



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년02월27일

(11) 등록번호 10-1496764

(24) 등록일자 2015년02월23일

(51) 국제특허분류(Int. Cl.)

G06F 21/60 (2013.01)

(21) 출원번호 10-2013-0054337

(22) 출원일자 2013년05월14일

심사청구일자 2013년06월18일

(65) 공개번호 10-2014-0134796

(43) 공개일자 2014년11월25일

(56) 선행기술조사문헌

JP2008278022 A

JP2007078739 A

KR1020110083189 A

US20120163592 A1

전체 청구항 수 : 총 18 항

(73) 특허권자

서울대학교산학협력단

서울특별시 관악구 관악로 1 (신림동)

(72) 발명자

박영훈

서울 영등포구 당산로 214, 405동 2001호 (당산동5가, 삼성래미안4차아파트)

서승우

서울 관악구 관악로 1, 뉴미디어통신공동연구소 132동 405호 (신림동, 서울대학교)

(74) 대리인

김영철, 김 순 영

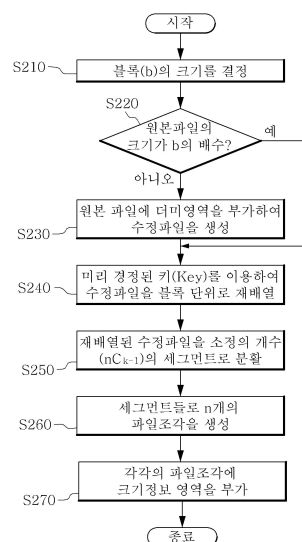
심사관 : 구대성

(54) 발명의 명칭 일정 수 이상의 파일 조각으로 복구 가능한 파일 분산 관리 장치 및 방법

(57) 요약

본 발명에서는 주어진 파일을 이용하여 소정의 파일 조각들을 생성하되, 파일 조각들 중 일정 수 이상의 파일로부터는 원래의 파일로 복구가 가능하지만, 일정 수 미만의 파일로부터는 파일 복구가 불가능하도록 파일을 관리하는 파일 분산 관리 장치 및 방법에 관한 것이다. 또한, 본 발명에서는 파일 조각 생성시 원본 파일의 순서를 섞어 보안성을 향상시킨다. 본 발명의 원본 파일의 순서를 섞는 방법에 있어서, 원본 파일의 순서는 블록 단위로 섞이고, 블록의 크기는 사용자가 임의로 선택할 수 있다. 또한, 원본 파일들로부터 복수의 세그먼트를 생성하여 파일 조각들을 구성하되, 복수의 세그먼트들의 크기는 일정한 비율을 갖거나 또는 임의의 값을 가질 수 있다.

대표도 - 도4



이 발명을 지원한 국가연구개발사업

과제고유번호 2009-0083495

부처명 교육과학기술부

연구관리전문기관 한국연구재단

연구사업명 기초연구사업

연구과제명 고신뢰도 임베디드 통신 플랫폼 최적 설계기술

기 여 율 1/1

주관기관 서울대학교

연구기간 2013.03.01 ~ 2014.02.28

특허청구의 범위

청구항 1

제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하는 단계;

상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계(상기 nC_{k-1} 에서, 'C'란 조합(Combination)을 의미함. 즉, nC_{k-1} 란 서로 다른 n 개에서 순서를 생각하지 않고 $k-1$ 개를 뽑는 경우의 수임); 및

상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n 개의 파일 조각들을 생성하는 단계를 포함하되,

상기 n 은 0보다 큰 정수이고,

상기 k 는 0보다 크고 상기 n 이하인 정수이고,

상기 n 개의 파일 조각들 중 임의의 k 개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함되는, 파일 분산 관리 방법

청구항 2

제 1 항에 있어서,

상기 제 1 파일은 상기 블록들의 크기에 따라, 상기 제 1 파일의 크기가 상기 블록들의 크기의 정수배가 되도록 원본 파일에 더미 영역을 선택적으로 부가하여 생성되는, 파일 분산 관리 방법.

청구항 3

제 2 항에 있어서,

상기 제 2 파일을 생성하는 단계는,

상기 블록들의 크기를 결정하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 4

제 1 항에 있어서,

상기 제 2 파일을 생성하는 단계는, 가역적이고 매 수행시마다 서로 다른 결과를 출력하는 재배열 알고리즘을 이용하여 상기 블록들의 배치 순서를 재배열하는, 파일 분산 관리 방법.

청구항 5

제 1 항에 있어서,

상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계는,

상기 nC_{k-1} 개의 세그먼트들 각각이 미리 결정된 비율의 크기들을 갖도록 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계를 포함하는, 파일 분산 관리 방법.

청구항 6

제 5 항에 있어서,

상기 n 개의 파일 조각들을 생성하는 단계는,

상기 n 개의 파일 조각들 각각에 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 추가하는 단계를 포함하는, 파일 분산 관리 방법.

청구항 7

제 1 항에 있어서,

상기 제 2 파일을 $n_{C_{k-1}}$ 개의 세그먼트들로 분할하는 단계는,

상기 $n_{C_{k-1}}$ 개의 세그먼트들 각각이 임의의 크기들을 갖도록 상기 제 2 파일을 $n_{C_{k-1}}$ 개의 세그먼트들로 분할하는 단계를 포함하는, 파일 분산 관리 방법.

청구항 8

제 7 항에 있어서,

상기 n 개의 파일 조각들을 생성하는 단계는,

상기 n 개의 파일 조각들 각각에, 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보 및, 상기 n 개의 파일 조각들 각각이 포함하는 세그먼트들의 크기들을 나타내는 크기 정보를 추가하는 단계를 포함하는, 파일 분산 관리 방법.

청구항 9

제 1 항에 있어서,

상기 생성된 n 개의 파일 조각들을 n 개의 분산 저장 장치들에 전송하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 10

제 9 항에 있어서,

상기 n 개의 분산 저장 장치들 중 적어도 k 개의 분산 저장 장치들로부터 k 개의 파일 조각들을 수신하는 단계;

상기 k 개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k 개의 파일 조각들로부터 상기 크기 정보 영역을 제거하는 단계;

상기 크기 정보 영역이 제거된 k 개의 파일 조각들로부터 $n_{C_{k-1}}$ 개의 복원 세그먼트들을 분할하는 단계;

상기 $n_{C_{k-1}}$ 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하는 단계; 및

상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 11

제 10 항에 있어서,

상기 크기 정보를 참조하여, 상기 복원된 제 1 파일로부터 상기 더미 영역을 제거하여 원본 파일을 복원하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 12

제 10 항에 있어서,

상기 nC_{k-1} 개의 복원 세그먼트들 각각은 미리 결정된 비율의 크기들을 갖는, 파일 분산 관리 방법.

청구항 13

제 10 항에 있어서,

상기 nC_{k-1} 개의 복원 세그먼트들 각각은 임의의 크기들을 갖는, 파일 분산 관리 방법.

청구항 14

제 13 항에 있어서,

상기 크기 정보는, 상기 k개의 파일 조각들에 포함된 상기 nC_{k-1} 개의 복원 세그먼트들의 크기들을 나타내는 정보를 포함하는, 파일 분산 관리 방법.

청구항 15

제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하고, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 파일 분할부;

상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n개의 파일 조각들을 생성하는 파일 조각 생성부; 및

상기 생성된 n개의 파일 조각들을 n개의 분산 저장 장치들에 전송하는 통신부를 포함하되,

상기 n은 0보다 큰 정수이고,

상기 k는 0보다 크고 상기 n이하인 정수이고,

상기 n개의 파일 조각들 중 임의의 k개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함되는, 파일 분산 관리 장치.

청구항 16

제 15 항에 있어서,

상기 통신부를 통해 상기 n개의 분산 저장 장치들 중 적어도 k개의 분산 저장 장치들로부터 k개의 파일 조각들을 수신하고, 상기 k개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k개의 파일 조각들로부터 상기 크기 정보 영역을 제거하고, 상기 크기 정보 영역이 제거된 k개의 파일 조각들로부터 nC_{k-1} 개의 복원 세그먼트들을 분할하고, 상기 nC_{k-1} 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하고, 상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 파일 복원부를 더 포함하는, 파일 분산 관리 장치.

