Singular Values Decomposition 을 이용한 분산/암호화 기법

최성진*, 윤희용*
*성균관대학교 정보통신공학부
e-mail : *{choisj, youn}@ece.skku.ac.kr

A Dispersal/Encryption Schemes using Singular Values Decomposition

Sung-Jin Choi*, Hee-Yong Youn*
* School of Information and Communications Engineering
SungKyunKwan University

요 약

오늘날 컴퓨터 네트워크 기술의 급속한 발전은 네트워크를 이용한 서비스를 다양하게 하였고, 많은 정보를 생산하게 되었다. 이에 따라 저장장치의 생존성(survivability)은 가장 중요한 사항으로 고려되고 있으며, 이러한 생존성을 높이기 위하여 새로운 분산저장기법의 연구개발이 절실히 필요한 실정이다. 따라서, 본 논문에서는 분산저장시스템의 생존성을 높이기 위해 필수적으로 필요한 새로운 분산/암호화 기법을 제안하고, 제안된 기법의 가용성을 평가한다. 제안된 기법은 데이터의 분할과 암호화를 동시에 허락하여 보안성을 높임과 동시에 기존의 기법과 비교하여 10%정도의 가용성 향상을 보인다.

1. 서론

오늘날 컴퓨터 네트워크 기술의 급속한 발전은 네트워크를 이용한 서비스를 다양하게 하였고, 많은 정보를 생산하게 되었다. 이에 따라 대용량 정보의 발생은 필수 불가결하게 되었고 이를 효율적으로 관리해야 하는 문제도 발생하게 되었다. 특히 기업의 데이터베이스 규모가 커지게 되었고, 각종 업무의 자료를 디지털화하여 보관하는 것이 일반화 되어있는 실정이다. 이런 환경에서 분산저장시스템의 고장이나 오류, 정보의 누출, 임의 변경 등은 기업 업무 마비와 경제적 손실은 물론 기업의 생존까지도 위협 받을 수 있기 때문에 미연에 방지해야 한다. 또한, 범국가적인 정보저장시스템의 구축이나 전자 상거래의 활성화를 위해서 분산저장시스템의 생존성(survivability)은 가장 중요한 사항으로 고려되고 있다 [1].

이러한 분산저장시스템의 생존성을 높이기 위해서는 분산저장기법을 통하여 저장시스템의 가용성을 높이는 것이 선행되어야 하며, 이를 위한 새로운 분산 및 데이터 보호기법의 연구개발이 절실히 필요하다.

분산저장시스템에서는 정보를 분산된 각 저장노드 단위로 저장한다. 이 때 데이터의 가용성은 데이터를 분산시키는 기법 및 정책에 따라 크게 변하게 되는데, 즉, 분산하는 알고리즘에 따라 거의 대부분의 데이터를 잃어도 완벽한 복구가 가능한 반면, 특정 수 이상의 데이터를 잃으면 복구가 불가능한 경우도 있다. 특히, 정보를 단순 분할 저장하거나 보안성을 높이기 위해 분할 저장하는 두 경우 모두 부분적인 데이터의 손실 및 파괴가 전체 데이터의 손실로 이어지는 경우가 발생하게 된다. 따라서, 본 논문에서는 분산정보저장시스템의 생존성을 높이기 위해 필수적으로 필요한 행렬 분해를 이용한 새로운 데이터 분산/암호화기법을 제안하고, 제안된 기법의 가용성을 평가한다. 제안된 기법은 데이터의 분할과 암호화를 동시에 허락하여 보안성을 높임과 동시에 기존의 기법과 비교하여 10%정도의 가용성 향상을 보인다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존에 대표적으로 사용되는 threshold 기법을 이용한 데이터 분산기법에 대하여 알아보고, 3 장에서는 본 논문에서 제안하는 분산/암호화기법을 소개하고, 제안된 기법의

수식적 해석을 통해 가용성을 평가하고자 한다. 마지막으로 4 장에서는 결론 및 향후 연구과제를 제시한다.

2. 관련 연구

안전한 분산저장시스템에서 일반적인 threshold 기법에 의거한 데이터 분산기법은 크게 복제(Replication), 스트라이핑 (Striping), 정보 분산 (Information Dispersal), 스플리팅(Splitting)등으로 나뉘어 진다.

