On Highly Secure and Available Data Storage Systems*

Sung Jin Choi[1], Hee Yong Youn[1], and Hyung Soo Lee[2]

[1] School of Information and Communications Engineering
Sungkyunkwan University, Suwon, Korea
{choisj, youn}@ece.skku.ac.kr
[2] Ubiquitous System Research Center,
Korea Electronics Technology Institute, Pyoungtaek, Korea
hslee@keti.re.kr

Abstract. Rapid technological advances are resulting in a greater use of data intensive applications. For this reason and that of the alarming growth in electronic crime, security and availability are critical factors that must be considered in designing a system. This paper proposes a novel data dispersal/encryption scheme to improve both the availability and security of distributed storage system by using the Singular Values Decomposition (SVD) theorem. It handles data represented by any size matrix, and it also allows complete recovery of original data even though stored data are partially damaged. Analysis shows that it improves the availability about 10% compared with an efficient existing scheme for both read and write operation, while it allows secure storage simultaneously.

1 Introduction

Rapid technological advances are resulting in a greater use of distributed storage system in world-wide organizations for communication, transport, banking, education, manufacturing, medicine, etc., leading to the handling of large quantities of data that are especially important for these organizations on many occasions. For this reason and that of the alarming growth in electronic crime, security becomes a serious aspect that must be considered specifically for distributed storage system. At the same time, due to harsh and unpredictable distributed environment, the system needs to be designed to provide continuous service even in the presence of failures. Indeed, both security and availability are two important criteria for distributed storage systems [1].

The main issues focused in typical distributed systems including database systems mainly have been with the following aspects: confidentiality, integrity, and availability. The range of possible threats that affect each one of the three factors is so wide that they cannot be tackled as a whole. Data distribution is one of the key techniques developed for achieving a desired level of security and availability, and it involves data encoding and partitioning algorithm. There exist many such algorithms applicable to distributed storage system including encryption, replication, striping, secret

* This work was supported by 21C Frontier Ubiquitous Computing and Networking, Korea Research Foundation Grant (KRF - 2003 - 041 - D20421) and Brain Korea 21 Project in 2004. Corresponding author : Hee Yong Youn

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 121–129, 2004.

sharing, and various combinations of them. They offer different levels of performance (throughput), availability (probability that data can be accessed), and security (effort required to compromise the confidentiality or integrity of stored data). Development of new efficient data distribution and protection scheme is inevitable for allowing highly available but secure system at the same time without significantly affecting the performance [2].

In distributed storage system security and availability of data are greatly influenced by the policy how the data are dispersed. High security and availability are usually conflicting requirement. This paper, however proposes a new scheme that can allow high security and availability of distributed storage system. It is achieved by using the SVD theorem which is an important matrix decomposition that applicable to any size matrix. A scheme targeting the same goal was presented in [3] using matrix eigenvalue. However, it can handle data represented by only square matrix and eigenvalues are not guaranteed to be found, while the proposed scheme solves both the problems. It also allows complete recovery of original data even though stored data are partially damaged. Analysis shows that the availability is improved about 10% compared with information dispersal scheme, while it allows secure storage since decryption without the key is absolutely impossible.

The rest of the paper is organized as follows. Section 2 discusses existing approaches for data encryption and distribution, and Section 3 presents the proposed scheme. Section 4 evaluates and compares the performance of the proposed scheme with the earlier schemes, and finally concluding remark is given in Section 5.

2 Related Work

There is a wide array of data distribution algorithms including encryption, replication, striping, erasure-resilient coding, information dispersal, and secret sharing. Threshold algorithms, characterized by three parameters (p, m, and n), represent a large set of these algorithms. In a p-m-n threshold scheme, data is encoded into n shares such that any m of the shares can reconstruct the data and less than p reveal no information about the encoded data. Thus, a stored value is available if at least m of the n shares can be retrieved. Attackers must compromise at least p storage nodes before it is even theoretically possible to determine any part of the encoded data. Table I lists a number of well-known threshold schemes [4].

