DOI: 10.3745/KIPSTC.2010.17C.5.399

t-(v,k,1) 조합 디자인 기반의 데이터 분산 관리 방식

송 유 진[†] · 박 광 용[††] · 강 연 정[†††]

요 약

유비쿼터스 네트워크 환경에서 다양한 데이터 서비스가 가능해지면서 악의적인 공격자나 내부 사용자에 의한 보안 취약성 및 프라이버시 침해로 인한 문제를 해결하기 위해 다양한 콘텐츠나 대용량 데이터의 안전한 보안 관리 문제가 주요 이슈로 떠오르고 있다. 기존 다항식 기반을 이용한 Ito, Saito, Nishizeki 할당 방식은 분산된 비밀정보를 복원하기 위해 Share 모두가 필요하게 된다. 반면, 본 제안 방식의 경우, 임계치 이상의 Share가 모아지면 비밀정보를 복원할 수 있다. 또한, 데이터를 분산· 복원하는 분산 DBMS 운영상의 효과 특히, 데이터베이스(DB) 서버 및 Share 선택의 규칙성이 있고 조합적 구조인 Combinatorial Design상의 파라메터 t, v, k를 이용함으로써 구현상의 유연성(flexibility)을 갖는다. 본 논문에서는 Share의 할당 문제를 해결하고자 비밀분산 방식을 이용하여 데이터를 분산저장관리 할 때 Share의 할당을 위해 $t-(v,k,1)$ 디자인이 갖는 행렬구조를 적용시킴으로써 새롭게 Share 할당법을 구성하고 데이터 분산저장관리에의 응용에 대해 검토한다.

키워드 : $t-(v,k,1)$ 조합디자인, Share 할당, 분산저장관리

Distributed Data Management based on t-(v,k,1) Combinatorial Design

Youjin Song[†] · Kwangyong Park[††] · Yeonjung Kang[†††]

ABSTRACT

Many problems are arisen due to the weakness in the security and invasion to privacy by malicious attacker or internal users while various data services are available in ubiquitous network environment. The matter of controlling security for various contents and large capacity of data has appeared as an important issue to solve this problem. The allocation methods of Ito, Saito and Nishizeki based on traditional polynomial require all shares to restore the secret information shared. On the contrary, the secret information can be restored if the shares beyond the threshold value is collected. In addition, it has the effect of distributed DBMS operation which distributes and restores the data, especially the flexibility in realization by using parameters t, v, k in combinatorial design which has regularity in DB server and share selection. This paper discuss the construction of new share allocation method and data distribution/storage management with the application of matrix structure of $t-(v,k,1)$ design for allocating share when using secret sharing in management scheme to solve the matter of allocating share.

Keywords : $t-(v,k,1)$ Design, Share Allocation, Distribution/Storage Management

1. 서 론

최근 네트워크 서비스 환경이 클라우드 서비스 환경[9]으로 진화됨에 따라 많은 대용량 데이터들이 다양한 유통채널을 통해 생성되고 있으며 각종 휴대전화나 정보기기 등을 통해 디지털 콘텐츠가 다양화, 대용량화되고 있다. 또한, 네트워크 서비스의 다양화에 따라 일반인들이 일상생활에서 유비쿼터스 네트워크와 연결하여 여러 가지 정보들을 교환, 공유하고 있다.

이와 같이 서비스 환경의 진화로 인해 네트워크 접속형 스토리지의 이용이 증가하고, 동시에 단말 스토리지의 대용량화에 따라 네트워크상의 여러 장소에서 데이터를 저장 관리할 수 있는 안전하고 효율적인 스토리지 환경이 필요하게 되었다[10].

그러나 다양한 데이터 서비스가 가능해지면서 악의적인 공격자나 내부 사용자에 의한 보안 취약성 및 프라이버시 침해 문제를 해결하기 위해 다양한 콘텐츠나 대용량 데이터의 안전한 보안 관리 문제가 주요 이슈로 떠오르고 있다.

본 논문에서는 유비쿼터스 분산 환경에의 응용을 위한 분산저장관리 관점에서 비밀분산방식을 기반으로 Share 할당

※ 본 연구는 지식경제부의 지원을 받는 정보통신표준기술력향상사업의 연구결과로 수행되었음.
† 정 회 원 : 동국대학교 정보경영학과 교수
†† 준 회 원 : 동국대학교 전자상거래협동 석사과정
††† 정 회 원 : 한국인터넷진흥원 인터넷융합 · 정책본부 주임연구원
논문접수 : 2010년 4월 6일
수 정 일 : 1차 2010년 7월 1일
심사완료 : 2010년 2010년 8월 6일

방법에 대해 논한다. 비밀분산 방식에는 Shamir[6], 온라인 SS[8], XOR 비밀분산[7] 방식 등이 있다. 각 방식에서는 원 데이터를 분산할 때, 자체로는 원 데이터를 복원할 수 없는 Share(분산정보)가 발생하게 된다. 데이터의 용량이 실시간으로 증가하는 유비쿼터스 환경에서 분산정보를 안전하고 효율적으로 관리할 수 있는 방법이 필요하다. 즉, Share의 효율적인 할당방법이 요구된다.

