]_4 = \begin{bmatrix} (4,6) \\ (4,3) \\ (4,2) \end{bmatrix}, [s]_5 = \begin{bmatrix} (5,12) \\ (5,8) \\ (5,6) \end{bmatrix}.$$

Remark: The size of containers depends on the parameter k. However, it is not always the case that the bigger k, the smaller the size of containers would be theatrically.

3.2.2 Recover the Secret from t Containers

To recover the secret $s = \epsilon_1\epsilon_2 \cdots \epsilon_\ell = s_1 s_2 \cdots s_M$, we have to have at least t distinct containers. By the property c) of definition 3.3.1, without loss of generality, we can assume that they are $[s]_1, [s]_2, \cdots, [s]_t$. First of all, we form the M by t collector array

$$[s]_{12\cdots t} = \left[[s]_1, [s]_2, \cdots, [s]_t\right] = [p_j^i].$$

The i th row of the collector array consists of t distinct points $p_1^i, p_2^i, \cdots, p_t^i$ of $\Phi(s_i)$. By the properties a) and c) of Definition 3.1.1, we have

$$\Phi(s_i) = \Psi(p_1^i, p_2^i, \cdots, p_t^i)$$

and

$$s_i = \Phi^{-1}\left(\Psi(p_1^i, p_2^i, \cdots, p_t^i)\right)$$

for $i = 1, 2, \cdots, M$.

Finally, we recover the secret

$$\begin{aligned} s &= s_1 s_2 \cdots s_M \\ &= \Phi^{-1}(\Psi(p_1^1, p_2^1, \cdots, p_t^1))\Phi^{-1}(\Psi(p_1^2, p_2^2, \cdots, p_t^2))\cdots\Phi^{-1}(\Psi(p_1^n, p_2^n, \cdots, p_t^n)). \end{aligned}$$

Moreover, from property b) of Definition 3.1.1, the number of distinct containers needed to recover the secret s should be at least t .

3.2.3 Algorithms of *k* bit (*t*,*n*) Secret Sharing Distributed Storage

Assume a (Λ, Φ) is a *k* bit *t*-secret shareable pair, $n \geq t$ is an integer and $\pi_1, \pi_2, \cdots, \pi_n$ is a sequence of choice functions of Λ .

Algorithm for Distributed Storage

Step 1. Input a secret $s = s_1 s_2 \cdots s_M \in \mathbb{Z}_\ell$.

// $\ell = Mk$ //

Step 2. Form the container array $[s]$ by

$$[s]_{ij} = \pi_j(\Phi(s_i))$$

where $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, n$.

Step 3. Store the i th container $[s]_i$ into the i th storage.

Step 4. End

Algorithm for (*t*,*n*) Secret Sharing Recovery

Step 1. Get t containers $[s]_{j_1}, [s]_{j_2}, \cdots, [s]_{j_t}$ from t distinct storages

Step 2 Form the collector array $\left[s\right]_{j_1 j_2 \cdots j_t} = \left[[s]_{j_1}, [s]_{j_2}, \cdots, [s]_{j_t}\right]$.

Step 3. For $i = 1, 2, \cdots, M$, recover s_i from the i th row of the collector array as in section 3.2.2

Step 4 Output $s = s_1 s_2 \cdots s_M$.

Step 5 END

3.3 Permutations and encryptions

Definition 3.3.1: Let σ be an one to one onto function from $\{1, 2, 3, \cdots, n\}$. Then we call the function σ a permutation. The set of all the permutations on $\{1, 2, 3, \cdots, n\}$ is denoted by S_n. Let id be a permutation in S_n such that id maps every element of $\{1, 2, 3, \cdots, n\}$ to itself. For any $\sigma \in S_n$, we define $\sigma^0 = id$ and $\sigma^j = \sigma^{j-1} \circ \sigma$.

Given a permutation $\sigma \in S_n$ and $M \times n$ container array $[s]$ of a secret s, we define σ encrypted container array $[s]^\sigma$ to be $M \times n$ array of points in $\mathbb{R}^d$ such that

$$[s]^\sigma_{ij} = [s]_{i\sigma(j)}$$

for $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, n$. The i th column of $[s]^\sigma$ is said to be the i th σ encrypted container denoted by $[s]^\sigma_i$.

Let $2 \le m \le n$. A permutation σ' in S_n, denoted by $(n_1, n_2, \cdots, n_m)$, is called a cycle if there exist $n_1, n_2, \cdots, n_m$ are distinct numbers in $\{1, 2, 3, \cdots, n\}$ such that

$$\sigma'(n_r) = n_{\ \bmod_m (r+1)}$$

for $r = 1, 2, \cdots, m$ and σ' maps other element to itself. The number m is the length of σ' denoted by $|\sigma'|$.

Example 3.3.2: Let $\sigma' = (2,5,1) \in S_5$. Then $\sigma'(2) = 5, \sigma'(5) = 1, \sigma'(1) = 2, \sigma'(3) = 3$ and $\sigma'(4) = 4$. The length of $(2,5,1)$ is now equal to 3.

The period of a permutation σ is the smallest positive integer T such that $\sigma^T = \sigma$. Obviously, the period of a circle $\sigma' = (n_1, n_2, \cdots, n_m)$ is its length $\sigma' = m$. Since a permutation σ can be factorized uniquely into a product of cycles, we have the period of σ is the L. C. M. of the lengths of the cycles in the product. It is also known as Ruffini

Theorem (1799) [1]. We would like to use a permutation with biggest period for encryption for highest complexity of containers.

Example 3.3.3: Consider S_{10}. Since $10 = 5+3+2$, the pattern of permutations in S_{10} with maximum period $30 = 5\times 3\times 2$ is $(*****)(***)(**)$. There are 120960 of them in total and and we list some of them below:

(0, 8, 6, 7, 2)(1, 4, 9)(3, 5), (2, 4, 7, 6, 9)(0, 8, 5)(1, 3), (0, 3, 7, 9, 2)(4, 5, 8)(1, 6)

(1, 6, 9, 4, 5)(0, 2, 8)(3, 7), (0, 1, 8, 9, 4)(2, 5, 6)(3, 7), (0, 2, 8, 4, 9)(1, 6, 5)(3, 7)

(0, 8, 9, 5, 4)(1, 6, 7)(2, 3), (0, 2, 7, 8, 9)(1, 4, 3)(5, 6), (0, 9, 5, 4, 7)(1, 8, 3)(2, 6)

(0, 2, 9, 5, 3)(1, 8, 4)(6, 7).

Example 3.3.4: Let

$$[s]=\begin{bmatrix}(1,0) & (2,0) & (3,2) & (4,6) & (5,12)\\ (1,0) & (2,3) & (3,0) & (4,3) & (5,8)\\ (1,2) & (2,0) & (3,0) & (4,2) & (5,6)\end{bmatrix}$$

and $\sigma=(1,2,3)(4,5)$. Then $\sigma^2=(1,3,2)(5,4)$ and $\sigma^3=(4,5)$. Hence,

$$[s]^\sigma=\begin{bmatrix}(3,2) & (1,0) & (2,0) & (5,12) & (4,6)\\ (2,3) & (3,0) & (1,0) & (4,3) & (5,8)\\ (1,2) & (2,0) & (3,0) & (5,6) & (4,2)\end{bmatrix}$$

and the σ encrypted containers are

$$[s]_1^\sigma=\begin{bmatrix}(3,2)\\(2,3)\\(1,2)\end{bmatrix},[s]_2^\sigma=\begin{bmatrix}(1,0)\\(3,0)\\(2,0)\end{bmatrix},[s]_3^\sigma=\begin{bmatrix}(2,0)\\(1,0)\\(3,0)\end{bmatrix},[s]_4^\sigma=\begin{bmatrix}(5,12)\\(4,3)\\(5,6)\end{bmatrix},[s]_5^\sigma=\begin{bmatrix}(4,6)\\(5,8)\\(4,2)\end{bmatrix}.$$

Assume a (Λ,Φ) is a *k* bit *t*-secret shareable pair, $n\ge t$ is an integer, $\sigma\in S_n$ and $\pi_1,\pi_2,\cdots,\pi_n$ is a sequence of choice functions of Λ.

Algorithm for Encrypted Distributed Storage

Step 1. Input a secret $s=s_1s_2\cdots s_M\in\mathbb{Z}_\ell$.

// $\ell = Mk$ //

Step 2. Form the container array $[s]$ by

$$[s]_{ij} = \pi_j(\Phi(s_i))$$

where $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, n$.

Step 3. Obtain the σ encrypted container array $[s]^{\sigma}$ from $[s]$.

Step 4. Store the i th encrypted container $[s]_i$ into the i th storage.

Step 5. End

Note that the property c) of definition 3.1.1 is true regardless the order of the points $p_1, p_2, \cdots, p_t$. Therefore, it is no need to modify the recovery algorithm in subsection 3.2.3 and it also works well with encrypted containers.

3.4 Features of *k* bit (*t,n*) Secret Sharing Distributed Storage

In this section, we will mention some theoretical features of the framework of *k* bit (*t,n*) Secret Sharing Distributed Storage.

Recall that in Chapter 1, we mention that distributed storages will be under three major kinds of attacks:

-The attack on confidentiality reveals stored server contents to attackers;

-The attack on integrity modifies data in victim storage servers without being noticed;

-The attack on reliability makes storage server unavailable to legitimate users.

The propose framework will provide the following protections.

a) Ensuring confidentiality

The original secret is a binary string. However, a container is a column of points in $\mathbb{R}^d$ which carries only partial information of the secret. Even attackers successfully obtain less than t containers. They are not able to extract any information of the secret.

b) Ensuring integrity

Before the discussion, we would like to mention first that practically t will not be a large number and it is very likely less than 5 and n is much larger then t. Now assume that the storage system is under attack and the content of container is modified without notification and authorization. We are able to detect such modification and correct it when a defected container is used for recovery. For example, we have t containers of a secret s, namely $[s]_1$, $[s]_2$, $\cdots$, $[s]_t$ such that the first entry of the first container has been modified illegally. When we recover s from $[s]_1,[s]_2$, $\cdots$, $[s]_t$, the first row of collector array will determine an element in Λ which is not in the range of Φ. It is because the size of Λ is very much larger than the size of the range of Φ. Besides, in practice, we would like to make the elements of the range of Φ "away from each other". So we are able to know that the first row of $[s]_1,[s]_2$, $\cdots$, $[s]_t$ has been modified illegally. To identify the location and correct the illegal modification, we need another t good containers. Check the containers one by one with $t-1$ good containers. Then $[s]_1$ is the bad container. So we conclude that the first entry of the first container has been modified illegally. To correct this, firstly we use the first row of the t good containers to recover s_1 and then by using the choice function, we can recover the bad entry on $[s]_1$ (i.e. the first entry in this case.).

c) Ensuring reliability

Assume attackers have damaged a container. Administrators have no problem to restore the impaired container. First of all, we obtain t containers from non-compromised storages. Secondly, we recover the original secret s from those containers and finally, we can utilize the secret s to generate the damaged container again. Furthermore, we should note that getting t good containers is possible since n is much larger than t. For example, we choose $n=10$ and $t=3$ in this project. Therefore, we have $\frac{9!}{6!3!}=84$ many of combinations of three good containers ready for recovering the damaged container. Besides, with big n, the framework provides user with more access availability for t containers and hence the secret. Even, because of the fact that it is (t,n) secret sharing, it still maintains the good confidentiality while the number of containers or storages increases.