Table 1. Threshold schemes

p-m-n	Scheme
1-1-n	Replication
1-n-n	Striping
n-n-n	Splitting
1-m-n	Information Dispersal
m-m-n	Secret Sharing
p-m-n	Ramp Schemes

The simplest example of threshold scheme is n-way replication, which is a *1-1-n* threshold scheme. That is, out of the *n* replicas that are stored, any single replica provides the original data (*m = 1*), and each replica reveals information about the encoded data (*p = 1*). Another example is striping (or decimation, as in disk arrays), wherein a large block of data is partitioned into *n* sub-blocks, each containing one partition of the data (so, *p = 1* and *m = n*). At the other end of the example is splitting, an *n-n-n* threshold scheme that consists of storing *n-1* random values and one value is the exclusive-or of the original value and those *n-1* values; *p = m = n* for splitting, since all *n* shares are needed to recover the original data. Replication, decimation, and splitting schemes have a single tunable parameter, *n*, which affects their place in the trade-off space. With more mathematics, the full range of *p-m-n* threshold schemes becomes available. For example, secret sharing schemes are *m-m-n* threshold schemes. Shamir's implementation of secret sharing is based on interpolating points on a polynomial in a finite field. The secret values along with *m-1* randomly generated values uniquely determine the encoding polynomial of order *m-1*. Each share is generated by evaluating the polynomial at distinct points.

Information dispersal algorithm, a *1-m-n* threshold scheme, uses the same polynomial based mathematics as Shamir's secret sharing, but no random number; *m* secret values are used to determine the unique encoding polynomial. Thus, each share reveals partial information about the *m* simultaneously encoded values, but the encoding is much more space efficient. Ramp schemes are *p-m-n* threshold schemes, and they can also be implemented with the same approach. The points used to uniquely determine the encoding polynomial are *p-1* random values and *m-(p-1)* secret values. Ramp schemes thus theoretically offer confidentiality of up to *p-1* shares. They are also more space efficient than secret sharing (so long as *m>p*). For *p=1*, ramp schemes are equivalent to information dispersal; for *p=m*, they are equivalent to secret sharing [5].

3 The Proposed Scheme

We first present the theorem and definition required to explain the proposed encryption/dispersal scheme.

3.1 SVD's Theorem and Singular Values (Finding V, D, and U)

Theorem 1: Let A be an $m \times n$ matrix and $\sigma_1, ..., \sigma_r$ be all its nonzero singular values. Then there are orthogonal matrices U ($m \times m$) and V ($n \times n$) and an $m \times n$ matrix D of the form such that

$$A = UDV^T. \quad (1)$$

Proof 1: U, V, and D (of the indicated sizes) have been already explicitly defined. Moreover, U and V are orthogonal. It remains to show only that $A = UDV^T$. It suffices to show that AV = UD, because $V^T = V^{-1}$.

$$u_i = \frac{1}{\sigma_i} Av_i \text{ for } i = 1,...,r . \tag{2}$$

By Eq. (2) $\sigma_i u_i = Av_i$ for $i = 1,....,r$ and $\|Av_i\| = \sigma_i = 0$ for $i = r+1,\ldots,n$. So, $Av_i = 0$ for $i= r+1,...,n$. Therefore,

$$AV = [Av_1 \cdots Av_n] = [\sigma_1 u_1 \cdots \sigma_r u_r 0 \cdots 0]$$

$$= [u_1 \cdots u_m] = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \sigma_r & \vdots \\ 0 & \cdots & 0 \end{bmatrix} . \tag{3}$$

Let A be an $n \times n$ matrix. A nonzero vector v is an eigenvector of A if Eq. (4) holds for some scalarλ. λ is called an eigenvalue of A corresponding to the eigenvector v. Eigenvalues are also known as characteristic, proper values, or latent roots.

$$Av = \lambda v . \tag{4}$$

We define V, and then find the σ_i's along the diagonal of D. Consider the $n \times n$ symmetric matrix $A^T A$. By the spectral theorem, $A^T A$ is orthogonally diagonalizable and has eigenvalues, say $\lambda_1,...,\lambda_n$. Let $v_1,...,v_n$ be the corresponding eigenvectors so that they form an orthonormal basis of R^n . V is simply

$$V = [v_1 v_2 \cdots v_n] . \tag{5}$$

Next, we observe that all the eigenvalues are nonnegative (so $A^T A$ is positive semidefinite). Because $(A^T A)v_i = \lambda_i v_i$ and $\| v_i \| = 1$, we have $0 \le \|Av_i\|^2 = (Av_i)^T Av_i = V_i^T A^T Av_i = v_i^T \lambda_i v_i = \lambda_i \|v_i\|^2 = \lambda_i$. Hence, $\lambda_i \ge 0$ for $i=1,\ldots,n$. By renumbering, if necessary, we order the λ_i values from largest to smallest and take their square roots, such that $\sigma_1 = \sqrt{\lambda_1} \ge \cdots \ge \sigma_n = \sqrt{\lambda_n} \ge 0$. So, $\sigma_i = \|Av_i\|$, $i = 1,.....,n$. The numbers $\sigma_1,...,\sigma_n$ are called the singular values of A, and they carry important information on A. These are the diagonal entries of D [6].

