(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년08월23일
(11) 등록번호 10-1285281
(24) 등록일자 2013년07월05일

(51) 국제특허분류(Int. Cl.)
G06F 21/60 (2013.01) *G06F 15/16* (2006.01)
(21) 출원번호 10-2012-0032560
(22) 출원일자 2012년03월29일
심사청구일자 2012년03월29일
(56) 선행기술조사문헌
KR101082917 B1*
KR1020100081873 A*
KR101105205 B1
논문(2012.08)
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
주식회사 씨디에스
강원도 춘천시 삭주로145번길 46, 아이씨티 융합기술벤처타운 111호 112호 (후평동)
관동대학교산학협력단
강원도 강릉시 범일로579번길 24, 관동대학내 (내곡동)
(72) 발명자
이병관
경기도 하남시 대청로116번길 30 은행쌍용아파트 120동 702호
정은희
강원도 삼척시 정상로 35 현진에버빌아파트 107동 1503호
양승해
강원도 춘천시 후석로326번길 36-7 세경2차아파트 3-403
(74) 대리인
박종옥

전체 청구항 수 : 총 17 항 심사관 : 이석형

(54) 발명의 명칭 **자가조직 저장매체의 보안 시스템 및 그 방법**

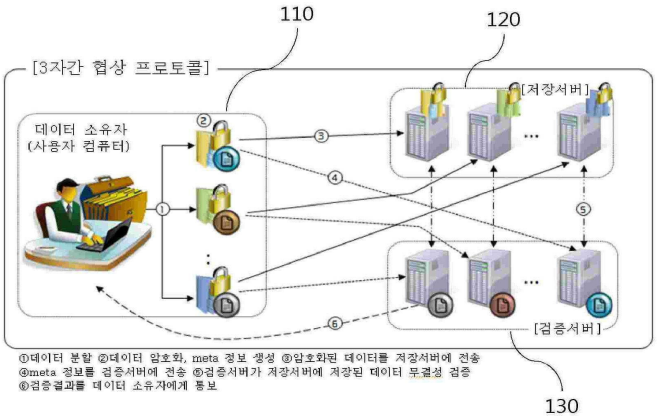
(57) 요 약

본 발명은 자가조직 저장매체의 보안 시스템 및 그 방법에 관한 것이다.

본 발명의 저장매체의 보안 방법은 데이터 소유자(사용자 컴퓨터)가 데이터를 저장할 크기로 분할하여 자신의 비밀키로 암호화한 후, 암호화된 데이터를 시스템 마스터 키로 다시 한 번 암호화하여 암호화된 데이터들을 저장서버로 저장한다. 그리고, 비밀키로 암호화된 데이터를 사용자 컴퓨터에 의해 해시(hash) 함수로 해시하여 메타(meta) 정보를 생성하여 검증 서버로 전송한다. 또한, 저장 서버에 저장되어 있는 데이터의 무결성을 검사하기 위해 검증 서버가 저장 서버로 검사 요청을 하면, 저장 서버는 자신이 저장하고 있는 암호화된 데이터를 해시 함수로 해시한 후 해시값을 검증 서버로 전달하고, 검증 서버는 저장 서버로부터 전송받은 해시값과 자신이 저장하고 있는 메타 정보를 비교하여 데이터의 무결성 여부를 검사한다.

이와 같은 본 발명에 의하면, 인프라 클라우딩 환경의 저장매체에 데이터를 저장함에 있어서 3자간 협상 프로토콜을 통해 데이터를 분할하여 분산 저장하고, 저장된 데이터를 권한을 위임받은 제3자가 검증하도록 함으로써 데이터에 대한 안전성, 무결성 및 신뢰도를 한층 향상시킬 수 있다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 1425068980
부처명 중소기업청
연구사업명 산학연 공동기술개발사업
연구과제명 인프라 클라우딩(Infra Clouding) 환경에서 자기조직 저장매체 보안을 위한 3자간 협상 프로토콜 개발
주관기관 관동대학교 산학협력단
연구기간 2011.06.01 ~ 2012.05.31

특허청구의 범위

청구항 1

자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하고, 분할된 데이터를 암호화하며, 암호화된 결과에 대한 메타(meta) 정보를 생성하는 사용자 컴퓨터;

상기 사용자 컴퓨터로부터 암호화된 데이터를 전송받아 상호 독립된 복수의 서버에 각각 분산하여 저장하는 저장 서버; 및

상기 사용자 컴퓨터로부터 메타 정보를 전송받아, 그 전송받은 메타 정보와 상기 저장 서버에 분산 저장된 데이터를 각각 비교하여 데이터의 무결성 여부를 검사하고, 그 검사 결과를 상기 사용자 컴퓨터측에 통보하는 검증 서버를 포함하며,

상기 메타 정보는 상기 자신의 비밀키로 암호화된 데이터를 해시 함수(hash function)로 해시하여 생성된 정보인 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 2