Reference

1. Gallian, J. *"Contemporary Abstract Algebra."* Brooks Cole, 6 edition, 2004.

4 Algebraic Methods

Let Λ_t be the set of the graphs of all the polynomials of degree $t-1$. By the elementary algebra result [1]:

Assume that functions f and g are polynomial functions of degree $t-1$ and they agree with each other on t distinct points, then $f=g$ and their coefficients are also equal.

Therefore, Λ_t satisfies properties a) and b) of definition of 3.1.1. To show that Λ_t also satisfies property c) of the definition, we need Lagrange polynomials.

4.1 Lagrange Polynomials.

Given (x_1, y_1), $(x_2, y_2), \cdots, (x_t, y_t)$ such that their x coordinates are distinct, let

$$L_j(x) = \frac{\prod_{\substack{i=1 \\ i\neq j}}^{t} (x - x_i)}{\prod_{\substack{i=1 \\ i\neq j}}^{t} (x_j - x_i)}$$

where $j = 1, 2, 3, \cdots$. So $L_j(x)$ is a polynomial of degree $t-1$ such that

$$L_j(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

where $i, j = 1, 2, 3, \cdots, t$ and they are called Lagrange basis functions. Hence, the Lagrange polynomial [2]

$$L(x) = y_1 L_1(x) + y_2 L_2(x) + \cdots + y_3 L_3(x)$$

will pass through all the given points.

Note that the y intercept of $L(x)$, $L(0)$, can be evaluated by

$$y_1 L_1(0) + y_2 L_2(0) + \cdots + y_3 L_3(0)$$

where

$$L_j(0)=\frac{\prod_{\substack{i=1\\ i\neq j}}^{t}(0-x_i)}{\prod_{\substack{i=1\\ i\neq j}}^{t}(x_j-x_i)}=\frac{(-1)^{t-1}\prod_{\substack{i=1\\ i\neq j}}^{t}x_i}{\prod_{\substack{i=1\\ i\neq j}}^{t}(x_j-x_i)}$$

for $j=1,2,3,\cdots$.

Therefore, Λ_t is t secret shareable if Ψ maps (x_1,y_1), $(x_2,y_2),\cdots$,(x_t,y_t) to the graph of the Lagrange polynomial $y_1L_1(x)+y_2L_2(x)+\cdots+y_3L_3(x)$.

4.2 Unique Factorization Property (UFP)

Let $A=\{m_1,m_2,\cdots,m_k\}$ be a set of positive integers bigger than 1 and we always assume that they are in increasing order i.e. $m_1<m_2<\cdots<m_k$. The set A (or the finite increasing sequence $m_1,m_2,\cdots,m_k$) has unique factorization property (UFP) if for any binary strings $\epsilon_1\epsilon_2\cdots\epsilon_k$ and $\overline{\epsilon_1}\overline{\epsilon_2}\cdots\overline{\epsilon_k}$,

$$m_1^{\epsilon_1}m_2^{\epsilon_2}\cdots m_k^{\epsilon_k}=m_1^{\overline{\epsilon_1}}m_2^{\overline{\epsilon_2}}\cdots m_k^{\overline{\epsilon_k}},$$

implies $\epsilon_i=\overline{\epsilon_i}$ for all $i=1,2,\cdots,k$.

Also, the span of an UFP set A denoted by $\overline{A}$ is defined to be

$$\overline{A}=\{m:m=m_1^{\epsilon_1}m_2^{\epsilon_2}\cdots m_k^{\epsilon_k}\text{ for some binary string }\epsilon_1\epsilon_2\cdots\epsilon_k\}.$$

Suppose now $A=\{1\le m_1\le m_2\le\cdots\le m_k\le\cdots\}$ is an infinite set of integers. For any $k=1,2,3,\cdots$, a k th segment A_k of A is defined to be the set of the first k elements of A. We call the set A has UFP if for any $k=1,2,3,\cdots$, A_k has UFP. The span of A, denoted by $\overline{A}$, is the union of the span of all the segments of A.

Example 4.2.1: The sets $\{2,3,p\}$ and $\{2,3,4\}$ are of UFP where p is a prime which is larger than 3.

Lemma 4.2.2: Let k be a positive integer. For any integer $0\le m<2^k$, there is a unique binary string $\epsilon_1\epsilon_2\cdots\epsilon_k$ such that

$$m=\epsilon_1 1+\epsilon_2 2+\cdots\epsilon_k 2^{k-1}.$$

Conversely, for any binary string $\epsilon_1\epsilon_2\cdots\epsilon_k$,we have $0\le\epsilon_1 1+\epsilon_2 2+\cdots\epsilon_k 2^{k-1}<2^k$.

Example 4.2.3: Let p be a prime number and let k be a positive integer. Define

$$X_k(p) = \{p^1, p^2, p^{2^2}, p^{2^3}, \cdots, p^{2^{k-1}}\}.$$

Assume that

$$p^{\epsilon_1 1} p^{\epsilon_2 2} p^{\epsilon_3 2^2} p^{\epsilon_4 2^3} \cdots p^{\epsilon_k 2^{k-1}} = p^{\overline{\epsilon}_1 1} p^{\overline{\epsilon}_2 2} p^{\overline{\epsilon}_3 2^2} p^{\overline{\epsilon}_4 2^3} \cdots p^{\overline{\epsilon}_k 2^{k-1}}.$$

So

$$p^{\epsilon_1 1 + \epsilon_2 2 + \epsilon_3 2^2 + \epsilon_4 2^3 + \cdots + \epsilon_k 2^{k-1}} = p^{\overline{\epsilon}_1 1 + \overline{\epsilon}_2 2 + \overline{\epsilon}_3 2^2 + \overline{\epsilon}_4 2^3 + \cdots + \overline{\epsilon}_k 2^{k-1}}.$$

By the Lemma 4.2.2, we have $\epsilon_i = \overline{\epsilon}_i$ for all $i = 1, 2, \cdots, k$. So $X_k(p)$ has UFP.

From the lemma again, the span of $X_k(p)$ is the set $\{p^0, p^1, p^2, \cdots, p^{2^k-2}, p^{2^k-1}\}$.
Therefore, $X(p) = \{p^1, p^2, p^{2^2}, p^{2^3}, \cdots, p^{2^{k-1}}, p^{2^k}, \cdots\}$ has UFP and its span is $\{p^0, p^1, p^2, p^3, \cdots\}$.

Example 4.2.4: Let $X = \bigcup \{X(p) : p \text{ is a prime}\}$. We arrange and index the elements of A in increasing order and

$$X = \{m_1, m_2, m_3, \cdots\} = \{2, 3, 4, 5, 7, 9, 11, 13, 16, 17, 19, 23, \cdots\}$$

from now on.

Let $X_k = \{m_1, m_2, m_3 \cdots, m_k\}$ be the set of first k elements of X where $k = 1, 2, 3, \cdots$. We claim that X_k has UFP. Assume that X_k consist of the powers of primes $p_1, p_2, \cdots, p_r$. Hence, X_k can be written as below:

$$X_k = \{p_1, p_1^2, \cdots, p_1^{2^{k_1-1}}, p_2, p_2^2, \cdots, p_2^{2^{k_2-1}}, \cdots, p_r, p_r^2, \cdots, p_r^{2^{k_r-1}}\}.$$

where $k_1 + k_2 + \cdots + k_r = k$. Let $\epsilon_1^1 \cdots \epsilon_{k_1}^1 \epsilon_1^2 \cdots \epsilon_{k_2}^2 \cdots \epsilon_1^r \cdots \epsilon_{k_r}^r$ and $\overline{\epsilon}_1^1 \cdots \overline{\epsilon}_{k_1}^1 \overline{\epsilon}_1^2 \cdots \overline{\epsilon}_{k_2}^2 \cdots \overline{\epsilon}_1^r \cdots \overline{\epsilon}_{k_r}^r$ such that

$$\begin{aligned} & p_1^{\epsilon_1^1 1} p_1^{\epsilon_2^1 2} \cdots p_1^{\epsilon_{k_1}^1 2^{k_1-1}} p_2^{\epsilon_1^2 1} p_2^{\epsilon_2^2 2} \cdots p_2^{\epsilon_{k_2}^2 2^{k_2-1}} \cdots p_r^{\epsilon_1^r 1} p_r^{\epsilon_2^r 2} \cdots p_r^{\epsilon_{k_r}^r 2^{k_r-1}} \\ = {} & p_1^{\overline{\epsilon}_1^1 1} p_1^{\overline{\epsilon}_2^1 2} \cdots p_1^{\overline{\epsilon}_{k_1}^1 2^{k_1-1}} p_2^{\overline{\epsilon}_1^2 1} p_2^{\overline{\epsilon}_2^2 2} \cdots p_2^{\overline{\epsilon}_{k_2}^2 2^{k_2-1}} \cdots p_r^{\overline{\epsilon}_1^r 1} p_r^{\overline{\epsilon}_2^r 2} \cdots p_r^{\overline{\epsilon}_{k_r}^r 2^{k_r-1}}. \end{aligned}$$

So

$$p_1^{\epsilon_1^1 1 + \epsilon_2^1 2 + \cdots + \epsilon_{k_1}^1 2^{k_1-1}} p_2^{\epsilon_1^2 1 + \epsilon_2^2 2 + \cdots + \epsilon_{k_2}^2 2^{k_2-1}} \cdots p_r^{\epsilon_1^r 1 + \epsilon_2^r 2 + \cdots + \epsilon_{k_r}^r 2^{k_r-1}}$$

$$= p_1^{\overline{\epsilon}_1^1 1+\overline{\epsilon}_2^1 2+\cdots+\overline{\epsilon}_{k_1}^1 2^{k_1-1}} p_2^{\overline{\epsilon}_1^2 1+\overline{\epsilon}_2^2 2+\cdots+\overline{\epsilon}_{k_2}^2 2^{k_2-1}} \cdots p_r^{\overline{\epsilon}_1^r 1+\overline{\epsilon}_2^r 2+\cdots+\overline{\epsilon}_{k_r}^r 2^{k_r-1}} .$$

Since such factorization is unique and $X_{k_i}(p_i)$ has UFP for all $i=1,2,\cdots,r$, we conclude that

$$\epsilon_j^i = \overline{\epsilon}_j^i$$

for all i, j and hence X_k has UFP and hence X has UFP.

Theorem 4.2.5 : Let X_k be the same as the one in above example and let m be a positive integer less than m_k. Then m is in the span of X_k.