청구항 17

제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하는 단계, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계 및 상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n 개의 파일 조각들을 생성하는 단계를 포함하되, 상기 n 은 0보다 큰 정수이고, 상기 k 는 0보다 크고 상기 n 이하인 정수이고, 상기 n 개의 파일 조각들 중 임의의 k 개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함되는 파일 분산 관리 방법을 실행하기 위한, 컴퓨터 프로그램을 기록하는 컴퓨터 판독 가능한 기록 매체.

청구항 18

제 17 항에 있어서,

상기 n 개의 분산 저장 장치들 중 적어도 k 개의 분산 저장 장치들로부터 k 개의 파일 조각들을 수신하는 단계, 상기 k 개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k 개의 파일 조각들로부터 상기 크기 정보 영역을 제거하는 단계, 상기 크기 정보 영역이 제거된 k 개의 파일 조각들로부터 nC_{k-1} 개의 복원 세그먼트들을 분할하는 단계, 상기 nC_{k-1} 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하는 단계 및 상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 단계를 더 포함하는 파일 분산 관리 방법을 실행하기 위한, 컴퓨터 프로그램을 기록하는 컴퓨터 판독 가능한 기록 매체.

명세서

기술분야

[0001]

본 발명은 파일 분산 관리 장치 및 그 방법에 관한 것으로서, 더욱 상세하게는 일정 수 이상의 파일 조각으로 복구 가능한 파일 분산 관리 장치 및 그 방법에 관한 것이다.

배경기술

[0002]

비밀 정보를 보관하는 경우, 비밀 정보가 분실되거나 파괴될 위험성은 항상 존재한다. 이와 동시에 비밀정보가 도난될 수 있는 위험성 역시 존재한다. 분실 또는 파괴의 위험성은 비밀 정보를 복수 곳의 장소에 보관해 둬으로써 줄일 수 있지만, 이 경우 도난의 위험성이 증가하는 단점이 있다. 이들 위험성을 함께 해결하는 방법의 하나로써 비밀 분산법(Secret Sharing)이 제시되었다.

비밀 분산법은 비밀 정보 MSK로부터 복수의 분산 정보(예를 들어, SH(1), ..., SH(N))를 생성하고, 이들을 복수의 분산 저장 장치(예를 들어, PA(1), ..., PA(N))에 분산하여 관리시키고, 이들 분산 정보 SH(1), ..., SH(N) 중 소정 수 이상의 정보를 취득할 수 있는 경우에만, 비밀 정보 MSK를 복원할 수 있는 방식이다.

비밀 분산법은 저장되는 값의 기밀성, 가용성, 무결성을 모두 보장해줄 수 있다는 장점 덕분에 많은 분산 관리 장치에 적용되어 왔다. 저장된 파일(F)을 작은 단위인 값(예를 들어, F[1], F[2], ..., F[s], 단, s는 파일을 이루는 값의 개수)으로 쪼갬 뒤, 각각의 값 F[i]들을 비밀 분산법을 이용하여 복수의 분산 정보 f[i,1], f[i,2], ..., f[i,n] (단, n은 저장 장치의 개수)을 만들어 내고, 분산 정보를 f(s) = f[1,m] || f[2,m] || ... || f[s,m] (단, m=1,2,...,n) 과 같이 연결하여 n개의 파일 조각 f(1), f(2), ..., f(n)을 만들어 낸다.

기존 비밀 분산법이 적용된 파일 관리 방법에서는 일정 수 미만의 파일 조각으로 전체 파일의 내용을 알 수 없으므로 저장된 파일의 기밀성을 제공할 수 있었고, 일정 수 미만의 파일 조각이 없어지거나 손상되더라도 나머지 파일 조각으로 파일을 복구할 수 있기 때문에 가용성을 보장해줄 수 있었다. 그러나, 최근 들어 저장되는 파일의 크기가 커지면서 연산량이 많은 기존 비밀 분산법을 적용하기에는 무리가 있음과 동시에 파일 조각이 원래의 파일과 크기가 동일하여 저장 공간 및 통신 비용의 낭비가 커질 가능성이 있다.

[0003] 삭제

[0004] 삭제

[0005] 삭제

발명의 내용

해결하려는 과제

[0006] 본 발명의 목적은 파일 시스템의 파일 저장시 파일의 기밀성, 무결성, 가용성을 모두 보장함과 동시에 분산된 파일의 복구를 빠르게 수행하는 파일 분산 관리 장치 및 그 방법을 제공하는 데 있다.

[0007] 또한, 사용자가 파일 블록의 크기를 임의로 조정할 수 있어서 파일에 따라 보안 수준을 결정할 수 있는 파일 분산 관리 장치 및 그 방법을 제공하는 데 있다.

과제의 해결 수단

본 발명의 파일 분산 관리 방법은 제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하는 단계; 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계; 및 상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n 개의 파일 조각들을 생성하는 단계를 포함하고, 상기 n 은 0보다 큰 정수이고, 상기 k 는 0보다 크고 상기 n 이하인 정수이고, 상기 n 개의 파일 조각들 중 임의의 k 개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함된다. 상기 nC_{k-1} 에서, 'C'란 조합(Combination)을 의미한다. 즉, nC_{k-1} 란 서로 다른 n 개에서 순서를 생각하지 않고 $k-1$ 개를 뽑는 경우의 수이다.

실시 예로서, 상기 제 1 파일은 상기 블록들의 크기에 따라, 상기 제 1 파일의 크기가 상기 블록들의 크기의 정수배가 되도록 원본 파일에 더미 영역을 선택적으로 부가하여 생성된다.

실시 예로서, 상기 제 2 파일을 생성하는 단계는, 상기 블록들의 크기를 결정하는 단계를 더 포함한다.

실시 예로서, 상기 제 2 파일을 생성하는 단계는, 가역적이고 매 수행시마다 서로 다른 결과를 출력하는 재배열 알고리즘을 이용하여 상기 블록들의 배치 순서를 재배열한다.

실시 예로서, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계는, 상기 nC_{k-1} 개의 세그먼트들 각각이 미리 결정된 비율의 크기들을 갖도록 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계를 포함한다.

실시 예로서, 상기 n 개의 파일 조각들을 생성하는 단계는, 상기 n 개의 파일 조각들 각각에 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 부가하는 단계를 포함한다.

실시 예로서, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계는, 상기 nC_{k-1} 개의 세그먼트들 각각이 임의의 크기들을 갖도록 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계를 포함한다.

실시 예로서, 상기 n 개의 파일 조각들을 생성하는 단계는, 상기 n 개의 파일 조각들 각각에, 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보 및, 상기 n 개의 파일 조각들 각각이 포함하는 세그먼트들의 크기들을 나타내는 크기 정보를 부가하는 단계를 포함한다.

실시 예로서, 상기 생성된 n 개의 파일 조각들을 n 개의 분산 저장 장치들에 전송하는 단계를 더 포함한다.

실시 예로서, 상기 n 개의 분산 저장 장치들 중 적어도 k 개의 분산 저장 장치들로부터 k 개의 파일 조각들을 수신하는 단계; 상기 k 개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k 개의 파일 조각들로부터 상기 크기 정보 영역을 제거하는 단계; 상기 크기 정보 영역이 제거된 k 개의 파일 조각들로부터 nC_{k-1} 개의 복

원 세그먼트들을 분할하는 단계; 상기 nC_{k-1} 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하는 단계; 및 상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 단계를 더 포함한다.

실시 예로서, 상기 크기 정보를 참조하여, 상기 복원된 제 1 파일로부터 상기 더미 영역을 제거하여 원본 파일을 복원하는 단계를 더 포함한다.

실시 예로서, 상기 nC_{k-1} 개의 복원 세그먼트들 각각은 미리 결정된 비율의 크기들을 갖는다.

실시 예로서, 상기 nC_{k-1} 개의 복원 세그먼트들 각각은 임의의 크기들을 갖는다.

실시 예로서, 상기 크기 정보는, 상기 k개의 파일 조각들에 포함된 상기 nC_{k-1} 개의 복원 세그먼트들의 크기들을 나타내는 정보를 포함한다.

본 발명의 파일 분산 관리 장치는 제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하고, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 파일 분할부; 상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n개의 파일 조각들을 생성하는 파일 조각 생성부; 및 상기 생성된 n개의 파일 조각들을 n개의 분산 저장 장치들에 전송하는 통신부를 포함하되, 상기 n은 0보다 큰 정수이고, 상기 k는 0보다 크고 상기 n이하인 정수이고, 상기 n개의 파일 조각들 중 임의의 k개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함된다.