제1항에 있어서,

상기 사용자 컴퓨터는 상기 분할된 데이터를 자신의 비밀키로 암호화하고, 그 암호화된 데이터를 시스템 마스터 키로 다시 한 번 더 암호화하는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 검증 서버는 상기 메타 정보와 상기 저장 서버에 분산 저장된 데이터를 각각 비교하여, 비교 값이 참이면 데이터가 손상없이 잘 보관되고 있는 것으로 판단하고, 비교 값이 거짓이면 데이터가 손상된 것으로 판단하는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 5

제1항에 있어서,

상기 사용자 컴퓨터에는 데이터 소유자(사용자 컴퓨터)의 파일 할당 테이블 (User File Allocation Table, UFAT), 상기 저장 서버에는 저장 서버의 파일 할당 테이블(Storage File Allocation Table, SFAT), 상기 검증 서버에는 검증 서버의 파일 할당 테이블(Verification File Allocation Table, VFAT)이 각각 설치되는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 6

제5항에 있어서,

상기 UFAT(사용자 파일 할당 테이블)에는 파일명칭, 저장서버 ID, 검증서버 ID가 저장되고, 상기 SFAT(저장서버 파일 할당 테이블)에는 파일명칭, 사용자 ID, 검증서버 ID가 저장되며, 상기 VFAT(검증서버 파일 할당 테이블)에는 파일명칭, 저장서버 ID, 사용자 ID가 저장되는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 7

제1항에 있어서,

상기 사용자 컴퓨터에는 사용자 컴퓨터, 상기 저장 서버, 상기 검증 서버 간의 시스템 마스터 키를 설정하고, 상기 사용자 컴퓨터, 저장 서버 및 검증 서버에 파일 할당 테이블을 각각 생성하는 SSM(System Setup Module)이

탑재되는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 8

제7항에 있어서,

상기 시스템 마스터 키는 3자간 협상 프로토콜의 구성요소인 데이터의 소유자(사용자 컴퓨터), 저장 서버, 검증 서버 간에 공유하는 키인 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 9

제7항 또는 제8항에 있어서,

상기 시스템 마스터 키는 상기 사용자 컴퓨터(데이터 소유자), 저장 서버, 검증 서버 간에 전송되는 데이터를 암호화 및 복호화할 때 사용하거나, 3자간 협상 프로토콜의 구성요소들 상호 간의 신분을 확인하고자 할 때 사용되는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 10

제1항에 있어서,

상기 사용자 컴퓨터는 상기 사용자 컴퓨터(데이터의 소유자)가 상기 저장 서버로 암호화하여 전송한 각각의 분할 데이터에 대한 메타 정보를 생성하여 데이터를 검증할 검증 노드에 전송하는 MGM(Meta Generation Module)을 갖는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 11

제1항에 있어서,

상기 검증 서버는, SOS(security operating service) 시스템의 시스템 마스터 키(SM-key)와 임의의 값인 난스(nonce)를 이용하여 상기 검증 서버가 상기 저장 서버에 저장되어 있는 데이터의 무결성을 검증하여 그 결과를 상기 사용자 컴퓨터(데이터 소유자)로 전송하는 DVM(Data Verification Module)을 갖는 것을 특징으로 하는 자가조직 저장매체의 보안 시스템.

청구항 12

사용자 컴퓨터, 저장 서버 및 검증 서버를 구비하는 자가조직 저장매체의 보안 시스템을 이용한 보안 방법으로서,

a) 상기 사용자 컴퓨터에 의해 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하는 단계;

b) 상기 사용자 컴퓨터에 의해 상기 분할된 데이터를 자신의 비밀키로 암호화하는 단계;

c) 상기 사용자 컴퓨터에 의해 상기 비밀키로 암호화된 데이터를 시스템 마스터 키로 다시 한번 암호화하는 단계;

d) 상기 사용자 컴퓨터에 의해 상기 비밀키로 암호화된 데이터를 해시 (hash) 함수로 해시하여 메타(meta) 정보를 생성하는 단계;

e) 상기 비밀키 및 시스템 마스터 키로 암호화된 데이터들을 상기 사용자 컴퓨터에 의해 상기 저장 서버로 전송하여 분산 저장하는 단계;

f) 상기 단계 d)에서 생성된 메타 정보를 상기 사용자 컴퓨터에 의해 상기 검증 서버로 전송하는 단계;

g) 상기 저장 서버에 저장되어 있는 데이터의 무결성을 검사하기 위해 상기 검증 서버가 상기 저장 서버로 검사 요청을 전달하는 단계;

h) 상기 저장 서버에 의해 자신이 저장하고 있는 암호화된 데이터를 해시 함수로 해시한 후 그 결과인 해시값을 검증 서버로 전달하는 단계;

i) 상기 검증 서버에 의해 상기 저장 서버로부터 전송받은 해시값과 자신이 저장하고 있는 메타 정보를 비교하여 데이터의 무결성 여부를 검사하는 단계; 및

j) 상기 검사 결과를 상기 검증 서버에 의해 상기 사용자 컴퓨터로 통보하는 단계를 포함하는 자가조직 저장매체의 보안 방법.