Proof: Assume that $m = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$. Since $m < x_k$, we have $p_1^{\alpha_1}, p_2^{\alpha_2}, \cdots, p_r^{\alpha_r} < m_k$ and hence $p_1, p_2, \cdots, p_r \in X_k$. For any $i = 1,2,3,\cdots,r$, there is a positive integer k_i such that

$$p_i^{2^{k_i-1}} < x_k \leq p_i^{2^{k_i}} .$$

Since $X_{k_i}(p_i) \subset X_k$ and $p_i^{\alpha_i} < p_i^{2^{k_i}}$ is in its span, $p_i^{\alpha_i} = p_i^{\epsilon_1^i 1} p_i^{\epsilon_2^i 2} p_i^{\epsilon_3^i 2^2}, p_i^{\epsilon_4^i 2^3} \cdots, p_i^{\epsilon_{k_i}^i 2^{k_i-1}}$ for some binary string $\epsilon_1^i \epsilon_2^i \cdots \epsilon_{k_i}^i$. Therefore, m is in the span of X_k.

Let k be a positive integer and let A be an UFP set with $|A| = k$. Define ΠA to be the product of all its elements. A is minimum if for any UFP set A' with $|A'| = k$, we have $\Pi A \leq \Pi A'$. A is completely minimum if every segment of A has minimum UFP.

Example 4.2.6: The first eight minimum UFP sets are: $X_1 = \{2\}$, $X_2 = \{2,3\}$, $X_3 = \{2,3,4\}$, $X_4 = \{2,3,4,5\}$, $X_5 = \{2,3,4,5,7\}$, $X_6 = \{2,3,4,5,7,9\}$, $X_7 = \{2,3,4,5,7,9,11\}$ and $X_8 = \{2,3,4,5,7,9,11,13\}$. Also, $\Pi X_1 = 2$, $\Pi X_2 = 6$, $\Pi X_3 = 24$ and $\Pi X_4 = 120$.

Note that

X_4 is useful in this project since it has minimum UFP and $\Pi X_4 = 120$ is smaller than 127, the absolute limit of a character type variables.

We conclude the section by two interesting conjectures: For, $k = 1,2,3,\cdots$,

Conjecture 1) X_k is the unique set which has completely minimum UFP ;

Conjecture 2) X_k is the unique set which has minimum UFP.

4.3 Proposed Encoding Function Φ_{Λ_t}

Recall that Λ_t is the set of the graphs of all the polynomials of degree $t-1$. For k bits $\epsilon_1\epsilon_2\cdots\epsilon_k$ and $k_1,k_2,\cdots,k_{t-1}\geq 1$ such that $k_1+k_2+\cdots+k_{t-1}=k$,

$\Phi_{\Lambda_t}(\epsilon_1\epsilon_2\cdots\epsilon_k)$ is the graph of

$$(x-m_1^{\epsilon_1}m_2^{\epsilon_2}\cdots m_{k_1}^{\epsilon_{k_1}})(x-m_{k_1+1}^{\epsilon_{k_1+1}}m_{k_1+2}^{\epsilon_{k_1+2}}\cdots m_{k_2}^{\epsilon_{k_2}})\cdots(x-m_{k_{t-1}+1}^{\epsilon_{k_{t-1}+1}}m_{k_{t-1}+2}^{\epsilon_{k_{t-1}+2}}\cdots m_k^{\epsilon_k}).$$

Since X_k is of UFP, we have the function Φ_{Λ_t} is one to one. Hence, $\left(\Lambda_t,\Phi_{\Lambda_t}\right)$ is k bit t secret sharing pair.

Example 4.3.1: Let $X_4=\{2,3,4,5\}$ and $k_1=k_2=2$. Therefore, $k=4$. Let Λ_3 be the set of the graphs of all the quadratic polynomials. For any 4 bits $\epsilon_1\epsilon_2\epsilon_3\epsilon_4$, Φ maps $\epsilon_1\epsilon_2\epsilon_3\epsilon_4$ to the graph of the quadratic polynomial $(x-2^{\epsilon_1}3^{\epsilon_2})(x-4^{\epsilon_3}5^{\epsilon_4})$. For convenience, we would like to identify the graph of $\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)$ with the polynomial $\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)$ in this case.

So the y intercepts of $\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)$, $\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)(0)$,can be summarized in the table below:

$\epsilon_1\epsilon_2\epsilon_3\epsilon_4$	$\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)$	$\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)(0)$	$\epsilon_1\epsilon_2\epsilon_3\epsilon_4$	$\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)$	$\Phi(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)(0)$
0000	$(x-2^0 3^0)(x-4^0 5^0)$	1	0110	$(x-2^0 3^1)(x-4^1 5^0)$	12
1000	$(x-2^1 3^0)(x-4^0 5^0)$	2	0101	$(x-2^0 3^1)(x-4^0 5^1)$	15
0100	$(x-2^0 3^1)(x-4^0 5^0)$	3	0011	$(x-2^0 3^0)(x-4^1 5^1)$	20
0010	$(x-2^0 3^0)(x-4^1 5^0)$	4	1110	$(x-2^1 3^1)(x-4^1 5^0)$	24
0001	$(x-2^0 3^0)(x-4^0 5^1)$	5	1101	$(x-2^1 3^1)(x-4^0 5^1)$	30
1100	$(x-2^1 3^1)(x-4^0 5^0)$	6	1011	$(x-2^1 3^0)(x-4^1 5^1)$	40
1010	$(x-2^1 3^0)(x-4^1 5^0)$	8	0111	$(x-2^0 3^1)(x-4^1 5^1)$	60
1001	$(x-2^1 3^0)(x-4^0 5^1)$	10	1111	$(x-2^1 3^1)(x-4^1 5^1)$	120

Then (Λ_3,Φ) is a 4 bit 3-secret sharing pair.

Remark: We would like to choose a encoding pair in order to making the size of containers to be as small as possible. In the example 4.3.1, the size of resulted containers is double of the size of origin secret.

4.4 Choice functions of Λ and the Calculation of $\Phi_{\Lambda_t}^{-1}(\Psi)$

Let $\left(\Lambda_t,\Phi_{\Lambda_t}\right)$ be the k bit t secret sharing pair as above and $n\geq t$. For any $C\in\Lambda$ such that C is a graph of a polynomial $P(x)$ in Λ_t. We define

$$\pi_j(C) = (j, P(j))$$

for $j = 1, 2, \cdots, n$.

Given (x_1, y_1), $(x_2, y_2), \cdots ,(x_t, y_t)$, we are going to find

$$\Phi_{\Lambda_t}^{-1}\left(\Psi\left((x_1, y_1), (x_2, y_2), \cdots, (x_t, y_t)\right)\right).$$

First of all, we build up a look up table which show the one to one correspondence between k bits onto y intercept of polynomial in the range of Φ. By using Lagrange polynomials, we can find a polynomial of degree $t-1$, $L(x)$ passing through all the given points. Find the y intercept of $L(x)$ by evaluating $L(0)$. Then we can find $\epsilon_1\epsilon_2\cdots\epsilon_k$ such that

$$\Phi_{\Lambda_t}\left(\epsilon_1\epsilon_2\cdots\epsilon_k\right) = \Psi\left((x_1, y_1), (x_2, y_2), \cdots, (x_t, y_t)\right).$$

by looking up from a table which gives the correspondence between $\epsilon_1\epsilon_2\cdots\epsilon_k$ and $\Phi_{\Lambda_t}(\epsilon_1\epsilon_2\cdots\epsilon_k)(0)$.

Exmaple 4.4.1: Look up table for $(\Lambda_2, \Psi_{\Lambda_2})$ is

$\epsilon_1\epsilon_2\epsilon_3\epsilon_4$	$\Phi_{\Lambda_2}(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)(0)$	$\epsilon_1\epsilon_2\epsilon_3\epsilon_4$	$\Phi_{\Lambda_2}(\epsilon_1\epsilon_2\epsilon_3\epsilon_4)(0)$
0000	1	0110	12
1000	2	0101	15
0100	3	0011	20
0010	4	1110	24
0001	5	1101	30
1100	6	1011	40
1010	8	0111	60
1001	10	1111	120

4.5 Robustness Analysis of Lagrange Polynomial when *t=3*

Let $t = 3$. Given $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3), the y intercept of the Lagrange polynomial passing given points is given by

$$L(0) = y_1 \frac{x_3 x_2}{(x_1 - x_3)(x_1 - x_2)} + y_2 \frac{x_1 x_3}{(x_2 - x_1)(x_2 - c_3)} + y_3 \frac{x_1 x_2}{(x_3 - x_1)(x_3 - x_2)}.$$

We assume that noise presents on x_1, x_2, x_3 which is bounded by ϵ and the noise presents on y_1, y_2, y_3 which is bounded by δ. Also, x_1, x_2, x_3 is bounded by M and the distance between any pair of them is bigger than λ. We wish to found an error bound of the calculation of noisy $L(0), \bar{L}(0)$.

Observing that

$$\begin{aligned}
&\left|\frac{1}{x_2 - x_1} - \frac{1}{x_2 + \varepsilon_2 - x_1 - \varepsilon_1}\right| \\
&= \left|\frac{(x_2 - x_1 + \varepsilon_2 - \varepsilon_1) - (x_2 - x_1)}{(x_2 - x_1)(x_2 - x_1 + \varepsilon_2 - \varepsilon_1)}\right| \\
&= \left|\frac{\varepsilon_2 - \varepsilon_1}{(x_2 - x_1)(x_2 - x_1 + \varepsilon_2 - \varepsilon_1)}\right|
\end{aligned}$$

where $\varepsilon_1, \varepsilon_2$ are noises associated with x_1, x_2 respectively. If $|x_1 - x_2| \geq \lambda$, then

$$|x_2 + \varepsilon_2 - x_1 - \varepsilon_1| > \lambda - 2\varepsilon.$$

Therefore, we have

$$\left|\frac{1}{x_2 - x_1} - \frac{1}{x_2 + \varepsilon_2 - x_1 - \varepsilon_1}\right| \leq \frac{2\varepsilon}{\lambda(\lambda - 2\varepsilon)}.$$

Observe that

$$\begin{aligned}
&\left|\frac{x_3 x_2}{(x_3 - x_1)(x_2 - x_1)} - \frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)(x_2 + \varepsilon_2 - x_1 - \varepsilon_1)}\right| \\
&\leq \left|\frac{x_3 x_2}{(x_3 - x_1)(x_2 - x_1)} - \frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 - x_1)(x_2 - x_1)}\right| \\
&+ \left|\frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 - x_1)(x_2 - x_1)} - \frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)(x_2 - x_1)}\right| \\
&+ \left|\frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)(x_2 - x_1)} - \frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)(x_2 + \varepsilon_2 - x_1 - \varepsilon_1)}\right| \\
&= \frac{1}{|(x_3 - x_1)(x_2 - x_1)|}|x_2\varepsilon_3 + x_3\varepsilon_2 + \varepsilon_3\varepsilon_2| \\
&+ \left|\frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_2 - x_1)}\right|\left|\frac{1}{(x_3 - x_1)} - \frac{1}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)}\right| \\
&+ \left|\frac{(x_3 + \varepsilon_3)(x_2 + \varepsilon_2)}{(x_3 + \varepsilon_3 - x_1 - \varepsilon_1)}\right|\left|\frac{1}{(x_2 - x_1)} - \frac{1}{(x_2 + \varepsilon_2 - x_1 - \varepsilon_1)}\right|
\end{aligned}$$

