LDU 분해를 이용한 데이터 암호화 기법에 관한 연구

최성진* 윤희용* 최중섭** 이강신**
*성균관대학교 정보통신공학부
**한국정보보호진흥원 기반시설보호단/기반보호기술팀
*{choisj, youn}@ece.skku.ac.kr
**{jschoi, kslee}@kisa.or.kr

The Study on Data Encryption Schemes Using LDU Decomposition

*Sung Jin Choi, *Hee Yong Youn, **Joong Sup Choi, **Kang Shin Lee
*School of Information and Communications Engineering, SungKyunKwan University
**Critical Information Infrastructure Protection Division Infrastructure Protection Technology, KISA(Korea Information Security Agency)

요 약

저장장치의 발전과 인터넷 사용량의 증가, 전자 상거래의 활성화에 의해 많은 사람들이 디지털 정보를 편리하게 이용할 수 있게 되었다. 이에 따라 저장장치의 보안성과 생존성은 가장 중요한 사항으로 고려되고 있으며, 이러한 보안성과 생존성을 높이기 위하여 새로운 분산저장기법의 연구개발이 절실히 필요한 실정이다. 따라서, 본 논문에서는 분산저장시스템의 보안성과 생존성을 높이기 위해 필수적으로 필요한 분산/암호화 기법을 LDU 분해를 이용하여 제안하고, 제안된 기법의 가용성을 평가한다. 제안된 기법은 데이터의 분할과 암호화를 동시에 허락하여 보안성을 높임과 동시에 기존의 기법과 비교하여 10%정도의 가용성 향상을 보인다.

1. 서 론

저장장치의 발전과 인터넷 사용량의 증가, 전자 상거래의 활성화에 의해 많은 사람들이 디지털 정보를 편리하게 이용할 수 있게 되었다. 이에 따라 저장장치 의존도는 매우 높아지게 되었고, 저장장치의 보안성과 생존성에 대한 요구사항 또한 매우 높아져가고 있는 실정이다. 따라서 기업정보는 물론 개인 정보를 보호해야 하고 예기치 못한 상황에서도 데이터를 안전하게 읽고 쓸 수 있는 생존성 높은 분산저장시스템이 필요하게 되었다 [1].

이러한 분산저장시스템의 생존성을 높이기 위해서는 분산저장기법을 통하여 저장시스템의 가용성을 높이는 것이 선행되어야 하며, 이를 위한 새로운 분산 및 데이터 보호기법의 연구개발이 절실히 필요하다. 따라서 본 논문에서는 분산저장 시스템의 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 복구시킬 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 [2-4].

3. 제안하는 행렬 분해를 이용한 분산/암호화 기법

3.1 행렬 분해의 정의

• 정의 1: A 가 가우스 소거법에 의해 m×n 행 사다리꼴 형태 U 로 만들 수 있는 m×n 행렬이라고 하자. 그러면 행렬 A 는 LU Decomposition 을 가졌다고 하며, 다음과 같이 L 과 U 행렬로 분해할 수 있다.

$$A = LU$$

이때, L 을 주 대각 성분이 1 인 m×m 인 하 삼각행렬(Low Triangular)이라고 하며, L 행렬의 성분 l_{ij} 는 가우스 소거법에 의해 계산된다.

이제 열 교환이 필요 없는 비가역 정사각행렬 A 가 정의 1 를 만족한다고 하자. 그리고 U 의 대각선 성분을 피벗(pivot)시켜 모두 1 로 만든다. 따라서 0 이 아닌 피벗 d_i 로 U 의 각 i 번째 열을 나누면, 행렬 U 는 피벗 $d_1, d_2, ..., d_n$ 를 성분으로 하는 대각 행렬(Diagonal) D 와 대각성분을 1 로 가지는 새로운 상 삼각행렬(Upper Triangular)로 분해 할 수 있다. 이렇게 행렬 A 를 하 삼각행렬, 대각행렬 그리고 상 삼각행렬로 분해 할 수 있을 때, 행렬 A 는 LDU Decomposition 을 갖는다고 하며 다음과 같이 표현된다.