청구항 13

제12항에 있어서,

상기 단계 i)에서, 상기 검증 서버는 상기 해시값과 메타 정보와의 비교 값이 참이면 데이터가 위조나 변조 없이 잘 보관되고 있는 것으로 판단하고, 비교 값이 거짓이면 데이터가 손상된 것으로 판단하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

청구항 14

제12항에 있어서,

상기 사용자 컴퓨터는 분할된 데이터의 무결성에 대한 정보를 상기 검증 서버로부터 주기적으로 수신하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

청구항 15

제12항에 있어서,

상기 단계 e)에서 상기 저장 서버는 전송받은 암호화된 데이터를 시스템 마스터 키(SM_key)로 복호화한 후, 수신된 암호문의 무결성을 검증하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

청구항 16

제15에 있어서,

상기 수신된 암호문의 무결성 검증이 유효하면, 상기 저장 서버는 데이터 소유자의 비밀키로 암호화된 데이터를 그대로 저장하면서, SFAT(저장 파일 할당 테이블)에 데이터 정보와 데이터 소유자(사용자 컴퓨터) ID, 검증 서버 ID를 저장하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

청구항 17

제15에 있어서,

상기 수신된 암호문의 무결성 검증이 유효하지 않으면, 상기 사용자 컴퓨터(데이터 소유자)와 상기 검증 서버로 데이터 전송에 에러가 발생하였음을 통지하고, 데이터 수신을 재요청하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

청구항 18

제12항에 있어서,

상기 사용자 컴퓨터는 상기 저장 서버에 분산되어 있던 암호화된 데이터를 수집하여 복호화한 후 완성된 데이터로 병합하고, 병합된 데이터의 무결성을 한 번 더 검증하는 단계를 더 포함하는 것을 특징으로 하는 자가조직 저장매체의 보안 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 자가조직 저장매체의 보안 시스템 및 그 방법에 관한 것으로서, 더 상세하게는 데이터를 인프라 클라우딩 환경의 저장매체에 저장함에 있어서, 데이터를 분할하여 분산 저장하고, 저장된 데이터는 권한을 위임받은 제3자가 검증하도록 함으로써 데이터에 대한 안전성, 무결성 및 신뢰도를 향상시킬 수 있는 자가조직 저장매체의 보안 시스템 및 그 방법에 관한 것이다.

배 경 기 술

[0002] 오늘날, 인터넷상에서 대용량 데이터의 급속한 유통 및 지속적인 증가로 인해 서비스 시스템의 비용과 확장성이 인터넷 서비스 업체의 경쟁력 확보에 중요한 요소가 되고 있다. 구글, 야후 등 글로벌 인터넷 서비스 업체들은 인터넷 서비스 플랫폼의 중요성을 인식하고, 자체적인 연구/개발을 수행하여, 저가 상용 노드를 기반으로 한 대규모 클러스터 기반의 분산 컴퓨팅 플랫폼 기술을 개발하여 활용하고 있다.

[0003] 이와 같이 인터넷 서비스 데이터량의 지속적인 증가로 대량의 원시 데이터로부터 정보를 가공 처리하는 과정, 체계화된 정보의 저장 및 관리, 그리고 유용한 정보를 추출하기 위한 분석 등에 분산 컴퓨팅 기술을 적용하는 움직임이 활발히 진행되고 있다.

[0004] 대용량 데이터의 처리, 저장 및 관리가 필요한 대표적인 애플리케이션으로는 인터넷 서비스 분야뿐만 아니라, 예를 들면, 비즈니스 인텔리전스 등 다른 응용 영역으로 확대하여 클라우드 서비스로 제공하려는 비즈니스 모델이 제시되고 있다. 클라우드 서비스는 크게 컴퓨팅 인프라를 서비스로 제공하는 IaaS(Infrastructure as a Services), 응용을 개발 및 운영할 수 있는 소프트웨어 플랫폼을 서비스로 제공하는 PaaS(Platform as a Service), 개인이나 기업에서 필요로 하는 소프트웨어를 서비스로 제공, 이용할 수 있게 하는 SaaS(Software as a Service) 등으로 구분된다. 특히, PaaS 서비스는 애플리케이션 분야에 따라 소프트웨어 플랫폼의 구성 요소가 달라진다. 현재 PaaS 서비스로 논의되고 있는 서비스 중 하나가 대규모 데이터 관리 및 처리 서비스이다. 이는 데이터량이 지속적으로 증가하고 있어 많은 애플리케이션에서 대규모 데이터 처리 및 관리가 필요해지고 있는 상황이기 때문이다.

[0005] 한편, 이상과 같은 대용량 데이터의 처리 및 저장과 관련하여, 종래의 클라우드 스토리지 서비스에 있어서는 데이터를 인프라 클라우딩 환경의 저장매체에 저장할 때, 데이터를 한 곳에 집중하여 저장한다. 따라서 데이터에 대한 안전성(보안성)이 문제시 되고 있다. 또한, 데이터에 대한 무결성 및 신뢰도도 상대적으로 낮을 수밖에 없다는 것이 하나의 문제점으로 지적되고 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 상기와 같은 문제점을 개선하기 위하여 창출된 것으로서, 인프라 클라우딩 환경의 저장매체에 데이터를 저장함에 있어서, 3자간 협상 프로토콜을 통해 데이터를 분할하여 분산 저장하고, 저장된 데이터를 제3자가 검증하도록 함으로써 데이터에 대한 안전성, 무결성 및 신뢰도를 향상시킬 수 있는 자가조직 저장매체의 보안 시스템 및 그 방법을 제공함에 그 목적이 있다.