Therefore,

$$\left|\frac{x_3x_2}{(x_3-x_1)(x_2-x_1)}-\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}\right|$$

$$\le\frac{1}{\lambda^2}(2M+\varepsilon)\varepsilon+(M+\varepsilon)^2\frac{2\varepsilon}{\lambda^2(\lambda-2\varepsilon)}+(M+\varepsilon)^2\frac{2\varepsilon}{\lambda(\lambda-2\varepsilon)^2}$$

$$\le\frac{1}{\lambda}\left[2(\frac{M+\varepsilon}{\lambda-2\varepsilon})\varepsilon+\left(\frac{M+\varepsilon}{\lambda-2\varepsilon}\right)^2 2\varepsilon+\left(\frac{M+\varepsilon}{\lambda-2\varepsilon}\right)^2 2\varepsilon\right]$$

$$\le\frac{1}{\lambda}\left[2\omega+4\omega^2\right]\varepsilon$$

where $\omega=\dfrac{M+\varepsilon}{\lambda-2\varepsilon}$. So we have

$$\left|\frac{x_3x_2}{(x_3-x_1)(x_2-x_1)}y_1-\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}(y_1+\delta_1)\right|$$

$$\le\left|\frac{x_3x_2}{(x_3-x_1)(x_2-x_1)}y_1-\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}y_1\right|$$

$$+\left|\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}y_1-\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}(y_1+\delta_1)\right|$$

$$\le\left|\frac{x_3x_2}{(x_3-x_1)(x_2-x_1)}-\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}\right||y_1|$$

$$+\left|\frac{(x_3+\varepsilon_3)(x_2+\varepsilon_2)}{(x_3+\varepsilon_3-x_1-\varepsilon_1)(x_2+\varepsilon_2-x_1-\varepsilon_1)}\right||\delta_1|$$

$$\le\frac{1}{\lambda}\left(2\omega+4\omega^2\right)\varepsilon\overline{M}+\omega^2\delta.$$

where $\bar{M}$ is an upper bound of absolute values of all possible y.

Theorem 4.5.1: Given $(x_1,y_1),(x_2,y_2)$ and (x_3,y_3), noise presents on x_1,x_2,x_3 which is bounded by ϵ and on y_1,y_2,y_3 with is bounded by δ. Also, x_1,x_2,x_3 and y_1,y_2,y_3 are bounded by M and $\bar{M}$ respectively. Also suppose that the distance between any pair of them is bigger than λ. Let $\bar{L}(0)$ be a noisy evaluation of $L(0)$. Then

$$\left|L(0)-\bar{L}(0)\right|\le\frac{1}{\lambda}\left(6\omega+12\omega^2\right)\varepsilon\overline{M}+3\omega^2\delta$$

where $\omega=\dfrac{M+\varepsilon}{\lambda-2\varepsilon}$.

The following corollary gives a relatively simple inequality when $x_1, x_2, \cdots, x_n$ are equally spaced and noise is not present on them.

Corollary 4.5.2: If $\epsilon = 0$, $\lambda = \frac{M}{n}$ where n is the number of containers then $\omega = n$ and

$$\left|L(0) - \bar{L}(0)\right| \le 3n^2\delta .$$

We assume $n = 10$ in our study and employ the choice functions in 4.4. From corollary 4.5.2. we have

$$\left|L(0) - \bar{L}(0)\right| \le 300\delta .$$

Therefore, keep the noise bounded by a “not very small” number $\delta = 0.001$. Then $\left|L(0) - \bar{L}(0)\right| \le 1/3$. Note that from the look up table in Example 4.4.1, the smallest distance between those y intercept is 1 which is bigger than $1/3$. So the calculation of $L(0)$ is still stable in this case.

4.6 Implementation of the Algebra Method

All the programs developed in this report is in C and their interface is Qt from Qt Project [3]. They are all running under testing environment as described below:

OS: Windows 8
CPU: Inter Core i5-3337U(1.8.Ghz / Turbo:2.7Ghz)
RAM: 4GB DDR3
Harddisk: 128GB SSD

The testing group is randomly generated text files with different sizes 2, 4, 6, 8 and 10 MB (megabytes) and the compression is performed by 7-Zip. (http://www.7-zip.org/)

Let (Λ_2, Φ) be the 4 bit 3 secret sharing pair as in Example 4.3.1. Therefore,

Φ maps 4 bits $\epsilon_1\epsilon_2\epsilon_3\epsilon_4$ to the graph of a quadratic function

$$(x - 2^{\epsilon_1}3^{\epsilon_2})(x - 4^{\epsilon_3}5^{\epsilon_4}) .$$

The entries of containers can be stored in the character type variables for obtaining smaller size of containers. It can be done since the range of quadratic function

$(x-2^{e_1}3^{e_2})(x-4^{e_3}5^{e_4})$ is from -120 to 120 and it is contained in the range of character which is from -127 to 127. One character occupies 1 byte or 8 bits and the method takes 4 bits at a time. Therefore, the size of the containers is double of that of the original secret file.

4.6.1 Speed Tests

a) Producing Containers

The following table shows how much time needed for producing 10 uncompressed containers.

Secret Size (in MB)	2	4	6	8	10
Generating Time for 10 Containers (in Sec)	1.183	2.349	3.58	4.73	5.917

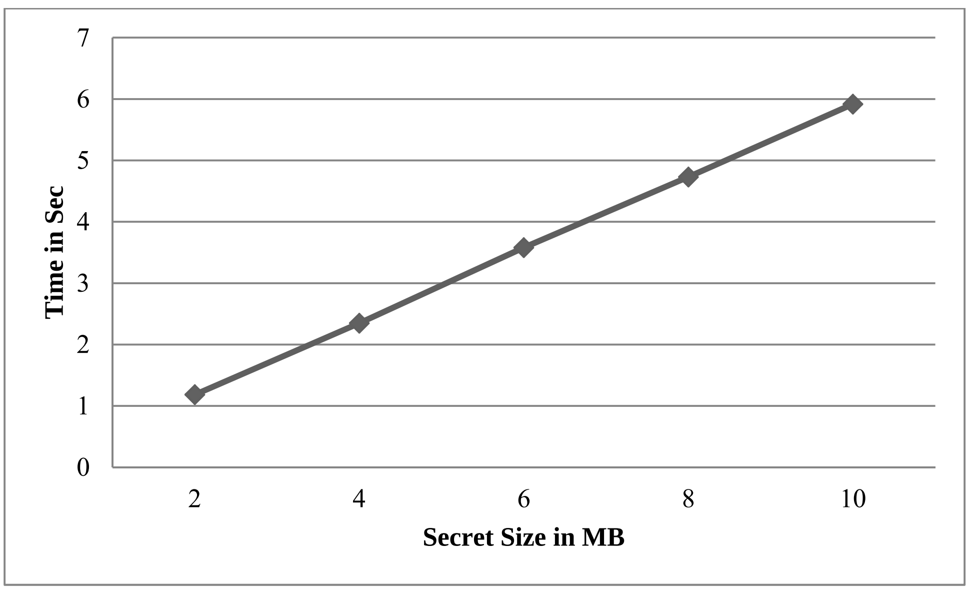

Figure 7: Time required for various sizes of secrets to generate 10 containers.

b) Recover the Original File

The following table shows how much time needed for recovering original files from their containers in a)

Secret Size (in MB)	2	4	6	8	10
Rcovert Time (in Sec)	0.237	0.471	0.708	0.935	1.176

Figure 8: Time required for various sizes of secrets to recover the secret from three containers.

4.6.2 Size Tests

The following table shows that average sizes of the compressed container files of the testing group.

Secret Size (in MB)	**2**	**4**	**6**	**8**	**10**
Compressed Secret Size (in Kb)	987	1973	2960	3947	4934
Compressed Container Size (in Kb)	984.5	1950.2	2919.5	3879.1	4678.9
Size Ratio of Compressed Secret and Compressed Container (in %)	100	99	99	98	95

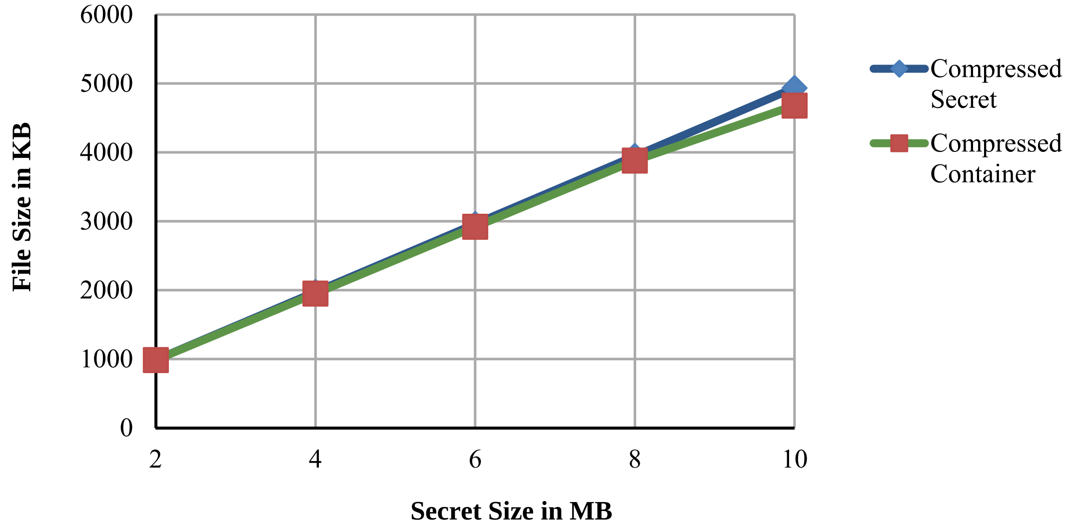

Figure 9: Comparison of Sizes of compressed secret

and compressed container for various sizes of secrets.