복제는 한 개의 데이터를 n 개의 완전한 복사본으로 만들어 분산 저장하는 기법으로 전체 노드 중 단지 하나만 살아 남아도 원래의 데이터를 완전하게 복구 시킬 수 있다. 이 기법의 단점은 침입자가 단지 하나의 노드에 접근하는 것만으로 모든 자료가 누출된다는 것이다. 스트라이핑은 하나의 데이터를 n 개의 조각으로 분할시켜서 저장하는 기법으로 1 개의 데이터를 저장하기 위해 1 개의 저장 공간만을 이용한다. 이 기법의 단점은 단 하나의 노드 손실만으로도 원래의 데이터를 완벽하게 복구시키지 못한다는 점이다. 정보 분산은 스트라이핑과 같이 데이터를 분할한 후에 복사본을 만들어 분산 저장하는 기법으로 일부 노드에 손실이 생겨도 원래의 데이터를 복구할 수 있다. 이 기법의 단점은 침입자가 일부분의 데이터를 얻더라도 완전하지는 않지만 어느 정도의 의미 있는 정보를 알 수 있다는 것이다. 스플리팅은 n-1 개의 저장 노드에는 임의의 난수를 발생시켜 저장하고 나머지 하나의 노드에는 발생시킨 n-1 개의 난수 들과 원래 데이터 값을 XOR 시켜 저장하는 기법으로 하나의 노드에 대한 침입은 전혀 의미가 없다. 이 기법의 단점은 하나의 노드 손실만으로도 원래 데이터의 복구가 불가능하다는 점이다 [2].

현재 분산 정보 저장 시스템을 위한 대표적인 분산 기법은 램프(Ramp) 기법과 Blakley 의 보안 분배가 있다 [3]. 램프 기법은 p-m-n threshold 기법으로 나타내는 여러 분산 기법들을 포괄적으로 적용 가능하게 하는 기법으로, p 는 최소로 유효한 단위 데이터의 묶음의 갯수를, m 은 원본 데이터로의 완전한 복구를 위해서 최소로 필요한 분할 단위 데이터 묶음의 갯수를, n 은 저장 되는 저장장치 노드의 갯수를 의미한다 [4]. Blakley 의 보안 분배기법은 정보를 m 차원의 공간에서의 한 점으로 나타내는데, m-1 차 다항식 m 개 이상의 교점으로써 표현한다. 각 저장 노드에는 하나의 다항식에 대한 정보만 있으므로 정보의 누출이 어렵게 되고, 정보의 재구성을 위해서는 최소한 m-1 개의 노드가 살아있기만 하면 된다 [5].

3. 제안하는 행렬의 Singular Value Decomposition 을 이용한 분산/암호화 기법

3.1 Singular Value Decomposition 의 정리와 고유값과 고유벡터의 정의

□ 정리 1

A를 $m \times n$ 행렬이라고 하고, $\sigma_1,...,\sigma_r$ 를 0 이 아닌 singular values라고 하면, 다음의 식을 만족하는 직교(orthogonal) 행렬 $U(m\times r), V(r\times r), D(r\times n)$가 존재한다.

$$A = UDV^T$$

□ 증명 1

U, V, D 는 명확하게 정의 되어 있고 U 와 V 는 직교행렬이기 때문에, 따라서 $A = UDV^T$ 만 증명하면 된다.

$x^TA^TAx = \|Ax\|^2 \geq 0$ 이기 때문에 행렬 A^TA 는 모든 $x \in R^n$ 에 대하여 양반정치행렬이다. 따라서 A^TA 는 항상 음수가 아닌 고유값을 갖는다. 이 고유값을

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$$

처럼 순서를 정하고 $\sigma_i = \sqrt{\lambda_i}$, i=1,...,n 라고 정의하자. 일반성을 잃지 않고 σ_i 들 중 정확하게 r 개가 0 이 아니라고 가정하면

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,\ \sigma_{r+1} = \sigma_{r+1} = \cdots = \sigma_n = 0$$

처럼 나타낼 수 있다.

$$D = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & & & 0 \\ 0 & & \sigma_r & & 0 \\ \vdots & & & 0 & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix}$$

라고 하자. 여기서, D_r 은 대각선 요소가 $\sigma_1,...,\sigma_r$ 인 $r\times r$ 대각행렬이다.

A^TA 가 대칭행렬이므로 A^TA 를 대각화하는 직교행렬 V 가 다음과 같이 존재한다.

$$V^TA^TAV = D^TD$$

$v_1,...,v_n$ 을 V 의 열벡터들이라고 하자. 그러면 v_i 는 i=1,...,r 에서 σ_i^2 에 대응되는 고유벡터이고, $v_{r+1},...,v_n$ 들은 0 에 대응되는 고유벡터들이다.