본 논문에서는 Share의 할당 문제를 해결하고자 비밀분산 방식[6, 7]을 이용하여 데이터를 분산저장관리 할 때 Share의 할당을 위해 $t-(v,k,1)$ 디자인이 갖는 행렬구조를 적용시킴으로써 새롭게 Share 할당법을 구성한다. 그리고 구성된 Share할당법을 데이터 분산저장관리에 응용한다. 이때, 신뢰성 있는 기관에 의해 비밀정보가 분산된다는 가정, 예를 들면 Shamir 방식에서는 비밀정보의 분배자(신뢰성 있는 기관 즉, Dealer)에 의해 비밀정보가 분산되는 것을 가정한다. 이러한 가정을 통해, 분산된 비밀정보들, 즉, Shares를 DB에 분산저장하는 경우, DB에의 Share 할당 방법에 초점을 맞춘다.

본 연구결과는 기밀성이 높은 의료 데이터, 고객의 개인정보를 포함하는 영업 비밀정보 등의 데이터를 효율적으로 분산 관리하는 기술로서 클라우드 서비스 환경에서 이용이 기대된다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 Share 할당방식과 Gridsharing Framework를 설명하고 3장에서는 조합디자인에 근거한 Share 할당 구조에 대해 분석한다. 4장에서는 제안된 할당방식을 데이터 분산 저장관리에 응용하며 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 Share 할당에 대한 기존의 구성법에 대하여 설명한다.

2.1 비밀분산 방식

비밀분산 방식은 어느 한 당사자만이 알고 있는 하나의 비밀정보를 여러조각으로 나누어 다른 사람에게 공평하게 분배하여 필요시 이를 재구성하여 사용하는 방법으로 1979년에 Blakely와 Shamir[6]에 의해 처음으로 제안되었다. 그 중 다항식 보간법을 사용하는 Shamir의 (k,n) 임계치 비밀분산 방식은 비밀정보를 n개의 Share로 분할하고 복원시 n개의 Share 중 임의의 k개가 모이면 원래의 비밀정보를 복원할 수 있는 방식이다. 이러한 비밀분산 방식의 개념이 제안된 이후 여러 가지 특징을 갖는 다양한 비밀분산 방식[6-8]들이 제안되었다.

예를 들면, 정보복원에 모든 Share가 필요한 경우, Share가 저장된 서버에 장애 등이 발생하면 원래의 비밀정보를 복원할 수 없게 된다. 그러나 임계치 방식은 서버의 장애 등으로 저장된 특정 Share가 사용불가능하게 되어도 임계치 이상의 Share가 모아진다면 비밀정보를 복원할 수 있는 장점이 있다. 또한, 서버 장애 등의 Fault가 발생할 경우 대책으로서 활용이 가능하다.

2.2 Ito, Saito, Nishizeki의 Share 할당 방식

Ito, Saito, Nishizeki[1]의 Share 할당 방식은 임계치 접근구조에서 Share를 할당하는 방식으로 r명의 참가자 집합 $\{P_1, P_2, \ldots, P_r\}$으로 구성된다. 각 참가자는 $(m+1)$개의 Share를 할당 받고, 비밀분산 방식[6]의 접근구조를 구성하기 위해 먼저 m명의 참가자로 가능한 모든 조합으로 구성된 q개의 그룹 B를 구성한다.

$$B=\{B_1, B_2, \ldots, B_q\},\ \text{여기서,}\ \left(q=\binom{r}{m}\right).$$

다음으로 (q,q) 임계치 비밀분산 방식으로 비밀정보 s을 분산한다. 이때, 각 그룹에 분산되는 Share 집합은 $\{s_1, s_2, \ldots, s_q\}$로 표시된다. 참가자 P_i에게 할당될 분산정보 집합은 함수 $g(i)=\{s_j | P_i \notin B_j, 1 \le j \le q\}$에 의해 할당된다. 각 참가자는 $\binom{r-1}{m}$ Share를 받고 각 Share는 $(r-m)$ 참가자에게 저장된다. 예를 들어, 분산된 비밀을 찾기 위해 적어도 3명의 참가자가 Share를 모아야하는 4명으로 구성된 참가자 집합을 생각해 보자. $r=4, m=2$인 참가자로 가능한 모든 조합 그룹 B는 다음과 같다.

$$B=\begin{Bmatrix}(P_1,P_2),(P_1,P_3),(P_1,P_4),\\(P_2,P_3),(P_2,P_4),(P_3,P_4)\end{Bmatrix}$$

비밀을 복원하기 위해 필요한 6개의 Share를 생성한다. 6개의 Share는 $\{s_1, s_2, s_3, s_4, s_5, s_6\}$로 나타낸다. 각 참가자는 함수 $g(i)$에 의해 다음과 같이 Share가 할당된다.