$$A = LDU$$

$$\text{where } U=E_n E_{n-1} L\ E_1 A,\ L=E_1^{-1}E_2^{-1}L\ E_n^{-1}$$

행렬 A 가 대칭이고, A=LU 로 유일하게 하삼각행렬 L 과 상삼각행렬 U 로 분해된다면 $LU=(LU)^T=U^TL^T$ 가 되어 $U=L^T$ 가 된다. 따라서 $A=LL^T$ 로 분해되어 저장공간을 줄일 수 있을 뿐만 아니라 시간 복잡도(Time-Complexity)도 줄일 수 있다. 이러한 분해가 가능한 행렬은 대칭 행렬이고 양의 정부호(positive definite)를 갖는 행렬이다. 이런 행렬 분해를 촐레스키분해(Cholesky-Decomposition)이라고 하며 다음과 같다.

• 정의 2: 대칭행렬 A 가 $x\neq 0$ 인 벡터에 대하여 $x^TAx>0$ 이면 양의 정부호를 갖는다고 하며, 양의 정부호를 갖는 행렬 A 가 다음과 같이 $A=LL^T$ 로 분해되면 이를 촐레스키분해라고 한다 [5-6].

3.2 분산 암호화 기법

우리가 제안하는 행렬분해를 이용한 분산/암호화 기법에 대해서 설명하겠다. 우선, 데이터를 분산/암호화 하기 위해서 원본 데이터를 행렬로 변환한다. A 가 원본 데이터로부터 변환된 행렬이라고 하면, 정의 1, 2 그리고 3 에 의해 각각 다음과 같이 나타낼 수 있다.

• LDU-Decomposition 을 이용한 분산/암호화 기법

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},\ L=\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

$$D=\begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & d_{nn} \end{bmatrix},\ U=\begin{bmatrix} 1 & u_{12}/d_1 & \cdots & u_{1n}/d_1 \\ 0 & 1 & u_{23}/d_{22} & u_{2n}/d_{22} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix}$$

위와 같이 나온 행렬 L과 U가 실제로 저장되는 데이터이고 행렬 D는 복호화 키가 된다.

가우스 소거법에 기반한 LDU-Decomposition의 시간 복잡도(Time-Complexity)를 계산하면 다음과 같다. 상삼각행렬을 만들기 위하여 고정된 j번째 열의 j+1번째 항부터 n항까지 모두 0으로 만들 때 필요한 곱셈의 수는(n-j+2)(n-j)이다. 따라서 시간 복잡도는 다음과 같다.

$$\begin{aligned}\sum_{j=1}^{n-1}(n-j+2)(n-j) &= \sum_{k=1}^{n-1}k(k+2) \\ &= \sum_{k=1}^{n-1}k^2+\sum_{k=1}^{n-1}k \\ &= \frac{n(n-1)(2n-1)}{6}+n(n-1) \\ &= \frac{n(n-1)(2n+5)}{6}\approx\frac{n^3}{3}\end{aligned}$$

• Cholesky-Decomposition 을 이용한 분산/암호화 기법

$$A=LL^T=\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}\begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{n2} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & l_{nn} \end{bmatrix}$$

위와 같이 나온 행렬 L 이 실제로 저장되는 데이터이며, 시간 복잡도는 다음과 같다.

$$\sum_{i=1}^{n}(i+\sum_{j=1}^{i-1}j)=\frac{1}{2}\sum_{i=1}^{n}(i^2+i)=\frac{1}{6}n^3+O(n^2)\approx\frac{n^3}{6}$$

따라서, 가우스 소거법에 기반한 LDU-Decomposition 에 비해 시간복잡도가 약 2 배 정도 빠른 것을 알 수 있으며, 또한 실제로 저장되는 데이터가 대칭이므로 저장 공간도 줄일 수 있다는 장점이 있다. 하지만 복호화 키가 따로 존재하지 않는다는 단점도 있다.