$V_1 = (v_1, v_2, \cdots, v_r)$, $V_2 = (v_{r+1}, \cdots, v_n)$ 라고 하자. i=r+1,...,n 에서 $A^TAv_i = 0$ 이므로,

$(AV_2)^TAV_2 = V_2^TA^TAV_2 = 0$ 이고, $AV_2 = 0$ 이다. 따라서 $A^TAV_1 = V_1D_r^2$ 이므로 $D_r^{-1}V_1^TA^TAV_1D_r^{-1} = I_r$ 이다. 여기서 I_r 는 $r\times r$ 단위행렬이다.

$U_1 = AV_1D_r^{-1}$ 라고 하면 $U_1^TU_1 = I_r$ 이고, U_1 은 직교 열벡터 $u_1,...,u_r$ 을 갖는 $m \times r$ 행렬이다. 집합 $\{u_1,...,u_r\}$ 은 R^m 을 위한 정규직교기저를 이루도록 확장할 수 있다. $U_2 = (u_{r+1},...,u_m)$ 라 하고, $U = [U_1\ U_2]$ 라고 하자. 그러면

$$U^TAV = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} A \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \begin{bmatrix} AV_1 & 0 \end{bmatrix} = \begin{bmatrix} U_1^TAV_1 & 0 \\ U_2^TAV_1 & 0 \end{bmatrix}$$

이다. 그러나 $U_1^TAV_1 = D_r^{-1}V_1^TA^TAV_1 = D_r$ 이고, $U_2^TAV_1$

$= U_2^T U_1 D = 0$ 이므로 $U^T AV = D$ 이고, $A = UDV^T$ 이다 [6].

□ 정의 1

A를 n×n 행렬이라고 하면, 스칼라 λ 에 대해서 $Av=\lambda v$ 식을 만족하는 0이 아닌 벡터 v를 고유벡터라고 한다. 이때 스칼라 λ 를 고유벡터에 대응하는 고유값이라고 한다.

λ를 A의 고유값이라 하면, $Ax=\lambda x \Leftrightarrow Ax=\lambda I_n x \Leftrightarrow (\lambda I_n - A)x=0$ 이 된다. $x \neq 0$ 이므로 동차연립방정식 $(\lambda I_n - A)x=0$ 은 0이 아닌 해를 가져야 한다. 따라서 $\det(\lambda I_n - A)=0$ 에서 나오는 특성 방정식을 풀면 고유값 λ 를 구할 수 있다 [7].

3.2 분산/암호화 기법

우리가 제안하는 Singular Values Decomposition 을 이용한 분산/암호화 기법에 대해서 설명하겠다. 우선, 데이터를 분산/암호화 하기 위해서 원본 데이터를 행렬로 변환한다. A 가 원본 데이터로부터 변환된 행렬이라고 하면, A 의 전치(Transpose) 행렬인 A^T 를 구한 후 A^TA 를 계산한 다음, 정의 1 을 이용하여 λ 값을 구한다. 이 $\lambda(\lambda_1,\lambda_2,...,\lambda_n)$ 를 고유값이라고 하고, 각 λ 에 대응하는 고유벡터를 다음과 같이 표현한다.

$$v_1=\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ \vdots \\ s_n \end{bmatrix} s_n \in R,\ v_2=\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ \vdots \\ t_n \end{bmatrix} t_n \in R,\ \cdots\cdots v_n=\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_n \end{bmatrix} z_n \in R$$

위와 같이 나온 n개의 고유벡터가 실제로 저장 되는 데이터(V)이고, 대각선 성분이 σ_i 인 행렬이 복호화 키(D)가 된다. 그리고 또 다른 복호화 키(U)는 다음 식에 의해 구할 수 있다.

$$U_i = \frac{1}{\sigma_i} A v_i ,\quad i=1,...,r$$

이 기법의 특징은 데이터의 분할과 암호화가 동시에 된다는 것으로서 일반적인 threshold 기법의 비밀분산 조건을 만족하고 있다. 즉, 고유벡터(분할된 데이터)를 가지고서 원래의 행렬(데이터)을 찾는 것은 잘 알려진 NP Hard 문제이기 때문에 분할 저장된 데이터 전체를 얻게 되어도 원래의 데이터를 알아 낼 수가 없다. 그리고 각 고유벡터들의 원소들은 실수에 대응하는 수의 비로 이루어졌기 때문에, 무한대의 조합으로 표현될 수 있다. 따라서 데이터의 부분적인 손실이나 파괴가 일어난다고 해도 원래의 데이터를 완벽하게 복구 할 수 있을 뿐만 아니라 두 개의 복호화 키를 가지고 있기 때문에 높은 가용성과 보안성을 갖는다.