- $i=1, j=1$인 경우, 함수 $g(i)$는 다음과 같이 나타낼 수 있다. 여기서, $q=6$는 분산된 Share의 수이다.

$$g(1)=\{s_1 | P_1 \notin B_1, 1 \le j \le 6\}$$

함수 $g(1)$이면, 참가자 P_1이 B_1에 포함되지 않으면 Share(s_1)을 할당 한다. 하지만 참가자 P_1이 B_1에 포함되어 있으므로 $P_1 \notin B_1$은 성립하지 않으므로 Share(s_1)을 할당하지 않는다. 이러한 과정을 Share q번째까지 반복한다.

- 만약, $i=1, j=4$인 경우, 함수 $g(i)$는 다음과 같이 나타낼 수 있다. 여기서, $q=6$는 분산된 Share의 수이다.

$$g(1)=\{s_4 | P_1 \notin B_4, 1 \le j \le 6\}$$

〈표 1〉 Share 정보 할당

참가자	Share 정보
P_1	s_4, s_5, s_6
P_2	s_2, s_3, s_6
P_3	s_1, s_3, s_5
P_4	s_1, s_2, s_4

함수 $g(1)$이면, 참가자 P_1이 B_1에 포함되지 않으면 Share(s_4)을 할당 한다. 여기서, 참가자 P_1이 B_1에 포함되지 않으므로 $P_1 \not\subset B_4$가 성립함을 알 수 있다. 따라서, P_1에게 Share(s_4)를 할당 한다. 이렇게 $i=4$가 되도록 계속 반복한다. <표 1>은 Share 할당함수 g에 의해서 각 참가자가 갖는 Share정보이다.

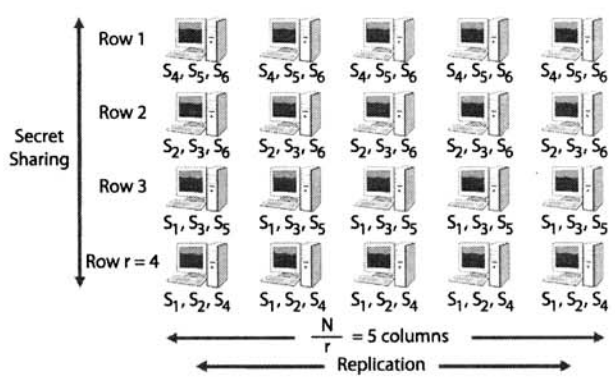
여기서, 비밀을 복원하기 위해서는 2명의 참가자로는 6개의 Share중 5개의 Share만 모을 수 있다. 복원시 필요한 6개의 Share가 없으므로 비밀정보를 복원할 수 없다. 따라서 분산된 비밀정보를 복원하는데 필요한 6개의 Share를 모두 모으기 위해서는 적어도 3명의 참가자가 필요하다.

2.3 GridSharing 프레임워크

GridSharing 프레임워크[2]는 문헌 [1]에서 제안된 Share 할당 방식을 기반으로 안전한 데이터 분산저장 관리 서비스를 구성하기 위한 프레임워크이다. 또한, 분산된 각 Share를 관리하기 위해 Replication기반 메커니즘을 사용한다. GridSharing 프레임워크는 N개의 서버로 구성된다. 여기서, N개의 서버는 r행과 $\frac{N}{r}$열의 배열로 이루어져 있고 각 행들은 $\frac{N}{r}$개의 서버로 구성된다(그림 1).

예를 들어, (그림 1)에서 $r=4$이고, $N=20$인 서버의 예를 보여준다. 만약, $\left(\binom{4}{2},\binom{4}{2}\right)=(6,6)$ XOR기반의 비밀분산 방식[7]의 경우,

비밀 정보 $S=s_1\oplus s_2\oplus s_3\oplus s_4\oplus s_5\oplus s_6$와 같이 6개의 Share($s_1, s_2, s_3, s_4, s_5, s_6$)로 분산된다고 가정한다. 문헌 [1]



(그림 1) GridSharing 프레임워크

의 Share 할당 함수 g에 따라 각 행의 서버에 저장된 Share는 <표 1>과 같이 분산되고 Replication된다. 여기서, 비밀정보 복원시 모든 Share를 모아야 하기 때문에 서버의 장애로 인한 문제가 발생하면 Replication된 서버의 이용으로 복원이 가능하다.

Ito, Saito, Nishizeki 방식은 분산된 비밀정보를 복원하기 위해 Share 모두가 필요하게 된다. 또한, Gridsharing Framework에서는 replication에 의한 DB 자원의 낭비 문제가 발생할 수 있다. 이를 해결하기 위해 조합디자인 파라메터의 자유로운 선택을 통해 즉, Share를 DB에 최적 할당함으로써 DB 자원 낭비를 없앨 수 있다. 그리고 임계치 이상의 Share가 모아지면 비밀정보를 복원할 수 있다는 관점에서 분산 DBMS 운영상의 효과가 있다는 점이다.