과제의 해결 수단

[0007] 상기의 목적을 달성하기 위하여 본 발명에 따른 자가조직 저장매체의 보안 시스템은,

[0008] 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하고, 분할된 데이터를 암호화하며, 암호화된 결과에 대한 메타(meta) 정보를 생성하는 사용자 컴퓨터;

[0009] 상기 사용자 컴퓨터로부터 암호화된 데이터를 전송받아 상호 독립된 복수의 서버에 각각 분산하여 저장하는 저장 서버; 및

[0010] 상기 사용자 컴퓨터로부터 메타 정보를 전송받아, 그 전송받은 메타 정보와 상기 저장 서버에 분산 저장된 데이터를 각각 비교하여 데이터의 무결성 여부를 검사하고, 그 검사 결과를 상기 사용자 컴퓨터측에 통보하는 검증 서버를 포함하는 점에 그 특징이 있다.

[0011] 여기서, 상기 사용자 컴퓨터는 상기 분할된 데이터를 자신의 비밀키로 암호화하고, 그 암호화된 데이터를 시스템 마스터 키로 다시 한 번 더 암호화한다.

[0012] 또한, 상기 사용자 컴퓨터는 상기 자신의 비밀키로 암호화된 데이터를 해시 함수(hash function)로 해시하여 메타 정보를 생성한다.

[0013] 또한, 상기 검증 서버는 상기 메타 정보와 상기 저장 서버에 분산 저장된 데이터를 각각 비교하여, 비교 값이 참이면 데이터가 위조나 변조 없이 잘 보관되고 있는 것으로 판단하고, 비교 값이 거짓이면 데이터가 손상된 것으로 판단한다.

[0014] 또한, 상기의 목적을 달성하기 위하여 본 발명에 따른 자가조직 저장매체의 보안 방법은,

[0015] 사용자 컴퓨터, 저장 서버 및 검증 서버를 구비하는 자가조직 저장매체의 보안 시스템을 이용한 보안 방법으로서,

[0016] a) 상기 사용자 컴퓨터에 의해 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하는 단계;

[0017] b) 상기 사용자 컴퓨터에 의해 상기 분할된 데이터를 자신의 비밀키로 암호화하는 단계;

[0018] c) 상기 사용자 컴퓨터에 의해 상기 비밀키로 암호화된 데이터를 시스템 마스터 키로 다시 한번 암호화하는 단계;

[0019] d) 상기 사용자 컴퓨터에 의해 상기 비밀키로 암호화된 데이터를 해시 (hash) 함수로 해시하여 메타(meta) 정보를 생성하는 단계;

[0020] e) 상기 비밀키 및 시스템 마스터 키로 암호화된 데이터들을 상기 사용자 컴퓨터에 의해 상기 저장 서버로 전송하여 분산 저장하는 단계;

[0021] f) 상기 단계 d)에서 생성된 메타 정보를 상기 사용자 컴퓨터에 의해 상기 검증 서버로 전송하는 단계;

[0022] g) 상기 저장 서버에 저장되어 있는 데이터의 무결성을 검사하기 위해 상기 검증 서버가 상기 저장 서버로 검사 요청을 전달하는 단계;

[0023] h) 상기 저장 서버에 의해 자신이 저장하고 있는 암호화된 데이터를 해시 함수로 해시한 후 그 결과인 해시값을 검증 서버로 전달하는 단계;

[0024] i) 상기 검증 서버에 의해 상기 저장 서버로부터 전송받은 해시값과 자신이 저장하고 있는 메타 정보를 비교하여 데이터의 무결성 여부를 검사하는 단계; 및

[0025] j) 상기 검사 결과를 상기 검증 서버에 의해 상기 사용자 컴퓨터로 통보하는 단계를 포함하는 점에 그 특징이 있다.

[0026] 여기서, 상기 단계 i)에서, 상기 해시값과 메타 정보와의 비교 값이 참이면 데이터가 위조나 변조 없이 잘 보관되고 있는 것으로 판단하고, 비교 값이 거짓이면 데이터가 손상된 것으로 판단한다.

발명의 효과

[0027] 이와 같은 본 발명에 의하면, 인프라 클라우딩 환경의 저장매체에 데이터를 저장함에 있어서, 3자간 협상 프로토콜을 통해 데이터를 분할하여 분산 저장하고, 저장된 데이터를 권한을 위임받은 제3자가 검증하도록 함으로써 데이터에 대한 안전성, 무결성 및 신뢰도를 한층 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

[0028] 도 1은 본 발명에 따른 자가조직 저장매체의 보안 시스템의 개략적인 구성도.

도 2는 3자간 협상 프로토콜의 구성요소들 간의 SM-key(System Master Key)를 생성하는 과정을 설명하는 도면.