실시 예로서, 상기 통신부를 통해 상기 n개의 분산 저장 장치들 중 적어도 k개의 분산 저장 장치들로부터 k개의 파일 조각들을 수신하고, 상기 k개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k개의 파일 조각들로부터 상기 크기 정보 영역을 제거하고, 상기 크기 정보 영역이 제거된 k개의 파일 조각들로부터 nC_{k-1} 개의 복원 세그먼트들을 분할하고, 상기 nC_{k-1} 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하고, 상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 파일 복원부를 더 포함한다.

본 발명에 따른 컴퓨터 판독 가능한 기록 매체는 제 1 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 제 2 파일을 생성하는 단계, 상기 제 2 파일을 nC_{k-1} 개의 세그먼트들로 분할하는 단계 및 상기 nC_{k-1} 개의 세그먼트들 중 $n-1C_{k-1}$ 개의 세그먼트들을 각각 포함하는 n개의 파일 조각들을 생성하는 단계를 포함하고, 상기 n은 0보다 큰 정수이고, 상기 k는 0보다 크고 상기 n이하인 정수이고, 상기 n개의 파일 조각들 중 임의의 k개의 파일 조각들에는 상기 nC_{k-1} 개의 세그먼트들이 포함되는 파일 분산 관리 방법을 실행하기 위한, 컴퓨터 프로그램을 기록한다.

실시 예로서, 상기 n개의 분산 저장 장치들 중 적어도 k개의 분산 저장 장치들로부터 k개의 파일 조각들을 수신하는 단계, 상기 k개의 파일 조각들의 크기 정보 영역으로부터 상기 제 1 파일에서 더미 영역을 제외한 부분의 크기 또는 상기 제 1 파일의 원본 파일의 크기를 나타내는 크기 정보를 읽어내고, 상기 k개의 파일 조각들로부터 상기 크기 정보 영역을 제거하는 단계, 상기 크기 정보 영역이 제거된 k개의 파일 조각들로부터 nC_{k-1} 개의 복원 세그먼트들을 분할하는 단계, 상기 nC_{k-1} 개의 복원 세그먼트들을 연결하여 상기 제 2 파일을 복원하는 단계 및 상기 복원된 제 2 파일에 포함된 블록들의 배치 순서를 블록 단위로 재배열하여 상기 제 1 파일을 복원하는 단계를 더 포함하는 파일 분산 관리 방법을 실행하기 위한, 컴퓨터 프로그램을 기록한다.

발명의 효과

[0008]

본 발명에 따르면, 하나의 파일이 여러 대의 기기에 나뉘어서 저장되므로 k개 미만의 기기에서 데이터가 유출되었을 경우, 원래 파일의 복원이 불가능하므로 보안성이 향상되는 효과가 있다.

또한, k개 이상의 기기가 있다고 하더라도, 파일 내용의 순서가 섞여 있으므로 원본 파일의 복구가 어려워 보안성이 더욱 향상될 수 있다.

또한, 파일의 일부가 손실되는 경우에도, 나머지 k개의 파일을 통해 원본 파일을 복구할 수 있어 가용성이 향상

될 수 있다.

또한, 조합적인 방법을 이용함으로써 연산량이 오로지 F에만 비례하므로, 종래의 비밀분산법에 비해 연산량 및 복잡성이 감소할 수 있다..

또한, 사용자가 파일 블록의 크기를 임의로 조정할 수 있으므로, 분산된 파일의 보안 수준을 사용자가 결정할 수 있다.

[0009] 삭제

[0010] 삭제

[0011] 삭제

[0012] 삭제

[0013] 삭제

도면의 간단한 설명

[0014] 도 1은 본 발명의 일 실시예에 따른, 파일 분산 관리 시스템(1000) 및 그 내부 구성을 나타낸 도면이다

도 2는 본 발명의 일 실시 예에 따른, 파일 분산 관리 장치(1100)의 내부 구성을 나타낸 도면이다.

도 3은 본 발명의 실시 예에 따른 파일 분산 관리 방법을 나타내는 순서도이다.

도 4는 본 발명의 실시 예에 따른 파일 조각 생성 방법을 나타내는 순서도이다.

도 5는 본 발명의 실시 예에 따른 원본 파일 복원 방법을 나타내는 순서도이다.

도 6은 본 발명의 일 실시 예에 따라 파일 조각을 생성하는 방법을 나타내는 블록도이다.

도 7은 본 발명의 다른 실시 예에 따라 파일 조각을 생성하는 방법을 나타내는 블록도이다.

발명을 실시하기 위한 구체적인 내용

후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시 예를 예시로서 도시하는 첨부 도면을 참조한다. 앞의 일반적인 설명 및 다음의 상세한 설명들은 모두 청구된 발명의 부가적인 설명을 제공하기 위한 예시적인 것이다. 그러므로 본 발명은 여기서 설명되는 실시 예에 한정되지 않고 다른 형태로 구체화될 수도 있다. 여기서 소개되는 실시 예는 개시된 내용이 완전해 질 수 있도록, 그리고 당업자에게 본 발명의 사상이 충분히 전달될 수 있도록 하기 위해 제공되는 것이다.

본 명세서에서, 어떤 부분이 어떤 구성요소를 포함한다고 언급되는 경우에, 이는 그 외의 다른 구성요소를 더 포함할 수도 있다는 것을 의미한다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

도 1은 본 발명의 일 실시예에 따른, 파일 분산 관리 시스템(1000) 및 그 내부 구성을 나타낸 도면이다. 파일 분산 관리 시스템(1000)은 파일 분산 관리 장치(1100) 및 n개의 파일 분산 저장 장치들(1100a, 1100b, ..., 1100n)로 구성된다.

파일 분산 관리 시스템(1000)은 원본 파일로부터 생성한 복수의 파일 조각들을 n개의 분산 저장 장치들(1100a, 1100b, ..., 1100n)에 분산하여 저장한다. 그리고, 파일 분산 관리 시스템(1000)이 1분산된 파일 조각들로부터 원본 파일을 복원하기 위해서는, k개(k는 0보다 큰 n이하의 정수)의 임의의 분산 저장 장치들에 저장된 k개의 파일 조각들을 획득하여야 한다. 즉, 파일 분산 관리 시스템(1000)에서 k개의 파일 조각들을 획득하는 경우 원본 파일이 완전히 복원될 수 있지만, k개보다 적은 수의 파일 조각들만을 획득하는 경우 원본 파일은 완전히 복

원되지 않는다.

분산 저장 장치들(1100a, 1100b, ..., 1100n) 각각은 원본 파일로부터 생성된 파일 조각을 저장한다. 그리고, 파일 분산 관리 장치(1100)의 제어 또는 파일 관리 장치(1100)로부터의 요청에 응답하여 저장된 파일 조각의 일부 또는 전부를 파일 분산 관리 장치(1100)에 제공한다.

분산 저장 장치들(1100a, 1100b, ..., 1100n)은 특별한 기능이 없더라도 각각의 파일 조각을 분리하여 저장할 수 있는 장치로서 구성되면 충분하다. 도 1에서, 분산 저장 장치들(1100a, 1100b, ..., 1100n)은 물리적으로 분리된 형태로 표현되어 있지만, 이에 한정되는 것은 아니며, 분산 저장 장치들(1100a, 1100b, ..., 1100n)은 1개의 물리적 장치에서 논리적으로 분리된 형태로도 가능할 수 있다.

일 실시예에서, 분산 저장 장치들(1100a, 1100b, ..., 1100n) 각각은 파일 분산 관리 장치(1100)와의 통신을 위한 별도의 통신부 또는 통신 인터페이스를 구비할 수 있다.

파일 분산 관리 장치(1100)는 분산 저장 장치들로부터 파일 조각들을 수신하고, 수신된 파일 조각들을 이용하여 원본 파일을 복원한다. 파일 분산 관리 장치(1100)에 대한 더욱 상세한 설명은 아래에서 도 2를 참조하여 제공된다.