3.2 Encryption/Dispersal Scheme

First, we find the singular values of the elements of D. Therefore, we need a general method for finding eigenvalue and eigenvector by using Eq. (4) and it is the Power Method. It computes the dominant eigenvalue and an eigenvector corresponding to the dominant eigenvalue. Without loss of generality, it is necessary to assume that A has the following two properties:

i. There is a single eigenvalue of maximum modulus.
ii. There is a linearly independent set of n eigenvectors.

According to the first assumption, the eigenvalues can be labeled such that $|\lambda_1| > |\lambda_2| \ge |\lambda_3| \ge \cdots \ge |\lambda_n|$. According to the second assumption, there is a basis $\{u^{(1)}, u^{(2)},\ldots,u^{(n)}\}$ for C^n such that

$$Au^{(j)} = \lambda_j u^{(j)} \ (1 \le j \le n) . \tag{6}$$

Let $x^{(0)}$ be an element of C^n such that when $x^{(0)}$ is expressed as a linear combination of the basis elements $u^{(1)}, u^{(2)}, ..., u^{(n)}$, the coefficient of $u^{(1)}$ is not 0. Thus,

$$x^{(0)} = a_1 u^{(1)} + a_2 u^{(2)} + \cdots + a_n u^{(n)} \ (a_1 \neq 0) . \tag{7}$$

We form then $x^{(1)}=Ax^{(0)}$, $x^{(2)}=Ax^{(1)}$,L,$x^{(k)}=Ax^{(k-1)}$ to have

$$x^{(k)} = A^k x^{(0)} . \tag{8}$$

In the following analysis there is no loss of generality in absorbing all the coefficients a_j in the vectors $u^{(j)}$ that they multiply. Hence, we may rewrite Eq. (7) as

$$x^{(0)} = u^{(1)} + u^{(2)} + \cdots + u^{(n)} . \tag{9}$$

By this equation and (8), we have $x^{(k)} = A^k u^{(1)} + A^k u^{(2)} + \cdots + A^k u^{(n)}$. Using Eq. (6), we arrive at $x^{(k)} = \lambda_1^k \left[u^{(1)} + \left(\frac{\lambda_2}{\lambda_1}\right)^k u^{(2)} + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k u^{(n)} \right]$. Since $|\lambda_1| > |\lambda_j|$ for $2 \le j \le n$, we see that the coefficients $\left(\frac{\lambda_j}{\lambda_1}\right)^k$ tend to 0 and the vector within the brackets converges to $u^{(1)}$ as $k \to \infty$. To simplify the notation, we write $x^{(k)}$ in the form $x^{(k)} = \lambda_1^k \left[u^{(1)} + \varepsilon^{(k)} \right]$, where $\varepsilon^{(k)} \to 0$ as $k \to \infty$. In order to be able to take ratios, let φ be any linear functional on C^n for which $\varphi(u^{(1)}) \neq 0$. Then $\varphi(x^{(k)}) = \lambda_1^k \left[\varphi(u^{(1)}) + \varphi(\varepsilon^{(k)}) \right]$. Consequently, the following ratios converges to λ_1 as $k \to \infty$: $r_k \equiv \frac{\varphi(x^{(k+1)})}{\varphi(x^{(k)})} = \lambda_1 \left[\frac{\varphi(u^{(1)}) + \varphi(\varepsilon^{(k+1)})}{\varphi(u^{(1)}) + \varphi(\varepsilon^{(k)})} \right] \to \lambda_1$.

This constitutes the Power Method for computing λ_1. Since the direction of the vector $x^{(k)}$ aligns more and more with $u^{(1)}$ as $k \to \infty$, the method can also give us the eigenvector, $u^{(1)}$. The eigenvectors found are

$$v^{(1)} = \left[s_1 \cdots s_n \right] \ s_n \in R, \ v^{(2)} = \left[t_1 \cdots t_n \right] \ t_n \in R, \ldots\ldots, \ v^{(n)} = \left[z_1 \cdots z_n \right] \ z_n \in R . \tag{10}$$

Therefore, by Eq. (2) and Eq. (5), U ($U = \frac{1}{\sigma_i} A[v^{(1)} v^{(2)} \cdots v^{(n)}]$) and V ($V = [v^{(1)} v^{(2)} \cdots v^{(n)}]$) are actually stored data and matrix D consisting of σ becomes Decryption Key.