도 3은 사용자 컴퓨터, 저장 서버 및 검증 서버의 각 파일 할당 테이블의 구조를 보여주는 도면.

도 4는 본 발명에 따른 자가조직 저장매체의 보안 방법의 실행 과정을 보여주는 흐름도.

도 5는 본 발명에 따른 자가조직 저장매체의 보안 방법에 있어서, 데이터 분할의 과정을 설명하는 흐름도.

도 6은 본 발명에 따른 자가조직 저장매체의 보안 방법에 있어서, 데이터 암호화와 무결성 검증 과정을 설명하는 도면.

도 7은 MGM(Meta Generation Module)의 데이터 처리과정을 설명하는 도면.

도 8은 DVM(Data Verification Module)에 의한 데이터 무결성 검증 절차 과정을 설명하는 도면.

발명을 실시하기 위한 구체적인 내용

[0029] 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정되어 해석되지 말아야 하며,

발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야 한다.

[0030] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있다는 것을 의미한다. 또한, 명세서에 기재된 "…부", "…기", "모듈", "장치" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0031] 이하 첨부된 도면을 참조하여 본 발명의 실시 예를 상세히 설명한다.

[0032] 도 1은 본 발명에 따른 자가조직 저장매체의 보안 시스템의 구성을 개략적으로 보여주는 도면이다.

[0033] 도 1을 참조하면, 본 발명에 따른 자가조직 저장매체의 보안 시스템은 사용자 컴퓨터(110), 저장 서버(120) 및 검증 서버(130)를 포함한다.

[0034] 상기 사용자 컴퓨터(110)는 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하고, 분할된 데이터를 암호화하며, 암호화된 결과에 대한 메타 (meta) 정보를 생성한다. 즉, 데이터 소유자는 자신의 컴퓨터(이를 '사용자 컴퓨터'라 칭함)를 이용하여 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할하고, 분할된 데이터를 암호화하는 것이다.

[0035] 상기 저장 서버(120)는 상기 사용자 컴퓨터(110)로부터 암호화된 데이터를 전송받아 상호 독립된 복수의 서버에 각각 분산하여 저장한다.

[0036] 상기 검증 서버(130)는 상기 사용자 컴퓨터(110)로부터 메타 정보를 전송받아, 그 전송받은 메타 정보와 상기 저장 서버(120)에 분산 저장된 데이터를 각각 비교하여 데이터의 무결성 여부를 검사하고, 그 검사 결과를 상기 사용자 컴퓨터 (110) 측에 통보한다.

[0037] 여기서, 상기 사용자 컴퓨터(110)는 상기 분할된 데이터를 자신의 비밀키로 암호화하고, 그 암호화된 데이터를 시스템 마스터 키(System Master Key)로 다시 한 번 더 암호화한다.

[0038] 또한, 상기 사용자 컴퓨터(110)는 상기 자신의 비밀키로 암호화된 데이터를 해시 함수(hash function)로 해시하여 메타 정보를 생성한다.

[0039] 또한, 상기 사용자 컴퓨터(110)에는 사용자 컴퓨터(110), 저장 서버(120), 검증 서버(130) 간의 시스템 마스터 키를 설정하고, 사용자 컴퓨터(110), 저장 서버(120) 및 검증 서버(130)에 파일 할당 테이블을 각각 생성하는 SSM(System Setup Module)이 탑재된다. 여기서, 상기 시스템 마스터 키에 대하여 부연 설명해 보기로 한다.

[0040] 시스템 마스터 키(System Master Key: SM-key)는 3자간 협상 프로토콜의 구성요소인 데이터의 소유자(사용자 컴퓨터(110)), 저장 서버(120), 검증 서버(130) 간에 공유하는 키를 말한다. 이와 같은 SM-key는 데이터 소유자(사용자 컴퓨터(110)), 저장 서버(120), 검증 서버(130) 간에 전송되는 데이터를 암호화 및 복호화 할 때 사용하거나, 3자간 협상 프로토콜의 구성요소들 상호 간의 신분을 확인하고자 할 때 사용된다.

[0041] 도 2는 3자간 협상 프로토콜의 구성요소들 간의 SM-key를 생성하는 과정을 설명하는 도면이다.

[0042] 도 2를 참조하면, 데이터 소유자는 타원곡선 암호 알고리즘을 이용하여 암호화에 필요한 파라미터를 생성하고, 그것을 저장 매체 내의 구성 노드에 공개하면, 각 노드들은 자신의 비밀키를 이용해 공개키를 생성한다. 각 노드는 EC-DH (Elliptic Curve Diffie-Hellman) 알고리즘을 이용하여 상대방의 공개키를 자신의 비밀키로 한 번 더 연산을 하며, 이에 의해 3자간 공유 비밀키인 SM-Key가 생성된다. 이 공유 비밀키를 사용자 컴퓨터(110), 저장 서버(120) 및 검증 서버(130)의SM-Key로 사용하며, 이 키로 3자간 협상 프로토콜의 구성 요소 간의 신분 인증에도 사용한다.