도 2는 본 발명의 일 실시 예에 따른, 파일 분산 관리 장치(1100)의 내부 구성을 나타낸 도면이다. 파일 분산 관리 장치(1100)는 파일 분할부(1110), 파일 조각 생성부(1120), 통신부(1130), 파일 복원부(1140), 크기 정보 관리부(1150) 및 저장부(1160)를 포함할 수 있으나, 이에 한정되는 것은 아니다. 예를 들어, 파일 분산 관리 장치(1100)는 파일 분산 관리 장치(1100)의 전체 동작을 제어하는 별도의 제어부(미도시)를 더 포함할 수 있다.

파일 분할부(1110)는 원본 파일을 분할하여 $n_{C_{k-1}}$ 개의 세그먼트를 생성한다. 또한, 파일 분할부(1110)는 원본 파일을 분할하기에 앞서, 소정의 크기(예를 들어, 2 바이트)의 블록 단위로 원본 파일의 순서를 뒤섞는 재배열을 수행하고, 재배열된 원본 파일을 분할하여 $n_{C_{k-1}}$ 개의 세그먼트를 생성할 수 있다.

실시 예로서, 파일 분할부(1110)는 원본 파일에 더미 영역을 추가하여 수정 파일을 생성하고, 수정 파일을 분할하여 $n_{C_{k-1}}$ 개의 세그먼트를 생성할 수 있다. 또한, 또한, 파일 분할부(1110)는 수정 파일을 분할하기에 앞서, 소정의 크기의 블록 단위로 수정 파일의 순서를 뒤섞는 재배열을 수행하고, 재배열된 수정 파일을 분할하여 $n_{C_{k-1}}$ 개의 세그먼트를 생성할 수 있다.

이때, 재배열의 기준이 되는 블록의 크기는 크기 정보 관리부(1150)로부터 참조될 수 있다. 또한, 재배열의 기준이 되는 블록의 크기는 사용자로부터 소정의 인터페이스(미도시)를 통해 입력될 수 있다.

파일 조각 생성부(1120)는 파일 분할부(1110)에 의해 생성된 $n_{C_{k-1}}$ 개의 세그먼트들로부터 n 개의 파일 조각들을 생성한다. 단, n 개의 파일 조각들 중 임의의 k 개 파일 조각들을 선택할 때, 선택된 k 개의 파일 조각들에는 $n_{C_{k-1}}$ 개의 세그먼트들이 모두 포함되도록, n 개의 파일 조각들이 생성된다. 따라서, n 개의 파일 조각들은 서로 중복되는 세그먼트들을 포함할 수 있다.

파일 분할부(1110) 및 파일 조각 생성부(1120)가 원본 파일로부터 n 개의 파일 조각들을 생성하는 구체적인 방법은 도 4 이하에서 구체적으로 후술된다.

통신부(1130)는 파일 분산 관리 장치(1100)와 외부 객체와의 통신을 수행하는 역할을 한다. 예를 들어, 파일 분산 관리 장치(1100)는 통신부(1130)를 통해 분산 저장 장치들(1100a, 1100b, ..., 1100n, 도 1 참조)과 통신할 수 있다. 통신부(1130)는 외부와 네트워킹 또는 통신이 가능한 모든 통신 수단을 포함할 수 있으며, 그러한 통신 수단에 수반되는 통신 인터페이스를 구비할 수 있다. 통신부(1130)가 이용하는 통신 인터페이스 또는 통신 방법은 유선 통신, 무선 통신, 3G, 4G, 또는 그 밖의 다양한 통신 인터페이스 또는 통신 방법을 포함할 수 있다.

파일 복원부(1140)는 분산 저장 장치들(1100a, 1100b, ..., 1100n, 도 1 참조)로부터 적어도 k 개의 파일 조각들을 수신하고, 수신된 파일 조각들로부터 원본 파일을 복원한다. 파일 복원부(1140)가 파일을 복원하는 구체적인 방법은 도 5 이하에서 후술된다.

저장부(1160)는 파일 분산 관리 장치(1100)에 수신된 데이터 또는 파일 분산 관리 장치(1100)에서 처리한 데이터를 저장한다. 예를 들어, 저장부(1160)는 파일 분할부(1110), 파일 조각 생성부(1120) 또는 파일 복원부

(1140)의 기능을 수행하는 과정에서 얻어진 데이터들을 저장할 수 있다.

크기 정보 관리부(1150)는 크기 정보들을 저장하고, 저장한 크기 정보들을 파일 분산 관리 장치(1100)의 각 부에 제공한다. 크기 정보 관리부(1150)가 저장하는 크기 정보는 다양할 수 있다.

실시 예로서, 크기 정보 관리부(1150)는 파일 분할부(1110)에 의해 부가되는 더미 영역에 대한 정보(예를 들어, 더미 영역의 크기)를 저장할 수 있다. 또는, 크기 정보 관리부(1150)는 파일 분할부(1110)가 파일의 순서를 재배열할 때 기준이 되는 블록의 크기 정보를 저장할 수 있다. 또는, 크기 정보 관리부(1150)는 파일 분할부(1110)에 의해 분할되는 세그먼트들에 대한 정보(예를 들어, 세그먼트들의 크기)나 파일 조각 생성부(1120)에 의해 생성되는 파일 조각에 대한 정보(예를 들어, 파일 조각의 크기)를 저장할 수 있다.

실시 예로서, 크기 정보 관리부(1150)는 통신부(1130)를 통해 수신되는 파일 조각들로부터 크기 정보들(예를 들어, 파일 조각의 크기 정보, 파일 조각에 포함된 세그먼트의 크기 정보 또는, 파일 재배열을 수행하기 위한 블록의 크기 정보)을 읽어내어 저장할 수 있다.

도 3 이하에서는 본 발명에 따른 파일 분산 관리 장치 및 방법의 구체적인 동작 방법 및 내용을 상술한다.

이를 위해, 먼저, 설명을 위해 사용될 용어들에 대한 개념을 정의한다.

(1) 개념 정의

n: 분산 저장 장치들의 수

$M_1, M_2, \dots, M_n$: 총 n개의 분산 저장 장치들

k: 원본 파일을 성공적으로 복구하기 위해 필요한 최소의 파일 조각들의 수. 단, k는 0보다 크고 n이하인 정수이다.

F: 원본 파일

F': 원본 파일에 더미 영역을 부가한 수정 파일

F'': 블록 단위로 파일(원본 파일 또는 수정 파일)의 순서를 뒤섞는 재배열을 수행한 재배열된 파일

s: 수정 파일(F')을 구성하는 블록의 개수

b: 블록의 크기(byte)

e: 파일(원본 파일 또는 수정 파일)의 재배열을 수행하기 위한 키

$B'[i]$: 수정 파일(F')의 i번째 블록(즉, F'는 $B'[1] \parallel B'[2] \parallel \dots \parallel B'[s]$ 로 나타내어진다). 이때, $X = A \parallel B$ 의 의미는 병렬 연결(A 옆에 B가 위치한 형태로 X가 구성됨)을 의미한다.

$B''[i]$: 재배열된 파일(F'')의 i번째 블록(즉, F''는 $B''[1] \parallel B''[2] \parallel \dots \parallel B''[s]$ 로 나타내어진다)

$S[1], S[2], \dots, S[nC_{k-1}]$: 재배열된 수정 파일(F'')로부터 분할되는 세그먼트들(즉, F''는 $S[1] \parallel S[2] \parallel \dots \parallel S[nC_{k-1}]$ 로 나타내어질 수 있고, 각 세그먼트들은 복수의 블록으로 구성됨).

$$DT = \begin{pmatrix} DT_1 \\ \vdots \\ DT_n \end{pmatrix}$$

DT: Distribution Table ($n \times_{n-1} C_{k-1}$ 짜리 행렬로 되어 있음), $DT_i = (DT_{(i,1)}, DT_{(i,2)}, \dots, DT_{(i,q)})$ 이다(단, 여기서 q는 $n-1C_{k-1}$).

$N = (1, 2, 3, \dots, n)$

$W = \{1, 2, 3, \dots, nC_{k-1}\}$

T: N의 원소가 $n-k+1$ 개인 모든 부분집합의 집합

$A_i = \{C \in T \mid i \in C\}$

다음으로 본 발명의 실시 예에 따른 초기화 과정이 설명된다. 초기화 과정은 분산 테이블(Distribution Table,

DT)과 무작위로 재배열을 수행하기 위한 키(e)를 결정하기 위한 과정이다.

(2) 초기화 단계

1: 일대일 대응 함수 $\Phi : T \rightarrow W$ 를 결정한다.