Example 1: If the original data is $A = \begin{bmatrix} -2 & 1 & 2 \\ 6 & 6 & 3 \end{bmatrix}$, we first calculate $A^T A$. Then we derive λ_1=81, $\lambda_2 = 9$, $\lambda_3 = 0$ by using Theorem 2. Eigenvectors that correspond to each λ become $\left[2s \ 2s \ s\right]$, $\left[-2t \ t \ 2t\right]$, $\left[u \ -2u \ 2u\right]$ s,t,u $\in R$. U is calculated by us-

ing $U = \frac{1}{\sigma_i} A[v^{(1)} v^{(2)} \cdots v^{(n)}]$. The data actually stored in each node and Decryption Key D and U are as follows. Here s=1, t=2 and u=3 are randomly selected.

$$\text{Node1}=[2\ 2\ 1],\ \text{Node2}=[-4\ 2\ 4],\ \text{Node3}=[3\ -6\ 6],\ U=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix},\ D=\begin{bmatrix}9 & 0 & 0\\ 0 & 3 & 0\end{bmatrix}$$

An important property of the proposed scheme is that it allows secret dispersal as a general threshold scheme and encryption of data at the same time. That is, one cannot extract original data even though the stored data U and V are available since it is NP-hard problem to decide the original matrix using U and V. Refer to the example above. There exist infinite ways to form V matrix by arbitrarily deciding the s, t, and u values. As a result, the proposed scheme offers high availability and security as well as complete recovery of original data even though data are partially damaged.

3.3 Data Recovery Scheme

We next show how the original data is recovered using Gram-Schmidt Process.

Theorem 2: Any subspace W of R^n has at least one orthogonal basis and at least one orthonormal basis. If $B = \{a_1, ..., a_n\}$ is a basis of W, then $B' = \{b_1, ..., b_n\}$ is an orthogonal basis, where

$$b_1 = a_1, ..., b_k = a_k - \frac{a_k \cdot b_1}{b_1 \cdot b_1} b_1 - \frac{a_k \cdot b_2}{b_2 \cdot b_2} b_2 \cdots - \frac{a_k \cdot b_{k-1}}{b_{k-1} \cdot b_{k-1}} b_{k-1} \tag{11}$$

Proof 2: An orthonormal Basis B'' is obtained by normalizing B':

$$B'' = \left(\frac{b_1}{\|b_1\|}, \ldots, \frac{b_k}{\|b_k\|}\right). \tag{12}$$

Let V $(v_1, ..., v_n)$ be the corresponding eigenvectors, so that they form an orthonormal basis of R^n. V^T is as follows, by Eq. (12).

$$V^T = \left[\frac{v_1}{\|v_1\|} \cdots \frac{v_k}{\|v_k\|}\right] \quad . \tag{13}$$

Therefore, the original data can be recovered from U, V that were stored in the storage nodes by Eq. (1).

Example 2: We obtain V^T using Gram-Schmidt orthonormal Process after reading the data stored in Node1, Node2 and Node3. Therefore,

$$v_1=[2, 2, 1],\ v_2=[-4, 2, 4],\ v_3=[3, -6, 6]$$

$$V_1 = \frac{v_1}{\| v_1 \|} = \frac{1}{3} v_1 = \begin{bmatrix}2/3\\ 2/3\\ 1/3\end{bmatrix},\ V_2 = \frac{v_2}{\| v_2 \|} = \frac{1}{6} v_2 = \begin{bmatrix}-2/3\\ 1/3\\ 2/3\end{bmatrix},\ V_3 = \frac{v_3}{\| v_3 \|} = \frac{1}{6} v_3 = \begin{bmatrix}1/3\\ -2/3\\ 2/3\end{bmatrix},$$

V^T is $\begin{bmatrix} 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \\ 1/3 & 2/3 & 2/3 \end{bmatrix}$. We get the original data by equation $A = UDV^T$.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 9 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}\begin{bmatrix} 2/3 & -2/3 & 1/3 \\ 2/3 & 1/3 & -2/3 \\ 1/3 & 2/3 & 2/3 \end{bmatrix}=\begin{bmatrix} -2 & 1 & 2 \\ 6 & 6 & 3 \end{bmatrix}$$

4 Performance Evaluation

Different parameter values for the threshold scheme (i.e. the values of *n*, *m* and *p*) create a large class of schemes. Each scheme has different property, and the best scheme needs different parameter values according to the given condition. In order to select an optimal scheme, we need to be able to evaluate and compare the performance of different schemes. In quantifying the schemes, the primary metric is availability.