□ 예제 1

원래의 데이터를 $\begin{bmatrix} -2 & 1 & 2 \\ 6 & 6 & 3 \end{bmatrix}$ 라고 하면, 정의 1 과 정리 1 에 의해 고유값은 $\lambda_1=81,\ \lambda_2=9,\ \lambda_3=0$ 이고, 각 λ 에 대응하는 고유벡터는 다음과 같다. $\begin{bmatrix} 2s \\ 2s \\ s \end{bmatrix}, \begin{bmatrix} -2t \\ t \\ 2t \end{bmatrix}, \begin{bmatrix} u \\ -2u \\ 2u \end{bmatrix}$ s,t,u∈R . 따라서, s,t,u∈R 이기 때문에 실제로 저장되는 데이터를 노드 1, 노드 2 그리고 노드 3 에 분산/암호화하여 저장하고, 복호화 키를 생성하면 다음과 같이 나타낼 수 있다.

노드1= $\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 6 \\ 6 \\ 3 \end{bmatrix}$ ...etc, 노드2= $\begin{bmatrix} -2 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} -4 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} -6 \\ 4 \\ 6 \end{bmatrix}$ ...etc

노드3= $\begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ -4 \\ 4 \end{bmatrix}, \begin{bmatrix} 3 \\ -6 \\ 6 \end{bmatrix}$ ...etc

복호화키: $D=\begin{bmatrix} 9 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$, $U=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

3.3 Gram-Schmidt 의 정규직교화(orthonormal) 과정을 이용한 데이터의 재 복구

우리가 제안한 Singular Values Decomposition 을 이용한 분산/암호화 기법에 의해 분산 저장된 데이터를 재 복구하기 위해서는 다음에 나오는 정의와 정리를 살펴보아야 한다.

□ 정의 2

R^n 의 벡터 $x_1, x_2, ..., x_k$ 에 대하여 $S=\{x_1, x_2, ..., x_k\}$ 라 하자. 이때, S 의 서로 다른 어느 두 벡터도 모두 직교하면 S 를 직교집합(orthogonal)이라 한다. 특히, 직교집합 S 에 속하는 벡터가 모두 단위 벡터일 때 S 를 정규직교집합(orthonormal set)이라고 한다. 즉 0 아닌 벡터들의 직교집합을 표준화하면 정규직교 집합을 얻을 수 있다 [8].

$$\{v_1,...,v_k\}: \text{직교집합} \Rightarrow \left\{\frac{v_1}{\|v_1\|},...,\frac{v_k}{\|v_k\|}\right\}: \text{정규직교집합}$$

3.2 장에서는 Singular Values Decomposition 을 이용한 분산/암호화 기법에 대해서 살펴 보았다. 다음에는 정의 2 를 이용하여 원래의 데이터를 복구하는 방법을 살펴 본다.

데이터의 재 복구를 위해서는 행렬 V(실제로 저장되는 데이터)의 집합들을 정의 2 를 이용하여 정규직교집합으로 만들어 전치 행렬 V^T 를 만든다. 그리고 복호화 키 U, D 와 V^T 를 이용하여, 증명 1 의 마지막에 나와있는 식 $UDV^T = A$ 에 의해 원래의 데이터 A 를 완전하게 복구 할 수 있다.

□ 예제 2

예제 1 에서 노드 1, 노드 2 그리고 노드 3 의 첫 번째, 두 번째 세 번째 데이터를 각각 읽어 들여 V 를 만든 후 정규직교화 과정을 이용하여 V^T 를 구한다. 그리고 복호화 키 U, D 와 V^T 를 $UDV^T = A$ 에 대입하면 다음과 같이 원래의 데이터를 구할 수 있다.

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

3.4 가용성(Availability) 평가

가용성이란 자료를 저장해 놓은 저장노드 중 일부에 이상이 생겼을 때 (특정 노드지역(Pool)의 정전, 디스크의 파괴, 시스템의 다운 등과 같은 경우) 남아 있는 노드만을 이용해서도 원래의 자료를 복구해낼 수 있는 특성을 말한다.

다음은 가용성 평가를 위해 사용된 식으로 m 은 원본데이터로의 완전한 복구를 위해서 최소로 필요한 분할 단위 데이터 묶음의 개수를 나타내며, n 은 저장될 저장장치 노드의 개수를 나타낸다. 마지막으로 Avail 은 각 노드가 가용할 확률을 나타낸다 [9].

$$\text{Availability} = \left(\sum_{i=m}^{n}\binom{n}{i}\times \text{Avail}^{i}\times(1-\text{Avail})^{n-i}\right)^{m}$$

그림 1 은 스트라이핑, 정보분산 기법의 가용성과 제안된 분산/암호화 기법의 가용성을 비교한 것이다.