2: (반복문 시작) $i=1$ 부터 n 까지 다음을 반복

3: $DT_{(i,1)}, DT_{(i,2)}, \dots, DT_{(i,q)}$ 에 $\{\Phi(b) \mid b \in B_i\}$ 의 원소들의 재배열을 대입(단, q 는 $n-1C_{k-1}$).

4: (반복문 끝)

5: e 의 값을 무작위로 결정한다.

위와 같은 방법으로 생성된 분산 테이블(DT) 및 키(e)는 파일 분산 관리 장치(1100, 도 1 참조)에 저장되고, 분산 저장 장치들(1100a, 1100b, ..., 1100n, 도 1 참조) 및 다른 인가된 장치에 제공된다.

다음은 본 발명의 실시 예에 따른 파일 분산 관리 방법이 설명된다.

도 3은 본 발명의 실시 예에 따른 파일 분산 관리 방법을 나타내는 순서도이다. 도 3을 참조하면, 파일 분산 관리 방법은 S110 단계 내지 S140 단계를 포함한다.

S110 단계 및 S120 단계는 원본 파일을 n 개의 파일 조각들로 분산하여 n 개의 분산 저장 장치들에 저장하는 분산 단계이다. S130 단계 및 S140 단계는 적어도 k 개의 분산 장치들로부터 k 개의 파일 조각들을 수신하여, 원본 파일을 복원하는 복원 단계이다.

S110 단계에서, 파일 분산 관리 장치(1100, 도 1 참조)는 원본 파일로부터 n 개의 파일 조각들을 생성한다. 이때, n 개의 파일 조각들은 다음과 같은 조건을 만족하도록 생성된다. n 개의 파일 조각들 중 임의의 k 개 파일 조각들로부터는 원본 파일이 성공적으로 복원될 수 있으나, 임의의 $k-1$ 개 파일 조각들로부터는 원본 파일이 성공적으로 복원될 수 없다. 파일 분산 관리 장치(1100)가 n 개의 파일 조각들을 생성하는 구체적인 방법은 도 4 이하에서 후술된다.

S120 단계에서, 파일 분산 관리 장치(1100)는 생성된 n 개의 파일 조각들을 n 개의 분산 저장 장치들(1100a, 1100b, ..., 1100n, 도 1 참조)에 분산 저장한다.

S120 단계가 완료되면, 원본 파일의 분산 단계가 완료된다. 이어서, 분산된 파일 조각들로부터 원본 파일을 복원하기 위해 S130 단계 이하가 수행된다.

S130 단계에서, 파일 분산 관리 장치(1100)는 k 개의 분산 저장 장치들(1100a, 1100b, ..., 1100k)로부터 k 개의 파일 조각들을 수신한다. 앞서 설명한 바와 같이 k 개의 파일 조각들에는 원본 파일을 구성하는 블록들이 모두 포함되어 있다.

S140 단계에서, 파일 분산 관리 장치(1100)는 수신된 k 개의 파일 조각들로부터 원본 파일을 복원해낸다. 파일 조각들로부터 원본 파일을 복원하는 과정은, 원본 파일에서 파일 조각들을 생성하는 과정의 역순과 유사하다. 파일 분산 관리 장치(1100)가 원본 파일을 복원하는 구체적인 방법은 도 5 이하에서 후술된다.

상기와 같은 파일 분산 관리 방법에 따르면, 하나의 파일이 여러 대의 기기에 나뉘어서 저장되므로 k 개 미만의 기기에서 데이터가 유출되었을 경우, 원래 파일의 복원이 불가능하므로 보안성이 향상되는 효과가 있다. 또한, k 개 이상의 기기가 있다고 하더라도, 파일 내용의 순서가 섞여 있으므로 원본 파일의 복구가 어려워 보안성이 더욱 향상될 수 있다. 또한, 파일의 일부가 손실되는 경우에도, 나머지 k 개의 파일을 통해 원본 파일을 복구할 수 있어 가용성이 향상될 수 있다. 또한, 조합적인 방법을 이용함으로써 연산량이 오로지 F 에만 비례하므로, 종래의 비밀분산법에 비해 연산량 및 복잡성이 감소할 수 있다. 또한, 사용자가 파일 블록의 크기를 임의로 조절할 수 있으므로, 분산된 파일의 보안 수준을 사용자가 결정할 수 있다.

다음은 본 발명에서 원본 파일을 복수의 파일 조각들로 분산하는 구체적인 방법이 설명된다.

도 4는 본 발명의 실시 예에 따른 파일 조각 생성 방법을 나타내는 순서도이다. 도 4를 참조하면, 파일 조각 생성 방법은 S210 단계 내지 S270 단계를 포함한다.

S210 단계에서, 파일 분산 관리 장치(1100, 도 1 참조)는 블록의 크기(b)를 결정한다. 블록의 크기(b)는 파일

분산 관리 장치(1100) 내에 저장된 별도의 크기 정보에 의해 결정될 수 있다. 또는, 블록의 크기(b)는 사용자에 의해 입력된 소정의 값에 따라 결정될 수 있다. 후술하겠지만, 블록의 크기가 작을수록 파일을 무작위로 재배열하는 단위가 작아지므로, 보안성이 향상될 수 있다. 반면에, 블록의 크기가 작으면, 파일 분산 관리 장치(1100)가 처리할 연산량이 증가하는 단점이 있다. 즉, 블록의 크기가 크게 설정되면 보안성은 낮아지나 연산량은 감소한다. 반면에 블록의 크기가 작게 설정되면 보안성은 높아지나 연산량은 증가한다. 본 발명에서, 사용자 또는 파일 분산 관리 장치(1100)는 필요에 따라 블록의 크기를 적절하게 선택할 수 있다.

S220 단계에서, 파일 분산 관리 장치(1100)는 원본 파일(F)의 크기가 블록의 크기(b)의 배수인지 판단한다. 원본 파일(F)의 크기가 블록의 크기(b)의 배수이면, 파일 조각 생성 방법은 S240 단계로 진행한다. 그렇지 않으면, 파일 조각 생성 방법은 S230 단계로 진행한다.

S230 단계에서, 파일 분산 관리 장치(1100)는 파일의 전체 크기가 블록의 크기(b)의 배수가 되도록, 원본 파일에 더미 영역을 부가하여 수정 파일(F')을 생성한다. 수정 파일(F')의 크기는 더미 영역에 의해 블록의 크기(b)의 배수가 된다.

S240 단계에서, 파일 분산 관리 장치(1100)는 파일(원본 파일 또는 수정 파일)을 블록 단위로 재배열한다. 다만, 원본 파일과 수정 파일의 차이는 더미 영역의 부가 여부에만 있고 이후의 처리 과정은 동일하므로, 이후의 단계에서는 양자를 혼용하여 수정 파일로 통칭하기로 한다. 이때, 파일 분산 관리 장치(1100)는 미리 결정된 키(e)를 이용하여, 수정 파일을 블록 단위로 무작위 재배열한다.

S250 단계에서, 파일 분산 관리 장치(1100)는 재배열된 수정 파일을 분할하여 nC_{k-1} 개의 세그먼트들을 생성한다.

S260 단계에서, 파일 분산 관리 장치(1100)는 생성된 nC_{k-1} 개의 세그먼트들로부터 n개의 파일 조각들을 생성한다. 단, n개의 파일 조각들 중 임의의 k개 파일 조각들을 선택할 때, 선택된 k개의 파일 조각들에는 nC_{k-1} 개의 세그먼트들이 모두 포함되도록, n개의 파일 조각들이 생성된다. 실시 예로서, n개의 파일 조각들 각각은 $n-1C_{k-1}$ 개의 세그먼트들을 포함할 수 있다. 따라서, n개의 파일 조각들은 서로 중복되는 세그먼트들을 포함할 수 있다. 이때, 각 파일 조각들에는 파일 조각을 구성하는 세그먼트들의 크기 정보가 부가될 수 있다.

S270 단계에서, 파일 분산 관리 장치(1100)는 n개의 파일 조각들을 n개의 분산 저장 장치들에 분산 저장한다.