Availability is defined as the probability that a file can be accessed at any given time. With threshold schemes, files are encoded into *n* shares, of which *m* or more are sufficient to fully reconstruct the file. We assume the failures of storage nodes are independent. The general availability read and write model are

$$\begin{aligned} \text{Availability}_{\text{read}} &= \left(\sum_{i=m}^{n}\binom{n}{i}\times p_a^i\times(1-p_a)^{n-i}\right)^m \\ \text{Availability}_{\text{write}} &= \left(\sum_{i=m}^{N}\binom{N}{i}\times p_a^i\times(1-p_a)^{N-i}\right)^m \end{aligned} \tag{14}$$

Here P_a is the node availability, and it is assumed to be 0.9. Also, $N = n+1$. Figure 1 and 2 show the comparison of read and write availability of the proposed scheme along with striping and information dispersal scheme. Here up to 100 nodes were tested. Note that the m value for striping, information dispersal, and proposed scheme are n, 5, 6, respectively. The figures reveal that the proposed scheme display substantially higher availability than striping while it is consistently better than the information dispersal scheme for about 10% for both read and write operation.

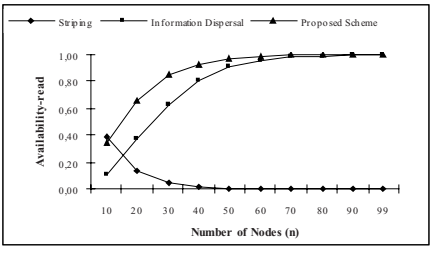

Fig. 1. Comparison of read availabilities.

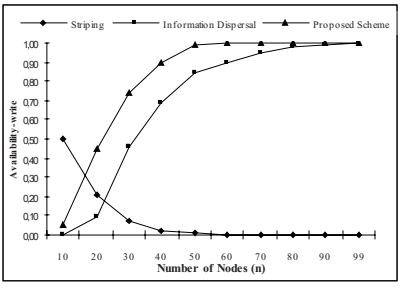

Fig. 2. Comparison of write availabilities.

Figure 3 shows that how availability of the proposed scheme varies as *m* and *n* value change. Form the figure we can see that availability gets higher as *m* decreases and *n* increases.

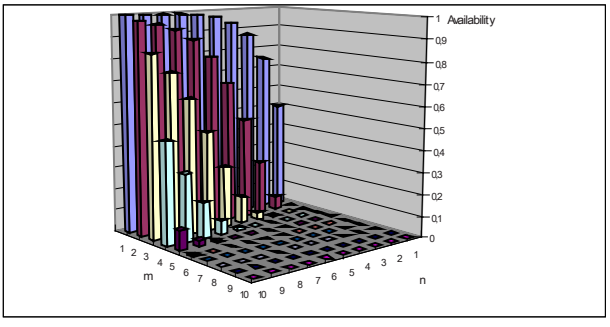

Fig. 3. The availability as *m* changes from 1 to 10

5 Conclusion and Future Work

Most earlier schemes apply data encryption and distribution separately, which result in some vulnerability and inefficiency. In this paper thus we propose a new approach which integrates data encryption and distribution process into one process to improve both the security and availability by using Singular Values Decomposition theorem. As a result, the proposed scheme offers high availability and security as well as complete recovery of original data even though partial damage occurs to the data. The proposed scheme can handle data represented by any size matrix, and eigenvalues are guaranteed to be found. Analysis shows that it improves the availability about 10% compared with information dispersal scheme for both read and write operation, while it allows secure storage since decryption without the key is absolutely impossible. A new model considering not only availability but also security in a more formal way will be developed.

References

1. Mehmet Bakkaloglu, Jay J. Wylie, Chenxi Wang, Gregory R. Ganger.: On Correlated Failures in Survivable Storage Systems: School of Computer Science Carnegie Mellon University, Pittsburgh, PA15213 (2002)
2. Jay J. Wylie, Michael W. Bigrigg, John D. Strunk, Gregory R. Ganger, Han Kiliccote, Pradeep K. khosla.: Survivable Information Storage systems: IEEE Computer (2000)
3. Sung Jin Choi, Hee Yong Youn, Bo Kyung Lee.: An Efficient Dispersal and Encryption Scheme for Secure Distributed Information Storage: ICCS2003, LNCS2660, Springer (2003) 958-967

4. A. De Santis and B. Masucci.: Multiple Ramp Schemes: IEEE Trans. Information Theory (1999) 1720-1728
5. R. Cannetti, R. Gennaro, S. Jarecki, H. Krawcxyk and T. Rabin.: Adaptive Security for Threshold Cryptosystems: In Advances in Cryptology-Crypto '99, LNCS, Springer (1999) 98-115
6. George Nakos, David Joyner.: Linear Algebra with Applications, Brooks/Cole USA (1998) 562-569