본 논문은 조합디자인 관점으로부터 Share 할당법을 검토한다. 즉, Share 할당을 위한 행렬 구조는 $t-(v,k,1)$ 조합디자인 파라메터와 연결시킬 수 있는 점에 착안한다.

3. 조합 디자인에 근거한 Share할당 구조

본 장에서는 Share 할당 구조와 조합디자인 파라메터와의 밀접한 관계에 대하여 고찰하고 조합디자인으로부터 Share 할당법을 구성한다.

3.1 Share 할당 구조와 조합디자인과의 관계

데이터 분산관리 관점에서 임의의 Share를 소유하는 DB 서버의 집합에 비밀을 복원할 수 있는 특성을 형식화하기 위해 Share 할당 구조를 도입한다. 기존 (k,n) 임계치 비밀분산 방식[6]의 정의는 다음과 같다.

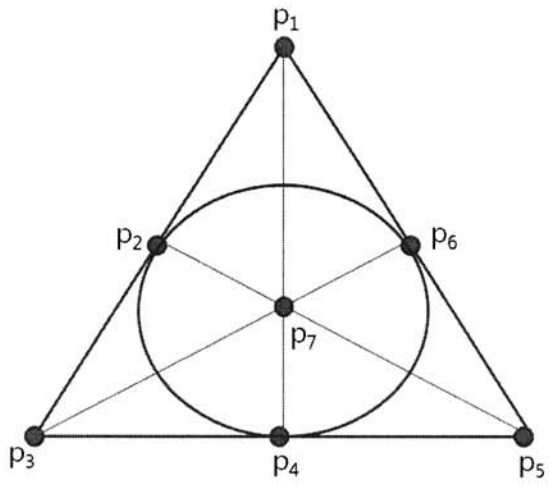
- 임의의 k개의 분산정보로부터 원래의 비밀정보 s를 복원할 수 있다.
- k-1개 이하의 분산정보로부터는 비밀정보 s에 관한 정보는 아무것도 얻을 수 없다.

이와 같이 Share 할당 구조는 $t-(v,k,\lambda)$ 디자인(λ=1인 경우 BIBD (Balanced Incomplete Block Design))과 같이 균형성이 있는 부분집합의 집합을 적절히 선택하여 Share 할당법의 비밀복원 특성을 만족하도록 할 수 있을까의 관점에서 해석할 수 있다.

여기서, 조합 디자인에 대하여 정의한다.

[정의 1] [3, 5] $t-(v,k,\lambda)$디자인은 다음의 성질을 만족하는 v개 점의 집합 $X=\{p_1,p_2,...,p_v\}$와 블록의 집합 D로 구성된다.

(1) 모든 블록은 정확히 X에 속한 서로 다른 k개의 점 $p_{i_1},p_{i_2},...,p_{i_k}(p_{i_j}\in X)$로 구성된다.



(그림 2) $2-(7,3,1)$ 디자인

(2) 임의의 서로 다른 t개의 점 $p_{i_1},p_{i_2},\ldots,p_{i_t}(p_{i_j}\in X)$을 포함하는 블록의 개수는 정확히 λ개다.

(예 1) v=7개 점의 집합 X, $k=3$개의 점으로 구성된 블록의 집합 D, $t=2$개의 임의의 점을 포함하는 블록을 개수로하여 $2-(7,3,1)$ 디자인으로 구성할 수 있다(그림 2).

$$X=\{p_1,p_2,p_3,p_4,p_5,p_6,p_7\}$$
$$D=\begin{Bmatrix}(p_1,p_2,p_3),(p_3,p_4,p_5),(p_1,p_5,p_6),(p_1,p_4,p_7),\\(p_2,p_5,p_7),(p_3,p_6,p_7),(p_2,p_4,p_6)\end{Bmatrix}$$

(1) 모든 블록은 정확히 k=3개의 점으로 구성된다.

(2) 임의의 t=2개의 점, 예를 들면 p_3,p_4를 포함하는 블록은 (p_3,p_4,p_5)로서 개수는 정확히 λ=1개이다.

3.2 Share 할당 구조의 행렬표현

$v_1,v_2,\ldots,v_n$을 분산정보(Share)로 한다. 여기서, 접근 집합을 비밀을 복원할 수 있는 Share 집합이라고 정의하고 복원불가능한 Share집합을 비접근 집합이라고 정의할 경우, Share 집합($t-(v,k,1)$ 디자인의 블록) 구조는 행렬에 의해 표현될 수 있다.

본 절에서는 조합디자인이 갖는 행렬구조를 Share 집합에 적용시키기 위해 예를 들어 설명하고 구성법을 정리한다. 특정 조합디자인이 갖는 행렬구조를 Share 집합으로 변환하고 이 Share 집합을 $t-(v,k,1)$ 디자인의 블록구조로 나타낸다. 여기서, 본 논문에서는 이러한 표현법을 Share 할당법이라고 한다. 가능한 Share의 집합 $(v_1,v_2,\ldots,v_n)$을 행으로 표현하는 행렬을 Share할당 행렬이라 하고 M으로 나타낸다. M을 이해하기 위해 (예 1)을 설명한다.