Reference

1. Fine, H. B. " *A College Algebra*." Ginn & company, 1904.
2. Hildebrand, F. B. *"Introduction to Numerical Analysis."* New York: McGraw-Hill, 1956.
3. Qt Project, http://qt-project.org/search/tag/qtgui

5. Geometric Methods

5.1 *3* Secret Shareable Set Λ^3

Let Λ^3 be the set of all circles on the plane. Recall that 3 non-collinear distinct points determine a circle. Since three points on a circle cannot be collinear in $\mathbb{R}^2$, we have Λ^3 satisfies a) and b). Indeed, it also satisfies property c). To see that, we consider the following calculation. [1]

Let (x_1, y_1), (x_2, y_2) and (x_3, y_3) lie on a circle

$$C: Ax^2 + By^2 + Cx + Dy + E \text{ or } C: (x - x_C)^2 + (y - y_C)^2 = r_C .$$

Consider the following determinant equation

$$J(x, y) = \begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 .$$

By evaluating the cofactors M_{1j} for the first row of the determinant, the determinant can be written as an equation of these cofactors:

$$M_{11}(x^2 + y^2) - M_{12}x + M_{13}y - M_{14} = 0$$

or

$$(x^2 + y^2) - \frac{M_{12}}{M_{11}}x + \frac{M_{13}}{M_{11}}y - \frac{M_{14}}{M_{11}} = 0$$

Since $J(x_1, y_1) = J(x_2, y_2) = J(x_3, y_3) = 0$, we have the above equation also represents the circle C. Extending $(x - x_C)^2 + (y - y_C)^2 = r_C$ into

$$x^2 + y^2 - 2x_C x - 2y_C y + \left(x_C^2 + y_C^2 - r_C^2\right)$$

and comparing the coefficients of

$$(x^2 + y^2) - \frac{M_{12}}{M_{11}}x + \frac{M_{13}}{M_{11}}y - \frac{M_{14}}{M_{11}} = 0,$$

we have

$$x_C = \frac{M_{12}}{2M_{11}}$$
$$y_C = \frac{M_{13}}{2M_{11}} \quad .$$
$$r_C = \sqrt{x_C^2 + y_C^2 + \frac{M_{14}}{M_{11}}}$$

Hence, Ψ maps (x_1, y_1), (x_2, y_2) and (x_3, y_3) to the circle with center $(x_C, y_C) = \left(\frac{M_{12}}{2M_{11}}, \frac{M_{13}}{2M_{11}}\right)$ and radius $\sqrt{x_C^2 + y_C^2 + \frac{M_{14}}{M_{11}}}$. Therefore, Λ^3 is 3 secret shareable.

5.2 Encoding Function Φ_{Λ^3}

Let k_1, k_2 and k_3 be positive integers such that $k = k_1 + k_2 + k_3$. Since a circle C can be determined by its center (x_C, y_C) and its radius r_C, we would like to define Φ_{Λ^3} to be a function that maps a k bit string $\epsilon_1\epsilon_2\cdots\epsilon_k$ to a circle C such that

$$(x_C, y_C) = (\epsilon_1 + \epsilon_2 2 + \cdots + \epsilon_{k_1} 2^{k_1}, \epsilon_{k_1+1} + \epsilon_{k_1+2} 2 + \cdots + \epsilon_{k_1+k_2} 2^{k_1+k_2})$$

and

$$r_C = 1 + \epsilon_{k_1+k_2+1} + \epsilon_{k_1+k_2+2} 2 + \cdots + \epsilon_k 2^k .$$

Then Φ_{Λ^3} is one to one since binary representation of integers is unique and hence, $(\Lambda^3, \Psi_{\Lambda^3})$ is a k bit 3 secret sharing pair.

5.3 Choice functions of Λ^3 and the Calculation of $\Phi_{\Lambda^3}^{-1}(\Psi)$

Let $n \geq 3$ and $0 < \theta \leq \frac{\pi}{2n}$. For any $C \in \Lambda^3$ such that C has center (x_C, y_C) and its radius r_C. We define

$$\pi_j(C) = (x_c + r_c \sin(j\theta), y_c + r_c \cos(j\theta))$$

for $j = 1, 2, \cdots, n$.

Given (x_1, y_1), (x_2, y_2) and (x_3, y_3), Ψ maps these points to the circle with center

$$(x_C, y_C) = \left(\frac{M_{12}}{2M_{11}}, \frac{M_{13}}{2M_{11}}\right)$$

and radius

$$r_C = \sqrt{x_C^2 + y_C^2 + \frac{M_{14}}{M_{11}}}\ .$$

Then the k bits $\epsilon_1\epsilon_2\cdots\epsilon_k$ such that

$$\Phi_{\Lambda^3}\left(\epsilon_1\epsilon_2\cdots\epsilon_k\right) = \Psi\left((x_1, y_1), (x_2, y_2), (x_3, y_3)\right)$$

is

$$\left(\frac{M_{12}}{2M_{11}}\right)_{\bmod 2}\left(\frac{M_{13}}{2M_{11}}\right)_{\bmod 2}\left(\sqrt{\left(\frac{M_{12}}{2M_{11}}\right)^2 + \left(\frac{M_{13}}{2M_{11}}\right)^2 + \frac{M_{14}}{M_{11}}} - 1\right)_{\bmod 2}.$$

A k bit $(3, n)$ secret sharing storage can be launched for the k bit 3 secret sharing pair $(\Lambda^3, \Phi_{\Lambda^3})$.

5.4 4 Secret Shareable Set Λ^4

In $\mathbb{R}^3$, if two spheres have four points in common (Ref. Figure 10), it does not implies that they are equal. It is because these four distinct points may be coplanar. (Ref. Figure 10)

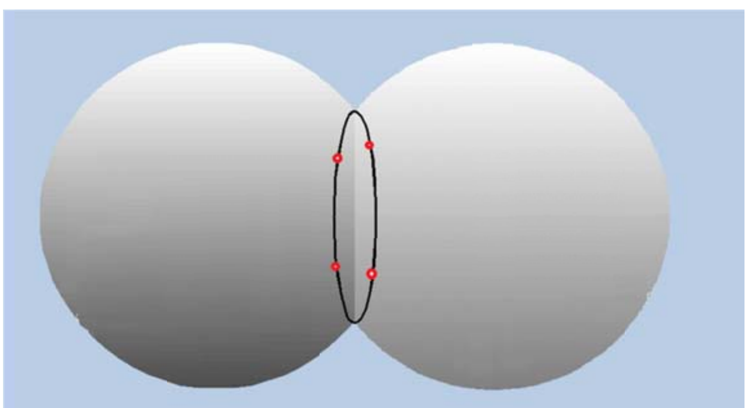

Figure 10: Two different spheres have four points in common

Therefore, the set of all the spheres in $\mathbb{R}^3$ is not 4 secret shareable. However, by Theorem 5.4.1, we can see that it is "almost" 4 secret shareable.

Theorem 5.4.1: The probability of picking n distinct points randomly and independently on the 3 dimensional sphere such that there are 4 point among them lying on the same plane is equal to zero.

Proof: We would like to prove it by induction.

Assume that we have chosen 3 points from the sphere. The area of the intersection of the sphere and the plane determined by chosen points is equal to zero. So the event, picking a point randomly such that it lies on the intersection, has zero probability.

Let A be the event such that n th point is chosen and there is 4 points among them are coplanar and B be the event such that $n-1$ point has been chosen and there is 4 points among them are coplanar. We wish to prove that $P(A)=0$ by using the formula of conditional probabilities

$$P(A)=P(A\,|\,B)P(B)+P(A\,|\,B^c)P(B^c)$$

where the event B^c is the negation of the event B. By the induction hypothesis $P(B)=0$, so $P(A)=P(A\,|\,B^c)P(B^c)$.

Now suppose that we have picked $n-1$ points and any 4 points among them are not coplanar. Similarly, the area of the intersection the sphere and the union of all planes determined by 4 points in the $n-1$ points is zero since the number of the planes is $\binom{n-1}{4}$ and the area of the intersection of the sphere and a plane is zero. Hence, $P(A\,|\,B^c)=0$. So we conclude that $P(A)=0$ and by the principle of Mathematical induction, the proof is complete.

According to Theorem 5.4.1, a non-painful way is that just let Λ be the set of all the spheres in $\mathbb{R}^3$ since the probability of making mistake is zero. However, how safe would it be probability 0 ?

If we would like to go for a safe approach, pick enough points from the unit sphere and form a set Ω. Note that $(x_1,y_1,z_1),(x_2,y_2,z_2)$ and (x_3,y_3,z_3) are coplanar if and only if

$$\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \neq 0.$$

Check all the combinations of four distinct points in Ω if they are coplanar. By the above Theorem, you are almost safe but it takes time. If not for all combinations, then stop and keep the set Ω. If yes for a combination, then choose n points and repeat the checking again unit we finally have a non coplanar set Ω of points on the sphere.

Let $r > 0$ and let v be a vector in $\mathbb{R}^3$. Define if D is in $\mathbb{R}^3$, the r dilation and v translation of D is

$$rD = \{\alpha p : p \in D\} \text{ and } v + D = \{p + v : p \in D\}.$$

Since coplanarity is invariant under dilation and then translation, we define

$$\Lambda^4 = \{v + r\Omega : r > 0 \text{ and } v \in \mathbb{R}^3\}.$$

Note that $v + r\Omega$ is a subset of the sphere with center v and radius r. Therefore, we can consider $v + r\Omega$ as a discrete model of the sphere such that any combination of four distinct points in $v + r\Omega$ are not coplanar. We would like to denote $v + r\Omega$ by $S(v, r)$.

Also, given four non-coplanar distinct points $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)$ and similarly consider the determinant

$$J(x, y, z) = \begin{vmatrix} x^2 + y^2 + z^2 & x & y & z & 1 \\ x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0,$$

$(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)$ should belong to $S((v_1, v_2, v_3), r)$ such that

$$v_1 = \frac{M_{12}}{2M_{11}}$$
$$v_2 = \frac{M_{13}}{2M_{11}}$$
$$v_3 = \frac{M_{14}}{2M_{11}}$$
$$r = \sqrt{v_1^2 + v_2^2 + v_3^2 + \frac{M_{15}}{M_{11}}}$$

where $M_{11}, M_{12}, M_{13}, M_{14}$ and M_{15} are cofactors corresponding to the first row of $J(x, y, z)$. Hence, Λ^4 is 4 secret sharing.

5.5 Encoding Function Φ_{Λ^4}

Let k be an integer bigger than or equal to 4 and let k_1, k_2, k_3 and k_4 be positive integers such that $k = k_1 + k_2 + k_3 + k_4$. Then the function Φ_{Λ^4} maps a k bit string $\epsilon_1\epsilon_2\cdots\epsilon_k$ to $S(v,r)$ such that

$$(v_1, v_2, v_3)^T = \begin{bmatrix} \epsilon_1 + \epsilon_2 2 + \cdots + \epsilon_{k_1} 2^{k_1} \\ \epsilon_{k_1+1} + \epsilon_{k_1+2} 2 + \cdots + \epsilon_{k_1+k_2} 2^{k_1+k_2} \\ \epsilon_{k_1+k_2+1} + \epsilon_{k_1+k_2+2} 2 + \cdots + \epsilon_{k_1+k_2+k_3} 2^{k_1+k_2+k_3} \end{bmatrix}$$

and

$$r = 1 + \epsilon_{k_1+k_2+k_3+1} + \epsilon_{k_1+k_2+k_3+2} 2 + \cdots + \epsilon_k 2^k .$$

Again, Φ_{Λ^4} is one to one since binary representation of integers is unique. Hence, $(\Lambda^4, \Phi_{\Lambda^4})$ is a k bit 4 secret sharing pair.