한편, S210 단계 내지 S270 단계에서 설명된 파일 조각 생성 방법은 세그먼트의 크기를 정하는 방법에 따라 두 가지 실시 예로 구분될 수 있다. 하나는 각 세그먼트의 크기의 비율을 일정하게 하는 방법이고, 나머지 하나는 각 세그먼트의 크기들을 임의로 지정하는 방법이다. 두 가지 방법은 도 4의 파일 조각 생성 방법에 각각 선택적으로 적용될 수 있다. 이하에서는 세그먼트의 크기를 정하는 두 가지 방법에 따라, S210 단계 내지 S270 단계를 구현하는 구체적인 방법에 대해 설명한다.

1) 각 세그먼트의 크기의 비율을 일정하게 하는 방법

이 방법에서는, 각각의 세그먼트의 크기의 비율이 미리 결정되어 있고, 미리 결정된 세그먼트 크기의 비율에 따라 조각 파일을 생성한다.

먼저, $r_1, r_2, \dots, r_p$ (단, p는 nC_{k-1})를 각각 p개의 세그먼트 S[1], S[2], ..., S[p]의 길이의 비라고 하고, $r_1 + r_2 + \dots + r_p = 1$ 을 만족하는 값으로 정의한다. 그리고 $p_i = r_1 + r_2 + \dots + r_i$, $p_0 = 0$ 으로 정의한다.

이 경우, n개의 파일 조각들은 아래의 방법에 의해 생성된다.

1: 블록의 크기(b)를 결정한다.

2: s에 $\lceil f/b \rceil$ 값을 대입한다. 이때, s는 원본 파일(F)을 구성하는 블록의 개수이고, f는 원본 파일(F)의 크기를 바이트로 나타낸 값이고, $\lceil x \rceil$ 는 보다 x작지 않은 최소의 정수이다.

3: 파일 F의 앞에 $s \times b - f$ 바이트 크기의 더미 영역 부가하고, 더미 영역이 부가된 파일을 수정 파일(F')이라 한다. 이때, 더미 영역은 원본 파일(F)의 맨 앞에 부가될 수 있다.

4: 1, 2, ..., s를 키(e)를 이용하여 무작위로 재배열하고, 재배열된 결과를 $t_1, t_2, \dots, t_s$ 라 한다.

5: $F'' = B''[1] \parallel B''[2] \parallel \dots \parallel B''[s]$ 와 같이 재배열된 수정 파일(F'')을 생성한다. 이때, $t=1, 2, \dots, s$ 에 대하여, $B''[i]=B'[t_i]$ 를 만족하도록 파일 블록들이 재배열된다.

6: 재배열된 수정 파일(F'')을 분할하여 세그먼트들을 생성한다.

이때, $j=1, 2, \dots, nC_{k-1}$ 에 대하여,

$$S[j] = B''[\rho_{j-1} \times s + 1] \parallel B''[\rho_{j-1} \times s + 2] \parallel \dots \parallel B''[\rho_j \times s]$$

와 같이 각각의 세그먼트가 구성된다. 이때, $\lfloor x \rfloor$ 는 x 를 넘지 않는 최대 정수이다.

7: 파일 조각을 생성한다.

이때, $h=1, 2, \dots, n$ 에 대하여,

$$f_h = \mathcal{Z} \parallel S[DT_{(h,1)}] \parallel S[DT_{(h,2)}] \parallel \dots \parallel S[DT_{(h,n-1)C_{k-1}}]$$

와 같이 각각의 파일 조각이 구성된다.

여기서, f_h 는 생성된 n 개의 파일 조각이고($h=1, 2, \dots, n$), Z 는 원본 파일(F)의 크기를 나타내는 영역이다.

8: 생성된 파일 조각들($f_1, f_2, \dots, f_n$)을 n 개의 분산 저장 장치들($M_1, M_2, \dots, M_n$)에 각각 전송한다.

한편, 이때, 각 세그먼트들의 크기(또는, 크기의 비율)를 나타내는 정보는 원본 파일의 복원을 위해 파일 분산 관리 장치에 저장되거나, 또는 분산 저장 장치들에 전송될 수 있다.

2) 각 세그먼트의 크기를 임의로 지정하는 방법

이하에서는, 각 세그먼트의 크기를 임의로 지정하는 방법에 대해 설명한다. 이 방법에서는 세그먼트의 크기가 미리 결정되지 않고, 파일 조각을 생성할 때마다 세그먼트의 크기를 임의로 정한다.

1: 블록의 크기(b)를 결정한다.

2: s 에 $\lceil f/b \rceil$ 값을 대입한다. 이때, s 는 원본 파일(F)을 구성하는 블록의 개수이고, f 는 원본 파일(F)의 크기를 바이트로 나타낸 값이고, $\lceil x \rceil$ 는 보다 x 작지 않은 최소의 정수이다.

3: 파일 F 의 앞에 $s \times b - f$ 바이트 크기의 더미 영역 부가하고, 더미 영역이 부가된 파일을 수정 파일(F')이라 한다. 이때, 더미 영역은 원본 파일(F)의 맨 앞에 부가될 수 있다.

4: $1, 2, \dots, s$ 를 키(e)를 이용하여 무작위로 재배열하고, 재배열된 결과를 $t_1, t_2, \dots, t_s$ 라 한다.

5: $F'' = B''[1] \parallel B''[2] \parallel \dots \parallel B''[s]$ 와 같이 재배열된 수정 파일(F'')을 생성한다. 이때, $t=1, 2, \dots, s$ 에 대하여, $B''[i]=B'[t_i]$ 를 만족하도록 파일 블록들이 재배열된다.

6: nC_{k-1} 개의 양의 정수 $m_1, m_2, \dots, m_p$ (단, p 는 nC_{k-1})를 결정하되, 양의 정수들($m_1, m_2, \dots, m_p$)은 아래 식을 만족하도록 결정된다.

$$m_1 + m_2 + \dots + m_p = s \quad (\text{단, } p \text{는 } nC_{k-1})$$

그리고, $\mu_0=0$, $\mu_i=m_1+m_2+\dots+m_i$ 인 μ_i 를 정의한다($i=1, 2, \dots, nC_{k-1}$).

7: 재배열된 수정 파일(F'')을 분할하여 세그먼트들을 생성한다.

이때, $j=1, 2, \dots, nC_{k-1}$ 에 대하여,

$$S[j] = B''[\mu_{j-1} + 1] \parallel B''[\mu_{j-1} + 2] \parallel \dots \parallel B''[\mu_j]$$

와 같이 각각의 세그먼트를 구성한다.

8: 파일 조각을 생성한다.

이때, $h=1, 2, \dots, n$ 에 대하여,

$$f_h = \mathcal{Z} \parallel \mathcal{Z}_{DT(h,1)} \parallel \mathcal{Z}_{DT(h,2)} \parallel \dots \parallel \mathcal{Z}_{DT(h,n-1,C_{h-1})} \parallel S[DT(h,1)] \parallel S[DT(h,2)] \parallel \dots \parallel S[DT(h,n-1,C_{h-1})]$$

와 같이 각각의 파일 조각이 구성된다.

여기서, f_h 는 생성된 n 개의 파일 조각이고($h=1, 2, \dots, n$), Z 는 원본 파일(F)의 크기를 나타내는 영역이고, Z_j 는 m_j 값을 나타내는 영역이다.

9: 생성된 파일 조각들($f_1, f_2, \dots, f_n$)을 n 개의 분산 저장 장치들($M_1, M_2, \dots, M_n$)에 각각 전송한다.

한편, 위에서 설명한 방법들에서 키(e) 이용하여 블록들의 순서를 무작위로 재배열하는 단계는 다음과 같은 기준들에 따라 수행될 수 있다.

첫째, 가역적이어야 한다. 즉, 키(e)를 이용하여 재배열된 블록들을 원래 순서대로 복원할 수 있어야 한다.

둘째, 재배열을 수행할 때마다 항상 다른 재배열 결과가 나올 수 있어야 한다.

위와 같은 기준을 만족하는 재배열 방법에는 여러가지가 공개되어 있으며, 그러한 재배열 방법들은 당해 기술 분야에 널리 알려져 있으므로, 그에 대한 설명은 생략된다.

다음은 본 발명에서 원본 파일을 복수의 파일 조각들로 분산하는 구체적인 방법이 설명된다.

도 5는 본 발명의 실시 예에 따른 원본 파일 복원 방법을 나타내는 순서도이다. 도 5를 참조하면, 원본 파일 복원 방법은 S310 단계 내지 S340 단계를 포함한다.