[0043] 한편, 이상과 같은 본 발명의 자가조직 저장매체의 보안 시스템에 있어서의 각 구성요소, 즉 사용자 컴퓨터(110), 저장 서버(120) 및 검증 서버(130)에는 각각 파일 할당 테이블이 설치된다.

[0044] 즉, 각 구성요소의 특성에 따라 사용자 컴퓨터(110)에는 데이터 소유자(사용자 컴퓨터(110))의 파일 할당 테이블(User File Allocation Table, UFAT), 저장 서버(120)에는 저장 서버(120)의 파일 할당 테이블(Storage File Allocation Table, SFAT), 검증 서버(130)에는 검증 서버(130)의 파일 할당 테이블(Verification File Allocation Table, VFAT)이 각각 설치된다. 이와 같은 각 파일 할당 테이블에는 데이터 소유자(사용자 컴퓨터(110)), 저장 서버(120), 검증 서버(130)에 대한 정보가 각각 저장된다. 이는 데이터 검색 시에 가용성을 향상

시키기 위한 것이다.

[0045] 도 3은 사용자 컴퓨터, 저장 서버 및 검증 서버의 각 파일 할당 테이블의 구조를 보여주는 도면이다.

[0046] 도 3을 참조하면, UFAT(사용자 파일 할당 테이블)에는 파일명칭, 저장서버 ID, 검증서버 ID가 저장되어 있고, SFAT(저장서버 파일 할당 테이블)에는 파일명칭, 사용자 ID, 검증서버 ID가 저장된다. 그리고, VFAT(검증서버 파일 할당 테이블)에는 파일명칭, 저장서버 ID, 사용자 ID가 저장된다.

[0047] 이 테이블들을 이용하여 사용자는 UFAT에서 분산 저장된 파일들의 위치를 알 수 있으며, 검증 서버(130)는 VFAT을 이용해 파일이 저장되어 있는 저장 서버(120)에 파일의 무결성 검증 요청을 한다.

[0048] 또한, 상기 사용자 컴퓨터(110)는 데이터 소유자(사용자 컴퓨터(110))가 데이터를 분할하고 데이터 소유자의 비밀키로 암호화한 후, 그것을 SM-key로 한 번 더 암호화한 후 각각의 저장 서버(120)에 전송하거나, 분할된 데이터들을 병합하는 역할을 하는 DMM(Division Merge Module)을 갖는다. 이때, DMM은 원본 데이터, 분할된 데이터들, 저장 서버(120)들에 대한 정보를 데이터 소유자의 파일 할당 테이블(UFAT)에 저장한다.

[0049] 또한, 상기 사용자 컴퓨터(110)는 데이터의 소유자(사용자 컴퓨터(110))가 저장 서버(120)로 암호화하여 전송한 각각의 분할 데이터에 대한 meta-정보를 생성하여 데이터를 검증할 검증 노드에 전송하는 역할을 하는 MGM(Meta Generation Module)을 갖는다. 이때 물론, 데이터 소유자(사용자 컴퓨터(110))는 어떤 검증 서버에 meta-정보를 전송하였는지를 데이터 소유자의 파일 할당 테이블(UFAT)에 저장한다.

[0050] 또한, 상기 검증 서버(130)는 상기 메타 정보와 상기 저장 서버(120)에 분산 저장된 데이터를 각각 비교하여, 비교 값이 참이면 데이터가 위조나 변조 없이 잘 보관되고 있는 것으로 판단하고, 비교 값이 거짓이면 데이터가 손상된 것으로 판단한다.

[0051] 여기서, 이상과 같은 검증 서버(130)는, 시스템 마스터 키인 SM-key와 임의의 값인 nonce를 이용하여 검증 서버(130)가 저장 서버(120)에 저장되어 있는 데이터의 무결성을 검증하여 그 결과를 데이터 소유자(사용자 컴퓨터(110))로 전송하는 역할을 하는 DVM(Data Verification Module)을 갖는다. 이때, 바람직하게는 임의의 값인 난스(nonce)가 검증을 요청할 때마다 새로운 값으로 생성되도록 한다. 이는 플러딩 공격과 재전송 공격을 방지하기 위한 것이다.

[0052] 그러면, 이상과 같은 구성을 갖는 본 발명에 따른 자가조직 저장매체의 보안시스템에 의한 자가조직 저장매체의 보안 방법에 대하여 설명해 보기로 한다.

[0053] 도 4는 본 발명에 따른 자가조직 저장매체의 보안 방법의 실행 과정을 보여주는 흐름도이다.

[0054] 도 4를 참조하면, 본 발명에 따른 자가조직 저장매체의 보안 방법은, 전술한 바와 같은 사용자 컴퓨터(110), 저장 서버(120) 및 검증 서버(130)를 구비하는 자가조직 저장매체의 보안 시스템에 의한 보안 방법으로서, 먼저 상기 사용자 컴퓨터(110)에 의해 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할한다(단계 S401).

[0055] 그런 후, 상기 사용자 컴퓨터(110)에 의해 상기 분할된 데이터를 자신의 비밀키로 암호화한다(단계 S402).