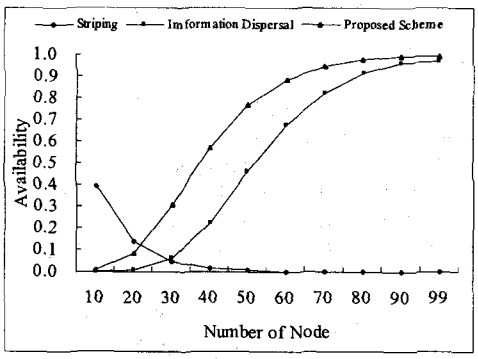

그림 1. 가용성 평가.

그림 1 의 그래프를 보면 제안된 분산/암호화 기법이 기존의 정보분산 기법과 비교하여 가용성에 있어서 대략 10%정도의 성능 향상을 보이고 있는 사실을 알 수 있다.

그림 2 는 가용성이 m 에 따라 변화하는 상태를 보여주는데, m 이 작을수록 가용성이 높아진다는 사실을 알 수 있다. 따라서 m 의 값이 작으면 사고가 일어나도 적은 수의 조각만으로도 원래 데이터를 재구성할 수 있다는 장점이 있다. 반대로 m 의 값이 증가하면, 침입자를 찾아내기가 용이하고 데이터의 중복이 줄게 되어 전체 저장장치 사용량이 줄어들게 되지만, 데이터를 읽고 쓰는데 필요로 하는 CPU, 네트워크 같은 리소스의 양이 증대하게 되고, 만일 사고가 일어났을 경우 살아남은 노드만으로는 전체를 복구할 수 없는 경우가 일어나게 된다. 따라서, 응용/정책에 따라 시스템의 목적에 적합한 최적의 파라미터를 절충하여 결정하는 것이 중요하다.

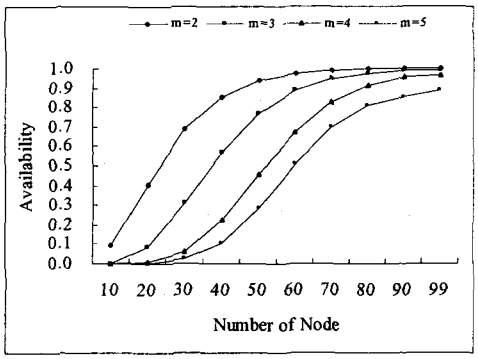

그림 2. m 에 대한 가용성 변화.

4. 결론 및 향후 과제

본 논문에서 제안한 분산/암호화 기법은 데이터의 분할과 암호화를 동시에 하고, 기존의 기법과 비교하여 가용성 측면에서 대략 10%정도의 가용성 향상을 보인다.

저장장치에 대한 의존도가 높아지면서 안전한 데이터 보관이 가능한 분산저장시스템의 필요가 부각되고 있는 이 시점에서, 새로운 분산 기법과 데이터 보호기법을 위해 많은 노력이 필요할 것으로 본다. 따라서, 향후 연구 과제로 이미 제안된 분산기법보다 가용성과 보안성이 높은 분산/암호화 기법을 개발하는 것이 목표이다.

참고문헌

[1] Jay J. Wylie, Michael W. Bigrigg, John D. Strunk, Gregory R. Ganger, Han Kiliccote, Pradeep K. khosla, "Survivable Information Storage systems", IEEE Computer , 2000.

[2] R. Cannetti, R. Gennaro, S. Jarecki, H. Krawcxyk and T. Rabin, "Adaptive Security for Threshold Cryptosystems", In Advances in Cryptology-Crypto '99, LNCS, Spriger pp.98-115, 1999.

[3] De Santis, A, Masucci, B, "Multiple Ramp Schemes," Information Theory, IEEE Transactions on, pp. 1720-1728, July 1999.

[4] G. R. Blakley and C. Meadows, "Security of Ramp Schemes,"in advances in Cryptology'Lecture Notes in Computer Science, Berlin, 1985.

[5] E. Karnin, J. Greene, M. Hellman, "On Secret Sharing Systems", IEEE Trans. Information Theory pp.35-41, 1983.

[6] George Nakos, David Joyner, "Linear Algebra with Applications", pp. 562-569, 1998.

[7] Birkhauser, "Linear Algebra", pp. 209-213, 1997.

[8] David Kincaid, Ward Cheney, "Numerical Analysis ", pp. 294-295, 1996.

[9] Jay J. Wylie, Mehmet Bakkaloglu, Vijay Pandurangan, Michael W. Bigrigg, Semih Oguz, "Selecting the Right Data Distribution Scheme for A survivable Storage System", CMU-CS-01-120, 2001.