S310 단계에서, 파일 분산 관리 장치(1100, 도 1 참조)는 수신된 파일 조각의 크기 정보를 이용하여 각 파일 조각을 세그먼트 단위로 분리한다. 이때, 크기 정보에는 파일 조각의 크기, 파일 조각을 구성하는 세그먼트들의 크기 또는 파일 조각에 포함된 더미 영역의 크기를 나타내는 정보가 포함될 수 있다. 실시 예로서, 크기 정보는 파일 분산 관리 장치(1100)에 저장될 수 있다.

S320 단계에서, 파일 분산 관리 장치(1100)는 분리된 세그먼트들을 연결하여 재배열된 수정 파일(F')을 복원한다.

S330 단계에서, 파일 분산 관리 장치(1100)는 재배열된 수정 파일(F')의 각 블록들의 순서를 키(e)를 이용하여 재배열함으로써, 수정 파일(F')을 복원한다.

S340 단계에서, 파일 분산 관리 장치(1100)는 수정 파일(F')로부터 더미 영역을 제거함으로써 원본 파일(F)을 복원한다. 실시 예로서, 파일 분산 관리 장치(1100)는 수정 파일(F')로부터 더미 영역을 제거하기 위해 크기 정보를 참조할 수 있다.

한편, S310 단계 내지 S340 단계에서 설명된 원본 파일 복원 방법은 앞서와 마찬가지로 세그먼트의 크기를 정하는 방법에 따라 두 가지 실시 예로 구분될 수 있다. 하나는 각 세그먼트의 크기의 비율을 일정하게 하는 방법이고, 나머지 하나는 각 세그먼트의 크기들을 임의로 지정하는 방법이다. 두 가지 방법은 도 5의 원본 파일 복원 방법에 각각 선택적으로 적용될 수 있다. 이하에서는 세그먼트의 크기를 정하는 두 가지 방법에 따라, S310 단계 내지 S340 단계를 구현하는 구체적인 방법에 대해 설명한다. 두 과정에 있어서 입력 데이터는 임의의 k 개의 파일 조각들이 된다.

1) 각 세그먼트의 크기의 비율을 일정하게 하는 방법

1: 각 파일 조각들로부터 원본 파일(F)의 크기를 나타내는 영역인 Z 를 제거하고, Z 로부터 원본 파일(F)의 크기(f)를 읽어낸다.

2: s 에 $\lceil f/b \rceil$ 값을 대입한다. 이때, s 는 원본 파일(F)을 구성하는 블록의 개수이고, f 는 원본 파일(F)의 크

기를 바이트로 나타낸 값이고, $\lceil x \rceil$ 는 보다 x 작지 않은 최소의 정수이다.

3: 아래 식에 따라 각 세그먼트들의 크기를 산출한다.

$$S[j] \text{의 크기: } [\rho_j \times s] - [\rho_{j-1} \times s]$$

여기서, $S[j]$, ρ_j 및 j 의 의미는 도 4에 대한 설명에서 사용된 각 기호들의 의미와 동일하다. 그리고, $\lfloor x \rfloor$ 는 x 를 넘지 않는 최대 정수이다.

4: 산출된 각 세그먼트들의 크기를 이용하여 파일 조각들로부터 세그먼트들을 분리하고, 분리된 세그먼트를 순서대로 연결하여 재배열된 수정 파일(F'')을 생성한다.

5: 재배열된 수정 파일(F'')의 블록들을 키(e)를 이용하여 재배열하고, 수정 파일(F')을 복원한다.

6: 수정 파일(F')로부터 더미 영역을 제거하여 원본 파일(F)을 복구한다. 예를 들어, 수정 파일(F')의 맨 앞에 더미 영역이 있는 경우, 수정 파일(F')의 마지막 f 바이트를 분리해내어 원본 파일(F)을 복원할 수 있다.

2) 각 세그먼트의 크기를 임의로 지정하는 방법

1: 각 파일 조각들로부터 원본 파일(F)의 크기를 나타내는 영역(Z)을 제거하고, Z 로부터 원본 파일(F)의 크기(f)를 읽어낸다. 또한, 각 세그먼트의 크기를 나타내는 영역(Z_j)들도 제거하고, Z_j 로부터 각 세그먼트들의 크기를 읽어낸다.

2: 읽어낸 각 세그먼트들의 크기를 이용하여 파일 조각들로부터 세그먼트들을 분리하고, 분리된 세그먼트를 순서대로 연결하여 재배열된 수정 파일(F'')을 생성한다.

5: 재배열된 수정 파일(F'')의 블록들을 키(e)를 이용하여 재배열하고, 수정 파일(F')을 복원한다.

6: 수정 파일(F')로부터 더미 영역을 제거하여 원본 파일(F)을 복구한다. 예를 들어, 수정 파일(F')의 맨 앞에 더미 영역이 있는 경우, 수정 파일(F')의 마지막 f 바이트를 분리해내어 원본 파일(F)을 복원할 수 있다.

도 6 및 도 7에서는 원본 파일로부터 파일 조각들을 생성하는 실시 예에 대한 설명이 도면과 함께 제공된다. 도 6 및 도 7에서, 생성되는 파일 조각들의 전체 개수는 5이고, 파일 조각들 중 임의의 3개로부터 원본 파일이 복원될 수 있는 것(즉, $n=5$, $k=3$)으로 가정한다. 한편, 이때, 분산 테이블(Distribution Table, DT)는 다음과 같다.

$$DT = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 7 & 8 & 9 \\ 1 & 4 & 5 & 7 & 8 & 10 \\ 2 & 4 & 6 & 7 & 9 & 10 \\ 3 & 5 & 6 & 8 & 9 & 10 \end{pmatrix}$$

도 6은 본 발명의 일 실시 예에 따라 파일 조각을 생성하는 방법을 나타내는 블록도이다. 도 6에서는 각 파일 조각을 구성하는 세그먼트들이 서로 일정한 크기의 비율을 갖는다.

먼저, 원본 파일(110)에 더미를 추가하여 수정 파일(120)이 생성된다. 그리고, 수정 파일(120)을 구성하는 블록들의 순서를 무작위로 재배열하여 재배열된 수정 파일(130)이 생성된다. 이때, 블록들의 순서를 무작위로 재배열하기 위해 미리 결정된 키(e)가 사용될 수 있다.

다음으로, 재배열된 수정 파일(130)을 10개의 세그먼트들로 분할한다. 이때, 각 세그먼트들은 크기의 비율이 일정하도록 분할되며, 각 세그먼트들을 분할하는 구체적인 방법은 위에서 설명한 것과 동일한 방법에 따른다.

그리고, 분할된 10개의 세그먼트들로부터 5개의 파일 조각들(150a, 150b, 150c, 150d, 150e)을 생성한다. 각각의 파일 조각들은 6개의 세그먼트들을 가지며, 임의의 3개의 파일 조각들로부터 원본 파일을 복원하기 위해 서로 중복되는 세그먼트들을 포함할 수 있다.

도 7은 본 발명의 다른 실시 예에 따라 파일 조각을 생성하는 방법을 나타내는 블록도이다. 도 7에서는 각 파일 조각을 구성하는 세그먼트들이 임의의 크기를 갖는다.

먼저, 원본 파일(210)에 더미를 추가하여 수정 파일(220)이 생성된다. 그리고, 수정 파일(220)을 구성하는 블록들의 순서를 무작위로 재배열하여 재배열된 수정 파일(230)이 생성된다. 이때, 블록들의 순서를 무작위로 재배

열하기 위해 미리 결정된 키(e)가 사용될 수 있다.

다음으로, 재배열된 수정 파일(230)을 10개의 세그먼트들로 분할한다. 이때, 각 세그먼트들의 크기는 임의로 결정되며, 각 임의의 크기를 갖도록 세그먼트들을 분할하는 구체적인 방법은 위에서 설명한 것과 동일한 방법에 따른다.

그리고, 분할된 10개의 세그먼트들로부터 5개의 파일 조각들(250a, 250b, 250c, 250d, 250e)을 생성한다. 각각의 파일 조각들은 6개의 세그먼트들을 가지며, 임의의 3개의 파일 조각들로부터 원본 파일을 복원하기 위해 서로 중복되는 세그먼트들을 포함할 수 있다.