5.6 Choice functions of Λ^4 and the Calculation of $\Phi_{\Lambda^4}^{-1}(\Psi)$

Suppose that $\Omega = \{u_1, u_2, \cdots, u_l\}$ such that $l \ge n \ge 4$. For any $S(v,r) \in \Lambda^3$,we define

$$\pi_j(S(v,r)) = v + r\left(u_j\right)$$

for $j = 1, 2, \cdots, n$.

Similarly, $\Phi_{\Lambda^4}^{-1}(\Psi)$ maps a k bits $\epsilon_1\epsilon_2\cdots\epsilon_k$ to

$$\left(\frac{M_{12}}{2M_{11}}\right)_{\mathrm{mod}\,2}\left(\frac{M_{13}}{2M_{11}}\right)_{\mathrm{mod}\,2}\left(\frac{M_{14}}{2M_{11}}\right)_{\mathrm{mod}\,2}\left(\sqrt{\left(\frac{M_{12}}{2M_{11}}\right)^2 + \left(\frac{M_{13}}{2M_{11}}\right)^2 + \left(\frac{M_{14}}{2M_{11}}\right)^2 + \frac{M_{15}}{M_{11}}} - 1\right)_{\mathrm{mod}\,2}$$

We can implement the k bit $\left(4,n\right)$ secret sharing storage for the pair $(\Lambda^4, \Phi_{\Lambda^4})$.

Note that to generate the geometric method to the case with $t \ge 5$,we only need the finite dimensional version of Theorem 3.4.1. Of course, it is true but we need the notion of "area" on the high dimensional sphere first. It is a topic of advanced Mathematics, called

Haar measure [2]. If we take it for granted, the generalization of the theorem for arbitrary t is straight forward from the $t=4$ case.

5.6 Implementation of the Geometric Method

Indeed, we have implemented the circle and sphere methods and their performance is satisfactory as we expect. In this section, we only present testing results for $(\Lambda^3, \Psi_{\Lambda^3})$ with $k=24$ and $k_1=k_2=k_3=8$. A point (x,y) is stored in two floating type variables which occupy 64 bits. So the size of the container is $\frac{64}{24}=2\frac{2}{3}$ times of that of the original secret file before compression.

5.6.1 Speed Test

a) Producing Containers

The following table shows how much time needed for producing 10 uncompressed containers.

Secret Size (in MB)	2	4	6	8	10
Generating Time for 10 Containers (in Sec)	1.139	2.264	3.378	4.506	5.676

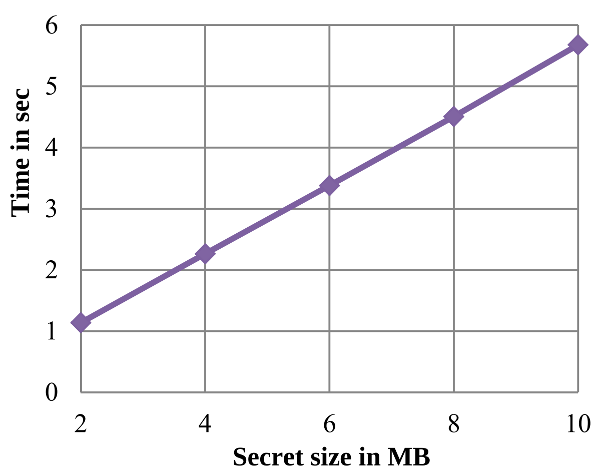

Figure 11: Time required for various sizes of secrets to generate 10 containers.

b) Recover the Original File

The following table shows how much time needed for recovering original files from their containers in a)

Secret Size (in MB)	2	4	6	8	10
Rcovert Time (in Sec)	0.151	0.313	0.457	0.607	0.763

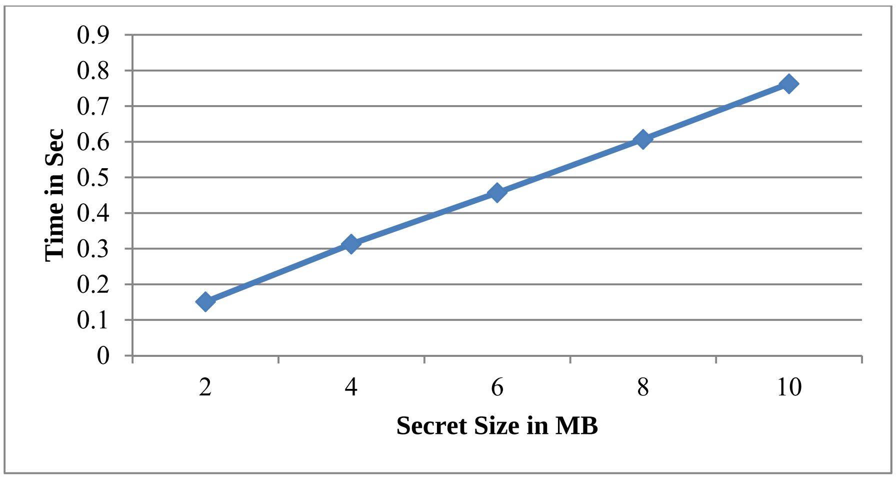

Figure 12: Time required for various sizes of secrets to recover the secret from three containers.

5.6.2 Size Tests

The following table shows that average sizes of the compressed container files of the testing group.

Secret Size (in MB)	2	4	6	8	10
Compressed Secret Size (in Kb)	987	1973	2960	3947	4934
Compressed Container Size (in Kb)	1693.37	3317.6	4917.33	6501.7	8073.47
Size Ratio of Compressed Secret and Compressed Container (in %)	172	168	166	165	164

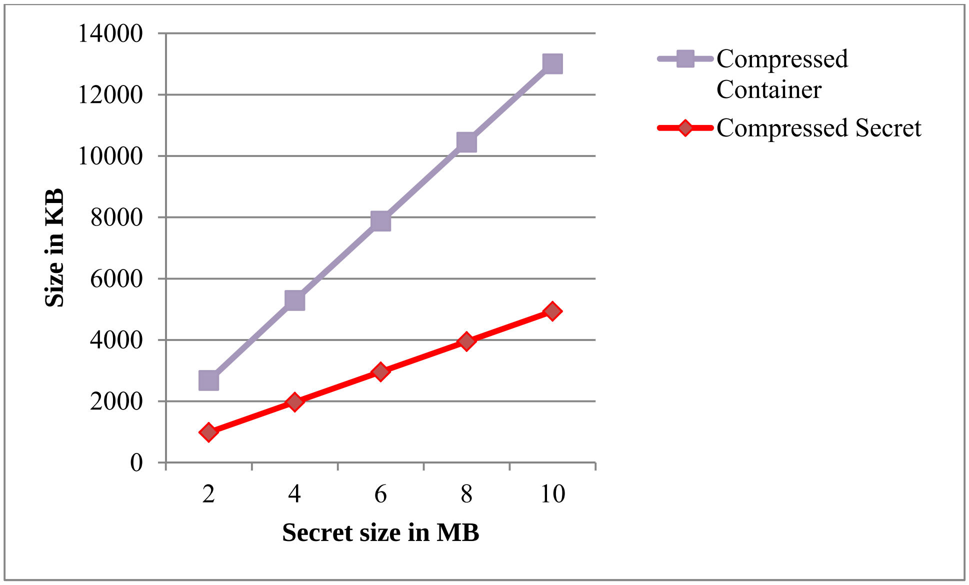

Figure 13: Comparison of Sizes of compressed secret

and compressed container for various sizes of secrets.

Reference

1. Center and Radius of a Circle from Three Points, http://www.abecedarical.com/zenosamples/zs_circle3pts.html

2. Halmos, R. "Measure Theory." Springer, 1974.

6. Number Theory-Chinese Reminder Theorem (CRT) Methods

In this chapter, we will develop a k bit $(3,n)$ secret sharing storage which is following the idea of Asmuth and Bloom [1]. Note that such n is not any integer larger than 3. It actually depends on the primes prepared for such secret sharing method. The Asmuth and Bloom secret sharing is based on Chinese remainder theorem (CRT). Firstly, we are going to demonstrate the basic idea of CRT by solving the following problem:

Find the smallest whole number that when divided by $3, 5$ and 7 *gives remainders of* $1,2,$ *and* 3*respectively.*

Formally, the above problem is asking for the smallest whole number such that

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 2 \pmod 5 \\ x &\equiv 3 \pmod 7. \end{aligned}$$

Instead of solving the above system, we would like to solve three much simpler systems:

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 0 \pmod 5 \\ x &\equiv 0 \pmod 7 \end{aligned}, \quad \begin{aligned} x &\equiv 0 \pmod 3 \\ x &\equiv 1 \pmod 5 \\ x &\equiv 0 \pmod 7 \end{aligned}, \quad \begin{aligned} x &\equiv 0 \pmod 3 \\ x &\equiv 0 \pmod 5 \\ x &\equiv 1 \pmod 7 \end{aligned}$$

The solutions of the first, second and third system are 70, 21 and 15 respectively. We call these numbers base solutions. Consider

$$70\times 1+21\times 2+15\times 3=157.$$

Although 157 divided by $3, 5$ and 7 gives remainders of $1, 2$ and 3 respectively, it is too big. Hence, $x=157-\mathrm{LCM}(3,5,7)=157-105=52$ is the answer. Moreover, if we want to solve a system such as

$$\begin{aligned} x &\equiv y_1 \pmod 3 \\ x &\equiv y_2 \pmod 5, \\ x &\equiv y_3 \pmod 7 \end{aligned}$$

then the general solutions are $70\times y_1+21\times y_2+15\times y_3-\rho\times 105$ where $\rho\in\mathbb{Z}$. Therefore, the key is obtaining the base solution. Formally, we can state the Chinese Remainder Theorem as below:

Theorem :(Chinese Remainder Theorem) Let $p_1, p_2, \cdots,$ and p_m be increasing distinct primes. For any integers $a_1, a_2, \cdots, a_m$, there is an integer x with

$$\begin{aligned} x &\equiv a_1 \pmod{p_1} \\ x &\equiv a_2 \pmod{p_2} \\ x &\equiv a_3 \pmod{p_3} \\ &\vdots \\ x &\equiv a_m \pmod{p_m} \end{aligned}$$

and x is unique mod $p_1 p_2 \cdots p_m$.