(예 2) (예 1)을 Share 할당 행렬 M으로 표현하면 다음과 같다.

$$M=\begin{bmatrix}1\,2\,3\\3\,4\,5\\1\,5\,6\\1\,4\,7\\2\,5\,7\\3\,6\,7\\2\,4\,6\end{bmatrix}$$

(예 3) $3-(10,4,1)$ 디자인으로부터 얻을 수 있는 Share 할당 행렬 M을 나타내면 아래와 같다.

$$X=\{p_1,p_2,\ldots,p_{10}\}$$

$$D=\begin{matrix}\{p_1,p_5,p_6,p_7\}\\\{p_1,p_2,p_8,p_9\}\\\{p_2,p_3,p_7,p_{10}\}\\\{p_3,p_4,p_6,p_9\}\\\{p_4,p_5,p_8,p_{10}\}\\\{p_2,p_3,p_4,p_8\}\\\{p_3,p_4,p_5,p_7\}\\\{p_1,p_4,p_5,p_9\}\\\{p_1,p_2,p_5,p_{10}\}\\\{p_1,p_2,p_3,p_6\}\\\{p_1,p_3,p_5,p_8\}\\\{p_1,p_2,p_4,p_7\}\\\{p_2,p_3,p_5,p_9\}\\\{p_1,p_3,p_4,p_{10}\}\\\{p_2,p_4,p_5,p_6\}\end{matrix}\qquad M=\begin{matrix}v_1\,v_2\,v_3\,v_4\\\begin{bmatrix}1\,5\,6\,\,7\\1\,2\,8\,\,9\\2\,3\,7\,10\\3\,4\,6\,\,9\\4\,5\,8\,10\\2\,3\,4\,\,8\\3\,4\,5\,\,7\\1\,4\,5\,\,9\\1\,2\,5\,10\\1\,2\,3\,\,6\\1\,3\,5\,\,8\\1\,2\,4\,\,7\\2\,3\,5\,\,9\\1\,3\,4\,10\\2\,4\,5\,\,6\end{bmatrix}\end{matrix}$$

[정리 1] $t-(v,k,1)$디자인의 각 블록을 행으로 하는 행렬을 M으로 한다. 이 M을 Share할당 행렬로 하는 Share 할당법에 있어서 임의의 t명은 비밀을 복원할 수 있는 최소한의 집합(접근집합) 이다.

(증명) [정의 1]의 (2)로부터 임의의 t개의 값$(x_1,\ldots,x_t)$는 M의 행(블록집합)을 결정한다. 따라서 임의의 t개의 값은 유일하게 특정 블록을 결정한다. 그러므로 임의의 t명은 접근 집합이다.

3.3 제안방식의 검토

$t-(v,k,1)$ 조합 디자인의 균형성 조건이 조합 디자인 파라메터 t,k,λ에 의해 비밀복원 특성을 어느정도 결정하는가를 검토한다. 즉, 분산정보를 점, 비밀분산함수를 블록 맵핑 함수로 대응시키면 비밀복원 특성은 조합디자인 파라메터 t,k,λ에 있어서 k점 중의 t개의 점이 주어지면 블록을 정확히 하나($\lambda=1$)를 결정할 수 있는 성질에 귀착된다.

조합디자인의 정의로부터 이러한 성질을 재해석하면,

- 각 블록에 포함되는 점의 개수는 k이다(블록의 크기 k가 비밀을 복원할 수 있는 인원수에 대응하고 있다).
- t개의 서로 다른 점에 대해서 이들을 모두 포함하는 블록의 개수는 일정하다(t가 비밀을 복원할 수 있는 최소 접근 집합(minimum access set)의 크기에 대응하고 있다).
- t개의 점을 포함하는 블록의 수는 λ이다 (λ가 최소접근 집합이 복원할 수 있는 비밀의 수에 대응하고 있다).

이와 같이 멤버의 부분집합인 k명중에서 t명이 모여지면 비밀분산 방식이 구성되고 최소 접근 집합인 t명은 $\lambda=1$개의 비밀 복원이 가능하다. 이와 같은 관점으로부터 비밀분산 방식은 조합디자인 파라메터 t,k,λ에 의해 특징지을 수 있다. 여기서, M과의 관계는 조합디자인 파라메터간의 관계에 의해 설명될 수 있다. 즉, $z-(\ell n,n,1)$ 디자인에서 DB의 총수인 m과 Share 총수인 $\ell n(=v)$가 $\ell n \le m$라는 관계식을 만족하는 경우, 본 논문에서의 할당법이 구성 가능하다(왜냐하면, $t-(v,k,1)$ 디자인에서 블록의 총수(m)는 점의 총수보다 크다는 관계식이 성립한다는 근거에 의함).

이러한 파라메터를 사전에 지정해서 비밀분산 방식을 구성하는 것이 조합 디자인 접근법의 이점이다. 반면 기존의 비밀분산방식은 $GF(q)$상의 파라메터 q에 의존하고 있다.