이상에서 설명된 실시 예들은 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어, 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 이때, 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 또는 그것들의 조합을 포함할 수 있다. 컴퓨터 판독 가능한 기록 매체는 예를 들어, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크 (floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 저장 장치를 포함할 수 있다. 이러한 저장 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈에 의해 작동하도록 구성될 수 있으며, 그 반대도 마찬가지이다. 또한, 여기서 설명된 프로그램 명령어는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

본 발명의 상세한 설명에서는 구체적인 실시 예를 들어 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한 각 실시 예는 여러 가지 형태로 변형될 수 있다.

또한, 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 발명의 범위는 상술한 실시 예에 국한되어 정해져서는 안되며 후술하는 특허 청구범위뿐만 아니라 이 발명의 특허 청구 범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

1000: 파일 분산 관리 시스템

1100: 파일 관리 장치

1100a, 1100b, ..., 1100n: 분산 저장 장치들

1110: 파일 분할부

1120: 파일 조각 생성부

1130: 통신부

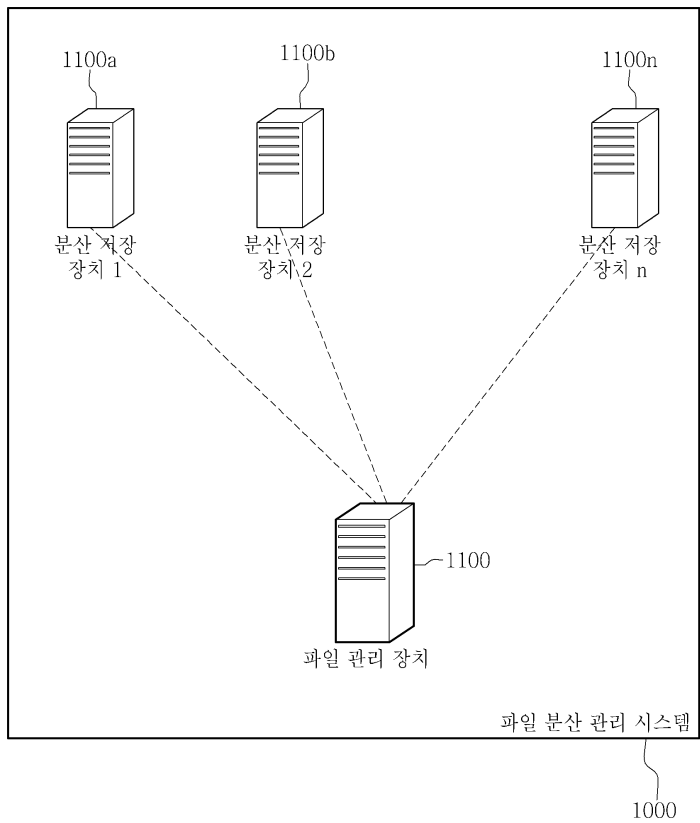
1140: 파일 복원부

1150: 크기 정보 관리부

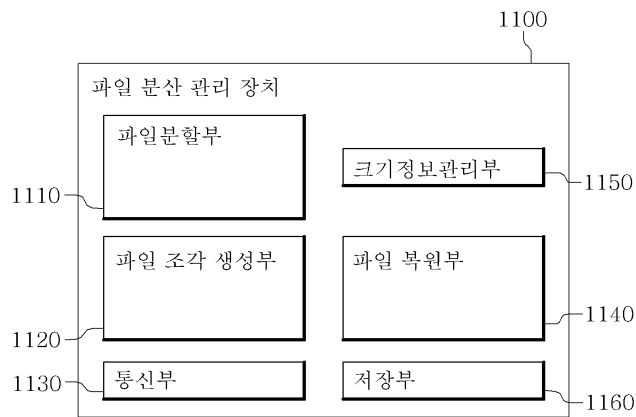
1160: 저장부

도면

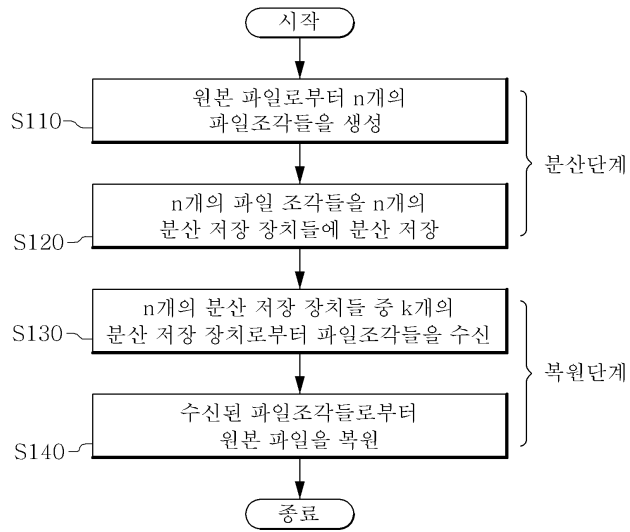
도면1



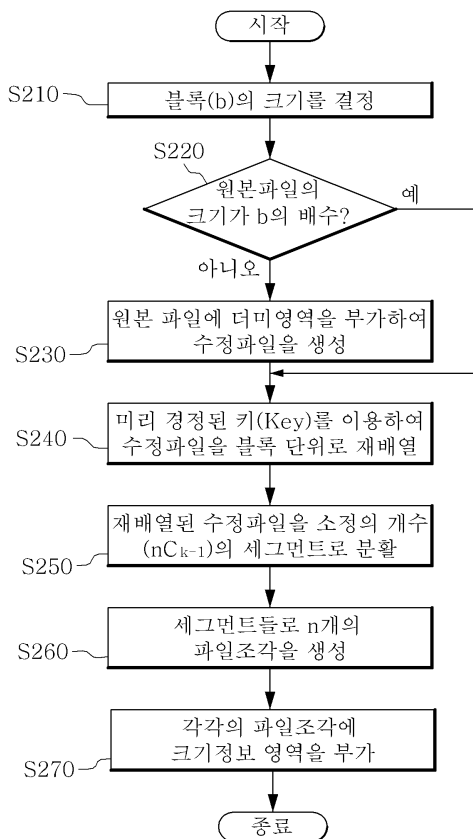
도면2



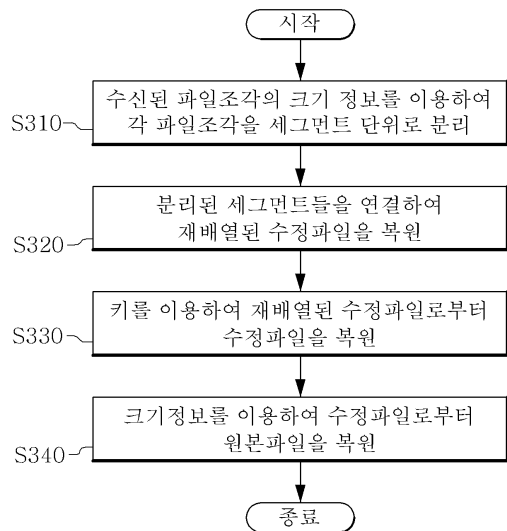
도면3



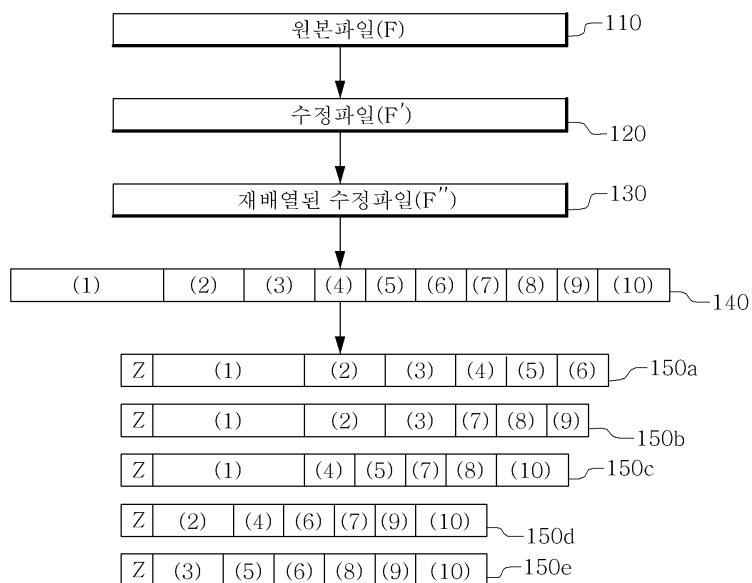
도면4



도면5



도면6



도면7