6.1 Three Shareable Set $\Lambda(p_1p_2\ldots p_m)$

Let $\{p_1, p_2, \cdots, p_m\}$ be the set of increasing prime numbers such that

$$p_{m-1} p_m < p_1 p_2 p_3 .$$

and let $\ell = \dfrac{p_1 p_2 p_3 - p_{m-1} p_m}{p_{m-1} p_m}$. For $p_{m-1} p_m < l < p_1 p_2 p_3$, we define

$$C_l = \{ (p_i, l (\bmod\ p_i)) : i = 1, 2, \cdots, m \} \subset \mathbb{N}^2 .$$

and

$$\Lambda(p_1, p_2, \cdots, p_m) = \{ C_l \subset \mathbb{N}^2 : p_{m-1} p_m < l < p_1 p_2 p_3 \} .$$

Suppose that C_{l_1} and C_{l_2} has three distinct points in common. Hence there are three prime $\overline{p}_1$, $\overline{p}_2$ and $\overline{p}_3$ such that

$$\begin{aligned} (\overline{p}_1, l_1 \bmod \overline{p}_1) &= (\overline{p}_1, l_2 \bmod \overline{p}_1) \\ (\overline{p}_2, l_1 \bmod \overline{p}_2) &= (\overline{p}_2, l_2 \bmod \overline{p}_2) \\ (\overline{p}_2, l_1 \bmod \overline{p}_2) &= (\overline{p}_2, l_2 \bmod \overline{p}_2) \end{aligned}$$

Since by CRT, l_1 and l_2 are the unique solutions of the systems

$$\begin{array}{lll} x \equiv l_1 \pmod{\overline{p}_1} & & x \equiv l_2 \pmod{\overline{p}_1} \\ x \equiv l_1 \pmod{\overline{p}_2} & \text{and} & x \equiv l_2 \pmod{\overline{p}_2} \text{ respectively} \\ x \equiv l_1 \pmod{\overline{p}_3} & & x \equiv l_2 \pmod{\overline{p}_3} \end{array}$$

and the systems in above are equivalent, we have $l_1 = l_2$. Hence, $C_{l_1} = C_{l_2}$.

Ψ can be defined as follow. Given three distinct points $(\overline{p}_1, y_1)$, $(\overline{p}_2, y_2)$ and $(\overline{p}_3, y_3)$ in $C \in \Lambda(p_1, p_2, \cdots, p_m)$, by applying CRT to the system

$$\begin{aligned} x &\equiv y_1 \mod \overline{p}_1 \\ x &\equiv y_2 \mod \overline{p}_2 , \\ x &\equiv y_3 \mod \overline{p}_3 \end{aligned}$$

we have a unique solution $0 < l \leq \overline{p}_1 \overline{p}_2 \overline{p}_3$. Since $p_1 p_2 p_3 \geq \overline{p}_1 \overline{p}_2 \overline{p}_3$ we have $C_l = C$.

Lastly, Let $(\overline{p}_1, y_1)$ and $(\overline{p}_2, y_2)$ be two distinct points in C_l. Suppose that $\overline{l}$ is the unique solution of the system

$$\begin{aligned} x &\equiv y_1 \mod \overline{p}_1 \\ x &\equiv y_2 \mod \overline{p}_2 \end{aligned}$$

which is between 0 and $\overline{p}_1 \overline{p}_2$. Since $\overline{p}_1 \overline{p}_2 \leq p_{m-1} p_m$ there exist at least ℓ numbers between $p_{m-1} p_m$ and $p_1 p_2 p_3$ equivalent to $\overline{l}$ in mod $\overline{p}_1 \overline{p}_2$. Therefore, C_l cannot be determined.

Hence, $\Lambda(p_1, p_2, \cdots, p_m)$ is 3 shareable.

6.2 Encoding Function $\Phi_{\Lambda(p_1 p_2 \ldots p_m)}$

Let k be the biggest positive integer such that $p_{m-1} p_m < 2^{k+1} - 1 < p_1 p_2 p_3$. We would like to define $\Phi_{\Lambda(p_1, p_2, \cdots, p_m)}$ to be a function that maps a k bit string $\epsilon_1 \epsilon_2 \cdots \epsilon_k$ to a circle C_l such that

$$l = \epsilon_1 + \epsilon_2 2 + \cdots + \epsilon_k 2^k .$$

Then $\Phi_{\Lambda(p_1, p_2, \cdots, p_m)}$ is one to one since binary representation of integers is unique and hence, $(\Lambda(p_1, p_2, \cdots, p_m), \Phi_{\Lambda(p_1, p_2, \cdots, p_m)})$ is a k bit 3 secret sharing pair.

6.3 Choice functions of $\Lambda(p_1 p_2 \ldots p_m)$ and the Calculation of $\Phi_{\Lambda(p_1 p_2 \ldots p_m)}{}^{-1}(\Psi)$

Let $3 < n \leq m$. For any $j = 1, 2, \cdots, n$, we define

$$\pi_j(C_l) = (p_j, l(\mathrm{mod}\ p_j))$$

for all $p_{m-1}p_m \le l < p_1p_2p_3$.

Given three distinct points $(\overline{p}_1, y_1)$, $(\overline{p}_2, y_2)$ and $(\overline{p}_3, y_3)$ in

$$\Phi_{\Lambda(p_1,p_2,\cdots,p_m)}(\epsilon_1\epsilon_2\cdots\epsilon_k) \in \Lambda(p_1, p_2, \cdots, p_m).$$

Then

$$\Phi_{\Lambda(p_1,p_2,\cdots,p_m)}(\epsilon_1\epsilon_2\cdots\epsilon_k) = C_l$$

where l is the unique solution of the system:

$$\begin{aligned} x &\equiv y_1 \mod \overline{p}_1 \\ x &\equiv y_2 \mod \overline{p}_2 \, . \\ x &\equiv y_3 \mod \overline{p}_3 \end{aligned}$$

between $p_{m-1}p_m$ and $p_1p_2p_3$. Then $\epsilon_1\epsilon_2\cdots\epsilon_k = (l - p_{m-1}p_m)_{\mod 2}$.

Hence, a k bit $(3,n)$ secret sharing storage can be launched for the k bit 3 secret sharing pair $(\Lambda(p_1, p_2, \cdots, p_m), \Phi_{\Lambda(p_1,p_2,\cdots,p_m)})$.

Example 6.3.1: Consider the following table:

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
31	37	41	43	47	53	59	61	67	71

Therefore, we have

$$p_1p_2p_3 = 47027 \text{ and } p_9p_{10} = 4757.$$

Hence, $\ell = \dfrac{p_1p_2p_3 - p_9p_{10}}{p_9p_{10}} = 8.884$.

Example 6.3.2: Also, look at

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
37	41	43	47	53	59	61	67	71

So,

$$p_1p_2p_3 = 65231 \text{ and } p_8p_9 = 4757.$$

Hence, $\ell = \dfrac{p_1p_2p_3 - p_8p_9}{p_8p_9} = 12.71.$

6.4 Implementation of the CRT Method

We consider the following sequence of prime numbers

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
41	43	47	53	59	61	67	71	73	79

and

$$p_1 p_2 p_3 = 82861 \text{ and } p_9 p_{10} = 5767 .$$

Hence, $\ell = \frac{p_1 p_2 p_3 - p_9 p_{10}}{p_9 p_{10}} = 13.37$. The size of containers is now half of the original secret file.

6.4.1 Speed tests

a) Producing Containers

The following table shows how much time needed for producing 10 uncompressed containers.

Secrect Size (in MB)	2	4	6	8	10
Generating Time for 10 containers (in sec)	0.161	0.304	0.454	0.588	0.718

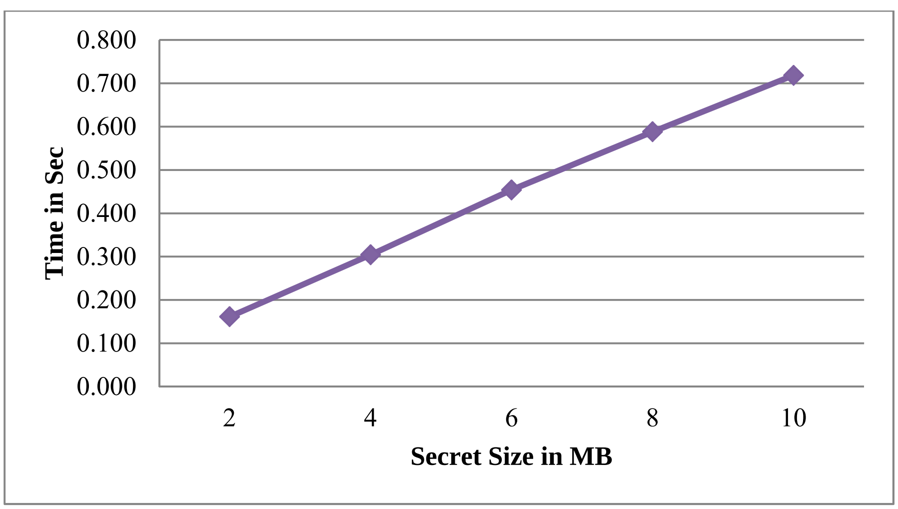

Figure 14: Time required for various sizes of secrets to generate 10 containers.

c) Recover the Original File

The following table shows how much time needed for recovering original files from their containers in a)

Secret Size (in MB)	2	4	6	8	10
Recover Time (in sec)	0.037	0.078	0.104	0.150	0.188

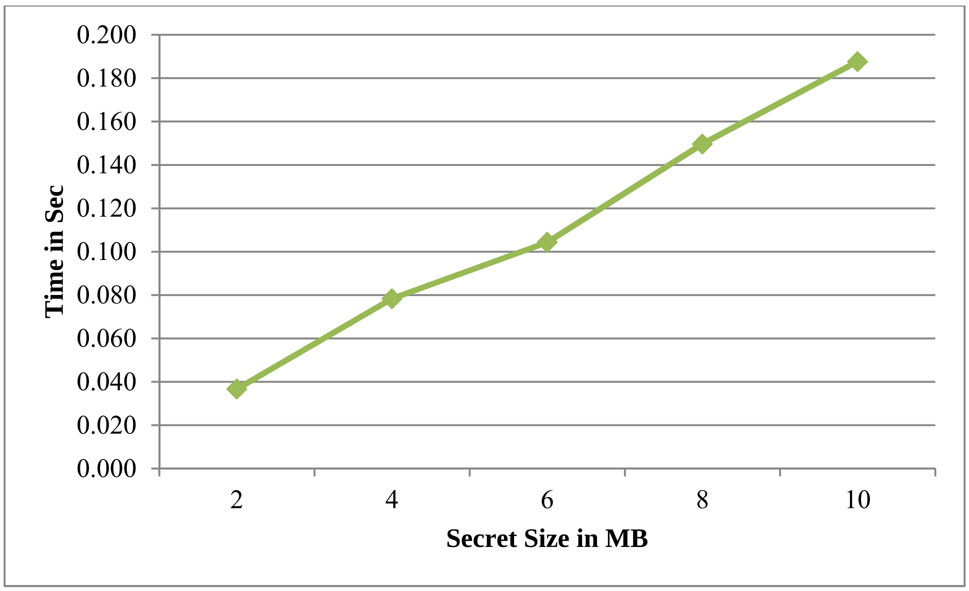

Figure 15: Time required for various sizes of secrets to recover the secret from three containers.

6.4.2 Size Tests

The following table shows that average sizes of the compressed container files of the testing group.