본 논문은 조합디자인 관점으로부터 비밀복원 특성을 형식화함으로써 비밀분산방식을 다음과 같이 재정의한다.

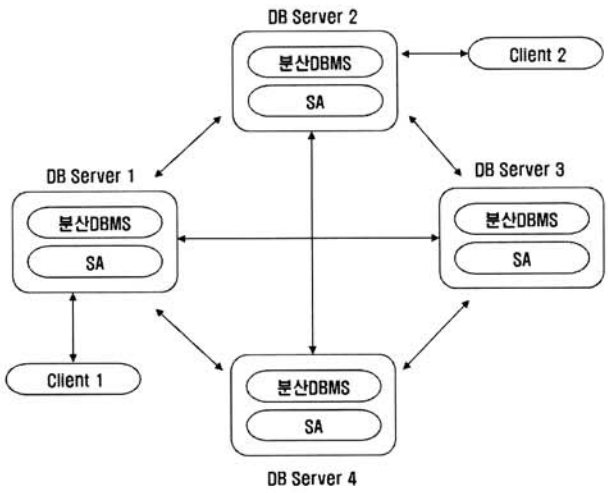
[정의 2] (t,k) 임계치 비밀분산방식은 다음과 같은 비밀복원특성을 만족하는 (P,F)이다. 여기서 P는 v개의 점(분산정보)의 집합, F는 크기 k의 P의 부분집합의 집합에 대응하는 비밀분산함수(블록맵핑함수)의 집합이다.

(1) k개의 점 중 t개 이상의 점으로부터 비밀을 유일하게 결정할 수 있다.
(2) k개의 점 중 $t-1$개의 이하의 점으로부터는 비밀을 전혀 결정할 수 없다.

4. 데이터 분산 저장관리에의 응용

본 제안 모델은 3장에서 검토한 Share 할당 구조를 적용하기 위해 DB의 한 테이블을 v개의 Fragment로 분할하고 각 Fragment를 n개의 Share로 나누어 Share 할당 구조를 기반으로 저장하고 복원시 z개의 Share로부터 원래의 Fragment를 복구하는 것이다. 제안모델은 네트워크상의 DB 서버에 분산 배치되어 Fragment를 관리하는 분산 DBMS와 Fragment를 분산, 복원하는 SA 그리고 Share를 저장하는 물리적인 Storage로 구성된다(그림 3).

여기서, Fragment란 분산 DBMS에서 제공되는 VP(Vertical Partitioning)기법을 이용하여 관계 테이블(Relation Table)을



(그림 3) 비밀분산 방식을 이용한 분산저장관리 모델

수직분할하는 것이고, $t-(v,k,1)$ 디자인은 Fragment를 비밀분산방식으로 분산시킨 Share를 물리적인 DB에 저장할 때 Share를 할당하는 방식이다.

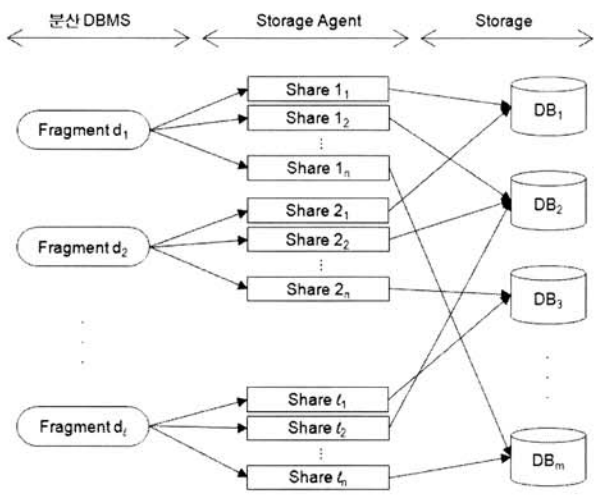
분산 DBMS는 쿼리 처리, 데이터 분할 관리 등 분산 DB의 기본적인 기능을 제공하며, SA는 XOR을 이용한 (t,k) 임계치 비밀분산 방식[7]으로 데이터 Relation의 Fragment를 분산, 복원한 후 데이터 베이스에 할당한다.

본 논문에서 사용되는 기호는 다음과 같다.
- m : DB의 총 수
- z : 임계치
- ℓ : Fragment의 총 수($(1 \le i \le \ell)$
- n : 분산정보(Share)의 수
- ℓn : 분산정보의 총 수
- d_i : i번째 Fragment

$t-(v,k,1)$디자인을 기반으로 본 제안 모델의 구성에 적용하기 위해 기호를 $z-(\ell n,n,1)$으로 변환하여 사용한다. 우선, v는 점들의 집합이고 ℓn은 Share의 집합으로 v와 ℓn은 같은 값을 갖는다. 그리고 k는 블록 집합에서 점의 개수이며, n은 Fragment d_i에서 분산된 Share의 개수이다. 마지막으로 t는 v개의 점의 집합에서 k개의 블록집합 들 중 임의의 t개의 점을 포함하는 블록의 개수는 정확히 1개라는 것인데 z는 n개의 Share 중에 임의의 z개를 모으면 1개의 Fragment를 복원할 수 있다는 것이다.