LDU-Decomposition 을 이용한 기법의 특징은 데이터의 분할과 암호화가 동시에 된다는 것으로서 일반적인 threshold 기법의 비밀 분산 조건을 만족하고 있다. 즉, 분할된 행렬 L 과 U 를 가지고서 원래의 행렬(데이터)를 찾는 것은 잘 알려진 NP Hard 문제 이기 때문에 분할 저장된 데이터 전체를 얻게 되어도 복호화 키(행렬 D)가 없다면 원래의 데이터를 알아 낼 수 가 없다. 그리고 Cholesky-Decomposition 을 이용한 기법의 특징은 데이터의 분할과 암호화가 동시에 된다는 것으로서 LDU-Decomposition 을 이용한 기법과 똑같지만, 분할된 행렬 L 전체를 얻게 되면 손쉽게 원래의 데이터를 구할 수 있다는 단점이 있다. 하지만, 전 기법에 비해 저장 공간을 줄일 수 있다는 특징과 시간 복잡도가 약 2 배 이상 빠르다는 특징을 가지고 있다.

3.3 데이터의 재복구

우리가 제안한 LDU-Decomposition 과 Cholesky-Decomposition 을 이용한 분산/암호화 기법에 의해 분산 저장된 데이터를 재 복구하기 위해서는 3.2 에 나오는 행렬식을 살펴보아야 한다.

첫 번째 기법에서는 저장된 행렬 L, U (데이터)와 복호화 키(D)를 이용하여 계산 하면 다음과 같이 원래의 데이터를 복구 할 수 있다.

$$a_{ij} = l_{ij} + \sum_{k=1}^{i-1} l_{ik}u_{kj},\ j \le i,\ i=1, 2, \cdots, n,\ l_{ii} = a_{ii}$$

$$a_{ij} = u_{ij}l_{ii} + \sum_{k=1}^{i-1} l_{ik}u_{kj},\ i \le j,\ j=2, 3, \cdots, n$$

두 번째 기법에서는 저장된 행렬 L(데이터)를 읽어들여 대칭행렬(L^T)로 만든 후 계산 하면 다음과 같이 원래의 데이터를 복구 할 수 있다.

$$a_{11} = l_{11}^2$$

$$a_{21} = l_{21}l_{11},\quad a_{22} = l_{21}^2 + l_{22}^2$$

$$a_{31} = l_{21}l_{11},\quad a_{32} = l_{31}l_{21} + l_{32}l_{22},\quad a_{33} = l_{31}^2 + l_{32}^2 + l_{33}^2$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$a_{n1} = l_{n1}l_{11},\quad a_{nj} = \sum_{k=1}^{j} l_{nk}\sum_{i=1}^{j} l_{ji},\quad a_{nn} = \sum_{k=1}^{n} l_{nk}^2$$

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) 남아있는 노드만을 이용해서도 원래의 자료를 복구해낼 수 있는 특성을 말한다 [7].

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.

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

그림 1 은 정보분산 기법의 가용성과 제안된 두 기법의 가용성을 비교한 것이다.

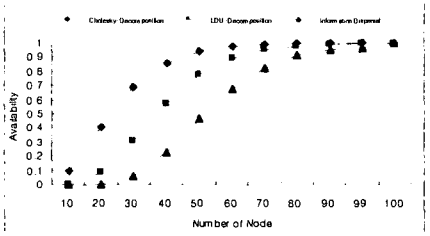

그림 1. 가용성 평가

그림 1 의 그래프를 보면 제안된 LDU-Decomposition 을 이용한 기법이 기존의 정보 분산 기법과 비교하여 가용성에 있어서 대략 10%정도의 가용성 향상을 보이고 있다는 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 결정하는 것이 중요하다.

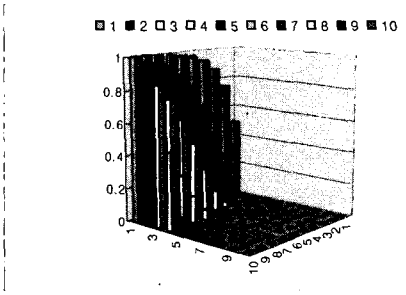

그림 2. m 에 대한 가용성 변화

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 보호 기법을 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] David Kincaid Ward Cheney, "Numerical Analysis Mathematics of Scientific Computing", Brooks/Cole Publishing Company.

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

[7] 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.