Secret Size (in MB)	2	4	6	8	10
Compressed Secret Size (in KB)	987	1973	2960	3947	4934
Compressed Container Size (in KB)	394.1	788.5	1183.4	1577	1971.9
Size Ratio of Compressed Secret and Compressed Containers (in %)	40	40	40	40	40

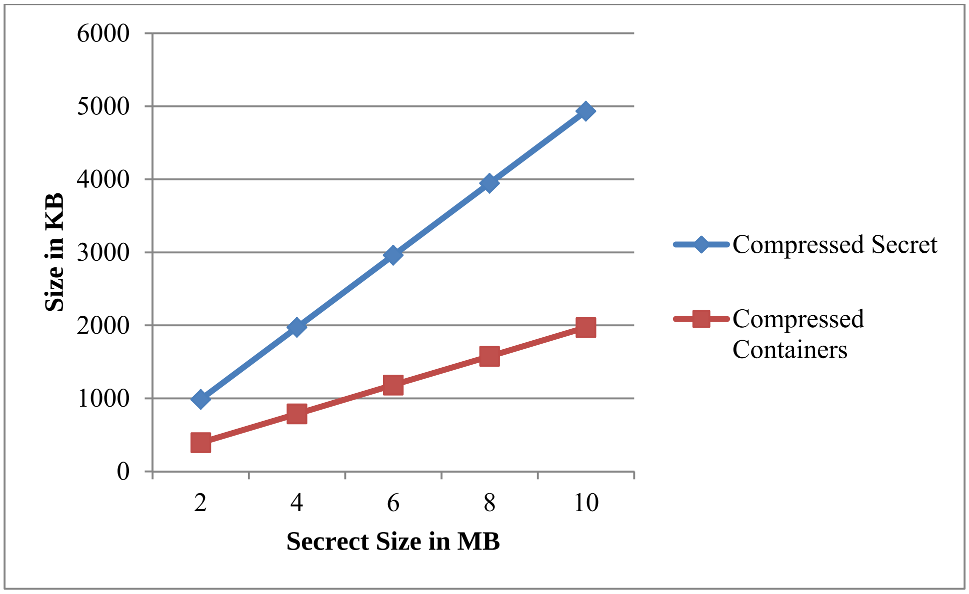

Figure 16: Comparison of Sizes of compressed secret

and compressed container for various sizes of secrets.

Reference

1. Asmuth, C.A. and Bloom, J. "A modular approach to key safeguarding". IEEE *Transactions on Information Theory*, IT-29(2):208-210, 1983.

7 Conclusion

In this report, we have mainly developed an abstract framework of k bit t secret sharing framework for cloud storage. This framework is clean and it can be implemented. A successful implementation of the framework would provide users with protection when the system is under the attack on its confidentiality, integrity and reliability. Furthermore, such system has its own encryption by using permutations and tailor made error detection, location and data rescue.

We make use of Lagrange polynomials and take the advantages of the algebraic property "t distinct points on the plane can uniquely determine a polynomial function of degree $t-1$" to design a k bit (t,n)-secret sharing distributed storage. We employ the set with unique factorization property (UFP) so that we simply need to calculate the y intercept of a Lagrange polynomial and then use a look up table to recover a secret. Moreover, the set which has minimum UFP would help us to design storage with smallest containers.

In addition to the algebraic methods, we can utilize the geometric facts that

a) three non collinear points determine a unique circle;
b) four non coplanar points determine a unique sphere.

to construct k bit $(3,n)$ and k bit $(4,n)$ secret sharing storage respectively. To generalize to arbitrary case, it is straight forward if we have defined the Haar measure on the higher dimensional unit sphere.

The last method is an application of Chinese Remainder Theorem and we have designed k bit (t,n) -secret sharing distributed storage and one of the designs can offer containers with the same size of the original secret. However, such k is no longer unrestricted and it is chosen within a certain range.

We have developed a C program for implementing both algebraic and geometric k bit $(3,n)$ secret sharing distributed storages. The performances of both algebraic and geometric designs are satisfactory in term of processing time and compressed container size. The container size reaches 50% of size of the original secret and 40% after compression in the CRT case. Besides, it is very speedy.

Finally, concerning the framework of distributed storage and its techniques, the notion is, clean, cute and mostly original. Also, it is proved to be working efficiently.

履歷表

姓名：張顥霆

性別：男

出生日期：1996 年 12 月 4 日

學歷： 澳門培正中學(高三年級)

獲獎經驗

年份	比賽/活動	主辦單位	所獲獎項	備註
2013	港澳數學奧林匹克公開賽	香港數學奧林匹克協會	金獎	第二名
2013	全國青少年信息學奧林匹克競賽	中國計算機學會	決賽選手	
2013	澳門信息學奧林匹克選拔賽	澳門電腦學會	一等獎	
2011	全澳校際數學比賽	澳門教育暨青年局	二等獎	
2010, 2009, 2008, 2007	全澳校際數學比賽	澳門教育暨青年局	一等獎	
2009, 2008, 2007	港澳數學奧林匹克公開賽	香港數學奧林匹克協會	金獎	

履歷表

姓名：譚知微

英文姓名拼音：TAN, Chih Wei

性別：女

出生日期：1996 年 09 月 27 日

學歷

2008-	培正中學
2001-	聖庇護十世音樂學院
2001-2008	培正中學小學部

才藝

鋼琴	皇家音樂學院八級 (Pass with Distinction)
大提琴	

參加團體

2008	澳門青年交響樂團 A 團	大提琴手
2007	澳門青年交響樂團 B 團	大提琴手
2006- 2010	澳門培正中學絃樂團	大提琴手

會員

2012-	Member of Society for Science & the Public
2013-	Student Member of Association for Computing Machinery

所獲獎項

年份	比賽／獎勵名稱	頒發機構	所獲獎勵	備注
2013	第 28 屆年全國青少年科技創新大賽 The 26th China Adolescents Science and Technology Innovation Contest	周培源基金會和中國科協、教育部、科技部、國家環保總局、國家體育總局、國家自然科學基金委、全國少工委、全國婦聯、國家自然科學基金委和南京人民政府共同主辦。	周培源青少年科技創新獎和二等銀獎	一等獎缺，二等獎兩名
2013	紅藍之光	澳門培正中學	得獎者	因在數學及科學上有優異表現，為澳為校爭光而獲獎。
2013	科普活動傑出獎學金	澳門培正中學	得獎者	
2013	Intel 國際科學與工程大獎賽 Intel International Science and Engineering Fair	Society for Science & the Public (SSP)	Finalist	Nominated -The IEEE Foundation Presidents' Scholarship Award
2013	科技創新成果競賽 2013	澳門教育暨青年局	高中組個人項目優異獎	
2012	第 3 屆丘成桐中學應用數學科學獎 The 3th Shing –Tung Yau Mathematical Science Award	丘成桐教授 泰康人壽保險股份有限公司	優勝獎	全球決賽
2012	澳門科學與工程大獎賽 2012	澳門教育暨青年局	優異獎	
2012	第 5 屆丘成桐中學數學科學獎-南部賽區	中山大學	一等獎	中國南部賽區決賽
2012	紅藍之光	澳門培正中學	得獎者	
2012	科普活動傑出獎學金	澳門培正中學	得獎者	
2012	Intel 國際科學與工程大獎賽 Intel International Science and Engineering Fair	Society for Science & the Public (SSP)	Finalist	國際大賽
2011	第 26 屆年全國青少年科技創新大賽 The 26th China Adolescents	中國科協、教育部、科技部、國家環保總局、國家體育總局、	內蒙古自治區主席獎和一等金獎	全國獎項

	Science and Technology Innovation Contest	國家自然科學基金委、全國少工委、全國婦聯、國家自然科學基金委和內蒙古自治區人民政府共同主辦		
2011	紅藍之光	澳門培正中學	得獎者	因在科學，數學，寫作及音樂上有優異表現，為澳為校爭光而獲獎。
2010	首屆澳門十大傑出少年選舉 1st Macau 10 outstanding teenager election	澳門基督教青年會 Y. M. C. A. Macau	十大傑出少年	
2010	第 8 屆走進美妙的數學花園–中國青少年數學論壇	中國少年科學院	一等一名金獎	全國獎項
2010	第 5 屆中國中學生作文比賽–澳區比賽	澳門青年聯合會	優異獎	獲選全國賽澳區代表
2010	聖庇護十世音樂院獎學金	聖庇護十世音樂院	高階組得獎者	各階選一人得獎
2009	第 27 屆澳門青年音樂比賽-鋼琴四手聯彈初級	澳門文化局	第二名	公開賽獎項
2009	區師達神父獎學金	聖庇護十世音樂院	得獎者	
2008	第 25 屆全澳學生朗誦比賽-普通話高小獨誦	澳門中華教育會	二等獎	校際賽獎項
2007	“我與世界遺產”中國校際作文徵集活動	中國聯合國教科文組織全國委員會	一等獎	全國獎項
2007	澳門青年交響樂團 – 優秀團員獎 (少年團)	澳門青年交響樂團	得獎者	
2006	第 23 屆全澳學生朗誦比賽-普通話初小獨誦	澳門中華教育會	二等獎	校際賽獎項
2005	第 23 屆澳門青年音樂比賽-鋼琴獨奏 B 組	澳門文化局	第三名	公開賽獎項
2005 - 2013	鋼琴成績優異獎	聖庇護十世音樂院	得獎者	

著作

譚知微 曾學為，""眾妙之門"——數學原理在電子門禁系統上的應用"，《中国科技教育[1]》2012 年 第 2 期 26-28 頁及第二十六屆全國青少年科技創新大賽獲獎作品集，科學普及出版社，180-183 頁

公演及展示

23-29/08/2013　香港第四十六屆聯校科學展覽（澳門區唯一所獲邀項目）

25/07/2013　聖庇護十世音樂學院五十週年慶典音樂會-雙鋼琴八手聯彈

https://www.youtube.com/watch?v=2azl-6OncQA

20/01/2013　扶樂普天頌-區師達神父作品音樂會-鋼琴獨奏

https://www.youtube.com/watch?v=bOPpoQQmzuk (由 00:20 開始）

https://www.youtube.com/watch?v=Lmpw21-s5BQ

(樂。跡。區師達神父生平及手稿展覽宣傳片 Opening Music)

11/09/2010　澳門青年交響樂團- 13 週年會慶音樂會

08/07/2010　培正中學管弦樂團新加坡公演

03/04/2009　培正創校 120 周年紀念音樂會-管弦樂團之大提琴,鋼琴四手聯彈。

20/12/2008　澳門回歸九週年-澳門青年交響樂團

29/11/2008　踏出第一步青少年音樂會-大提琴獨奏

30/08/2008　澳門青年交響樂團十一週年會慶音樂會-大提琴

27/08/2008　澳門青年交響樂團-深圳之行-深圳市保利劇院

02/08/2008　聖庇護十世音樂學院四十五週年慶典音樂會-雙鋼琴八手聯彈

[1]《中國科技教育》是中國科學技術協會主管、中國青少年科技輔導員協會主辦的一本關於科技教育的國家級專業科普刊物。