- 우선, 분산 DBMS에서는 VP(Vertical Partitioning)법을 이용하여 Partition 구조를 설정한 후 Relation Table을 ℓ개의 Fragment로 분할한다.
- Storage Agent에서 (k,n) 임계치 비밀분산에 의해 각 Fragment는 n개의 Share로 분산된다.

이때, ℓ개의 Fragment로 분산된 ℓn개의 Share는 m개의

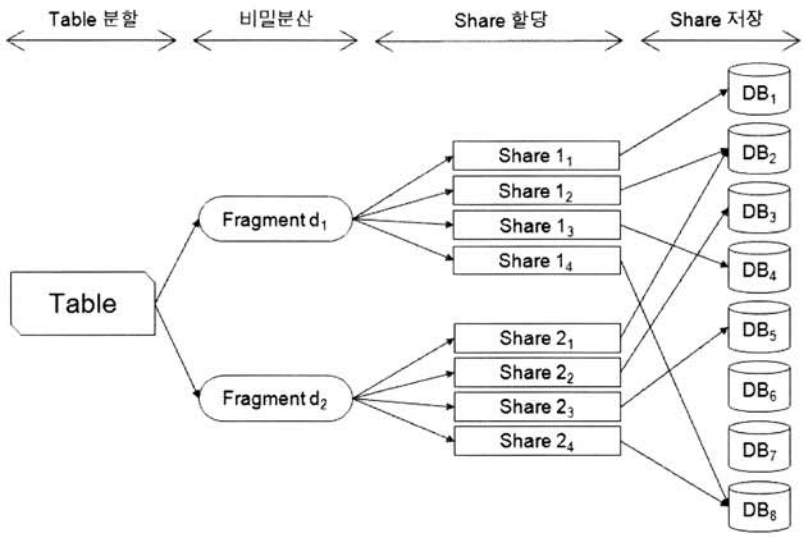


(그림 4) Fragment 작성 및 분산/저장

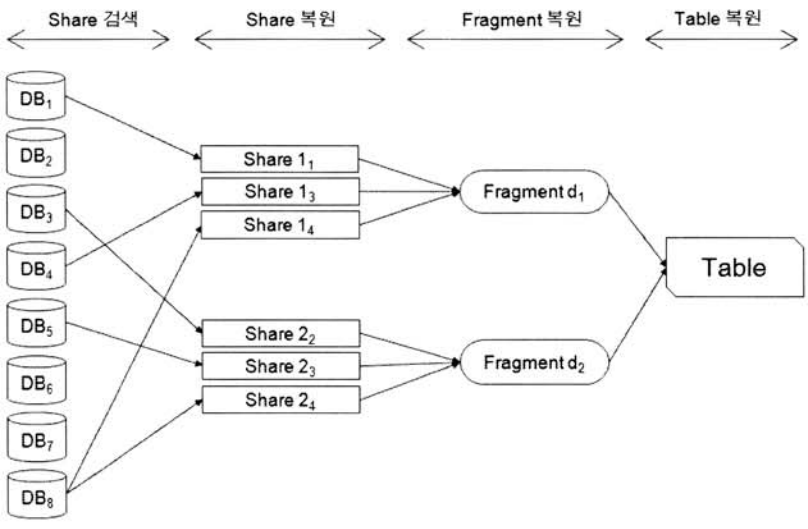
DB중에서 조합디자인, 특히 BIBD의 이산구조로 분산 배치한다(그림 4). 분할된 Fragment는 각각의 속성별로 구성되어 있는데 각 Fragment에는 식별할 수 있는 ID_i가 부여되고 (i=1,...,ℓ) 분할된 Fragment에서 분산된 Share는 Share 할당 테이블에 따라 DB_i (i=1,...,m)에 저장된다. Share 할당 테이블은 3.2절의 $t-(v,k,1)$ 디자인을 기반으로 하여 $z-(\ell n, n, 1)$의 형태로 구성된다.

(예 4) Relation Table에서 분할된 Fragment를 XOR기반 비밀분산방식으로 분산하는 경우, Fragment 수 $\ell=2$, 분산된 Share 수 $n=4$, 임계값 $z=3$으로 구성될 때, Share관리 테이블은 $3-(8,4,1)$ 디자인 구성법에 의해 DB 수 $m=8$에 저장되는 경우를 생각해보자

비밀정보를 복원하기 위해서는 3-(8,4,1) 디자인으로 저장된 Share에서 Fragment의 식별 아이디를 색인으로 하여 8개의 DB에서 검색하고자 하는 Fragment의 복원을 위해 4개



(그림 5) Share 할당과정 예, $m=8$, $n=4$, $\ell=2$



(그림 6) Share 복원과정 예, $m=8$, $n=4$, $\ell=2$

〈표 2〉 3-(8,4,1) 디자인의 경우 DB할당

ID	*DB*
$ID_1(d_1)$	DB_1, DB_2, DB_4, DB_8
$ID_2(d_2)$	DB_2, DB_3, DB_5, DB_8

의 DB중에서 임계치 3개의 Share 정보를 모아서 Fragment를 복원하고 분산 DBMS는 복원된 Fragment를 이용하여 관계 테이블을 생성하여 비밀정보를 복원하게 된다.