[0056] 그리고, 상기 사용자 컴퓨터(110)에 의해 상기 비밀키로 암호화된 데이터를 시스템 마스터 키로 다시 한 번 암호화한다(단계 S403).

[0057] 또한, 상기 사용자 컴퓨터(110)에 의해 상기 시스템 마스터 키로 암호화된 데이터를 해시(hash) 함수로 해시하여 메타(meta) 정보를 생성한다(단계 S404).

[0058] 그런 다음, 상기 비밀키 및 시스템 마스터 키로 암호화된 데이터들을 상기 사용자 컴퓨터(110)에 의해 상기 저장 서버(120)로 전송하여 분산 저장한다(단계 S405).

[0059] 그리고, 상기 단계 S404에서 생성된 메타 정보를 상기 사용자 컴퓨터(110)에 의해 상기 검증 서버(130)로 전송한다(단계 S406).

[0060] 한편, 상기 검증 서버(130)는 상기 저장 서버(120)에 저장되어 있는 데이터의 무결성을 검사하기 위해 저장 서버(120)로 검사 요청을 전달한다(단계 S407).

[0061] 그러면, 상기 저장 서버(120)는 자신이 저장하고 있는 암호화된 데이터를 해시 함수로 해시한 후 그 결과인 해시값을 상기 검증 서버(130)로 전달한다(단계 S408).

[0062] 이에 따라, 상기 검증 서버(130)는 상기 저장 서버(120)로부터 전송받은 해시값과 자신이 저장하고 있는 메타 정보를 비교하여 데이터의 무결성 여부를 검사한다(단계 S409,S410). 여기서, 상기 해시값과 메타 정보와의 비교 값이 참이면, 검증 서버(130)는 데이터가 위조나 변조 없이 잘 보관되고 있는 것으로 판단하고(단계 S411), 비교 값이 거짓이면 데이터가 손상된 것으로 판단한다(단계 S412).

[0063] 이렇게 하여 저장 서버(120)에 저장되어 있는 데이터에 대한 무결성 검사가 완료되면, 상기 검증 서버(130)는 그 검사 결과를 상기 사용자 컴퓨터(110)(데이터 소유자)로 통보한다(단계 S413).

[0064] 여기서, 상기 단계 S401에서의 데이터 분할, 상기 단계 S402 및 S403에서의 데이터 암호화, 그리고 상기 단계 S404에서의 메타 정보 생성 및 상기 단계 S409~S412에서의 데이터 무결성 검증과 관련하여 각각 부연 설명해 보기로 한다.

[0065] < 데이터 분할 >

[0066] 데이터 소유자(사용자 컴퓨터(110))는 원본 데이터를 일정한 크기로 분할한다. 이때, 일정한 크기란 데이터 소유자가 입력한 크기를 말한다. 이때 또한 분할된 데이터의 명칭은 원본 데이터의 명칭에 확장자를 덧붙여서 사용하는 것이 바람직하다. 이는 차후에 분할된 데이터를 병합할 때 별도의 데이터 명칭을 설정하는 번거로움을 해소할 수 있기 때문이다. 즉, data.txt라는 원본 데이터를 3개로 분할할 경우, "data-txt.d1", "data-txt.d2", "data-txt.d3"라는 분할 데이터로 저장하도록 하며, "data-txt.d1", "data-txt.d2", "data-txt.d3"이 병합될 때에는 자동적으로 파일 명칭이 data-txt인 것들만 병합하여 data.txt 명칭으로 저장하도록 한다.

[0067] 도 5는 본 발명에 따른 자가조직 저장매체의 보안 방법에 있어서, 데이터 분할의 과정을 설명하는 흐름도이다.

[0068] 전술한 바와 같이 데이터 분할은 데이터 소유자가 데이터를 분산 저장할 목적으로 적당한 크기로 나누는 것을 말한다. 예를 들어, data.txt가 1000kb의 크기를 갖는 원본 데이터일 때, 300kb으로 데이터를 분할한다면, data.txt는 300kb, 300kb, 300kb, 100kb로 나누어질 것이다. 즉, data-txt.d1, data-txt.d2, data-txt.d3, data-txt.d4로 나누어지는데, 각각의 파일의 크기는 300kb, 300kb, 300kb, 100kb가 된다. 이렇게 나누는 과정을 보여주는 것이 도 5이다.

[0069] 도 5를 참조하면, 먼저 파일을 읽고, 파일의 분할 크기를 입력 받는다. 위의 예에서와 같이 data.txt가 1000kb일 때, 300kb로 나눈다면, block 수는 4가 된다. 즉, data.txt가 4개의 파일로 분할된다. 이후 블록으로 나누어진 각 파일을 저장할 파일 명칭을 생성한다. 예를 들면, 각 파일 명칭을 "data-txt.d1", "data-txt.d2", "data-txt.d3", "data-txt.d4"와 같이 생성한다.