Ito, Saito, Nishizeki 할당 방식[1]은 분산된 비밀정보를 복원하기 위해 Share 모두가 필요하게 된다. 반면, 본 제안 방식의 경우, 임계치 이상의 Share가 모아지면 비밀정보를 복원할 수 있다. 또한, 데이터를 분산.복원하는 분산 DBMS 운영상의 효과 특히, DB 서버 및 Share 선택의 규칙성이 있고 조합적 구조인 Combinatorial Design상의 파라메터 t, v, k를 이용함으로써 구현상의 유연성(flexibility)을 갖는다.

5. 결 론

본 논문에서는 분산 저장되는 Share 할당 구조를 조합디자인이 갖는 행렬구조의 관점에서 해석하는 것에 의해 Share 할당법과 조합디자인이 밀접한 관계가 있음을 고찰하였다. 또한, 조합적 구조가 갖는 균형성 등을 최대한 활용함으로써 비밀을 분산저장 관리하는 메커니즘을 설계할 경우 구현의 용이성 및 유연성 확보가 가능하다. 향후 과제로서는 본 논문에서 고찰된 Share할당법과 기존 연구와의 실증적 비교 연구가 필요하다.

참 고 문 헌

[1] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," In Proceedings of the IEEE Global Communication Conference, 1987.

[2] A. Subbiah and D. M. Blough, "An approach for fault tolerant and secure data storage in collaborative work environements," In Proc. of the 2005 ACM Workshop on Storage Security and Survivabiilty, pp.84-93, Nov., 2005.

[3] T. Beth, D. Jungnickel, and H. Lenz, "Design theory," Cambridge Univ. Press, 1993.

[4] J. Benaloh and J. Letchter, "Generalized secret sharing and monotone functions," Advances in Cryptography-Proc. of Crypto'88, Notes Comp. Science, pp.27-35, 1990.

[5] D. R. Hughes and F. C. Piper, "Design theory," Cambridge Univ. Press, Cambidge, 1985.

[6] A. Shamir, "How to share secret," Comm. of the ACM, 22, pp.612-613, 1979.

[7] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "On a Fast (k,n)-Threshold Secret Sharing Scheme," IEICE Trans. Fundamentals, Vol.E91-A, No.9, pp.2365-2377, Jan., 2008.

[8] C. Cachin, "On-line Secret Sharing," Cryptography and Coding, LNCS Vol.1025, pp.190-198, 2005.

[9] Raghu Rmakrishnan, "Sherpa: Cloud Computing of the Third Kind," Data- Intensive Computing Symposium, 2008.

[10] Jeff Dean, "Handling Large Datasets at Google: Current Systems and Future Directions," Data-Intensive Computing Symposium, 2008.

송 유 진

e-mail : song@dongguk.ac.kr

1982년 한국항공대학교(학사)

1987년 경북대학교(석사)

1995년 일본 Tokyo Institute of Technology (박사)

1988년~1996년 한국전자통신연구원 선임연구원

2003년~2005년 미국 University of North Carolina at Charlotte 연구교수

2006년 7월~8월 일본 정보보호대학원대학 객원교수

1996년~현 재 동국대학교 정보경영학과/대학원 교수

2005년~현 재 동국대학교 부설 전자상거래연구소 소장

1998년~현 재 한국정보보호학회 이사

2006년~현 재 국제e-비즈니스학회 이사

2006년~현 재 한국사이버테러정보전학회 이사

2001년 ICISC2001 운영위원장

2003년 하계CISC2003 프로그램위원장

2006년 CISC-S2006 공동프로그램 위원장

2007년 한국정보시스템학회 추계학술발표대회 공동조직위원장

관심분야 : IT 융합보안(의료보안, 스마트그리드 보안) Cloud Security and Privacy, Secret Sharing, Context Aware Application Security

박 광 용

e-mail : freemickey@dongguk.ac.kr

2008년 동국대학교 전자상거래학과(학사)

2008년~현 재 동국대학교 전자상거래협동석사과정

관심분야 : 암호이론, 데이터 베이스 보안, 유비쿼터스 프라이버시 보호, 클라우드 보안 등

강 연 정

e-mail : yjkang@kisa.or.kr

2003년 한양대학교 전자전기공학부(학사)

2006년 한양대학교 수학과(이학석사)

2006년~2009년 한국정보보호진흥원 IT기반보호단 연구원

2009년~현 재 한국인터넷진흥원 인터넷 융합·정책본부 주임연구원

관심분야 : 암호학, 정보보호 등