[0070] 파일 명칭이 생성된 후, 블록으로 나누어진 파일들을 저장한다. 이때 파일들을 순차적으로 저장하며, 파일의 일련번호 수(i)가 나누어진 블록수(여기서는 4)보다 작으면(i<4), 파일을 블록수만큼 저장하도록 저장 과정을 반복 수행한다. 그리고, 파일의 일련번호 수(i)가 나누어진 블록수(4)와 같으면(i=4), 블록수만큼 파일을 저장하였으므로 파일을 닫고 작업을 종료한다.

[0071] < 데이터 암호화 >

[0072] 데이터 소유자(사용자 컴퓨터(110))는 데이터를 데이터 소유자의 비밀키로 암호화하고, 시스템 마스터 키인 SM_key로 한 번 더 암호화한 후 그 암호화된 데이터를 저장 서버(120)로 전송한다. 이때, 저장 서버(120)는 전송받은 암호화된 데이터를 시스템 마스터 키인 SM_key로 복호화한 후, 수신된 암호문의 무결성을 검증한다. 무결성 검증이 유효하다면, 저장 서버(120)는 데이터 소유자의 비밀키로 암호화된 데이터를 그대로 저장하면서, SFAT에 데이터 정보와 데이터 소유자 ID, 검증 서버 ID를 저장한다. 만약 무결성 검증이 유효하지 않다면 데이터 소유자(사용자 컴퓨터(110))와 검증 서버(120)로 데이터 전송에 에러가 발생하였음을 통지하고, 데이터 수신을 재요청한다.

[0073] 도 6은 데이터 암호화와 무결성 검증 과정을 설명하는 도면이다.

[0074] 도 6을 참조하면, 데이터 소유자(사용자 컴퓨터(110))는 data1을 데이터 소유자의 비밀키로 암호화한다(E_data1=Encode(data1, OS_key)). 그런 후, 시스템 마스터 키인 SM_key로 다시 한 번 암호화한다(S_data1=Encode (E_data1, SM_key)). 그런 다음, E_data1을 해시함수로 해시한다(H_data1=Hash(E_data1)). 그리고, S_data1과 H_data1을 저장 서버(120)에 전송한다.

[0075] 저장 서버(120)는 시스템 마스터 키인 SM_key로 S_data1을 복호화한다(E_data1= Decode(S_data1, SM_key). 그리고, 복호화된 E_data1을 해시함수로 해시한다(SH_data1=Hash(E_data1)). 그런 후, 전송받은 H_data1과 해시

함수로 해시한 SH_data1과 비교하여 무결성을 확인한다. 그리고, 전송받은 데이터의 무결성이 확인되면, 데이터 소유자의 비밀키로 암호화된 E_data1을 저장 서버(120)에 저장한다.

[0076] <메타 정보 생성>

[0077] 전술한 바와 같이, MGM(Meta Generation Module)은 데이터 소유자(사용자 컴퓨터(110))가 저장 서버(120)로 암호화하여 전송한 각각의 분할 데이터에 대한 메타 정보를 생성하여 검증 서버(130)로 전송한다.

[0078] 도 7은 MGM(Meta Generation Module)의 데이터 처리과정을 설명하는 도면이다.

[0079] 도 7을 참조하면, 데이터 소유자(사용자 컴퓨터(110))는 Data1을 데이터 소유자의 비밀키로 암호화한다(E_data1=Encode(Data1, OS_key). 그런 후, 그 암호화된 데이터를 해시함수로 해시한다(H_Data1=Hash(E_data1)). 그리고, 저장 서버 ID를 시스템 마스터 키인 SM_key로 암호화한다(E_ID=Encode(SID, SM_key). 그런 다음, 상기 암호화된 데이터를 해시 함수로 해시한 결과와 저장 서버 ID를 시스템 마스터 키인 SM_key로 암호화한 결과를 검증 서버(130)에 전달한다.

[0080] 이에 따라, 검증 서버(130)는 시스템 마스터 키인 SM_key로 복호화한 후, 검증 서버 파일 할당 테이블인 VFAT에 H_Data1, 데이터 소유자 ID, 저장 서버 ID를 저장한다.

[0081] < 데이터 무결성 검증 >

[0082] DVM(Data Verification Module)에 의한 데이터 무결성 검증 절차 과정을 정리하면 다음과 같은 4단계로 정리할 수 있다. 각 단계별 처리 과정을 살펴보면 다음과 같다.

[0083] 제1 단계: 검증 서버(130)는 VFAT에서 검증 서버가 검증을 위임받은 저장 서버(120)를 찾아서 검증 요청 신호를 전송한다. 이때 플로딩 공격 및 재응답 공격을 방지할 목적으로 nonce를 랜덤하게 생성하여 전송하고, 검증 서버(130)의 ID를 시스템 마스터 키인 SM_key로 암호화하여 저장 서버(120)에 전송한다.

[0084] 제2 단계: 저장 서버(120)는 시스템 마스터 키인 SM_key로 암호화된 검증 서버(130)의 ID를 복호화하고 검증 서버(130)의 ID가 SFAT에 존재하는지를 검사한다. 검증 서버(130)의 ID가 SFAT에 존재하면, E_Data1을 확인한다. 그리고 SFAT 테이블의 데이터 정보로 저장하고 있던 데이터를 해시한 값과 nonce값을 검증 서버(130)로 전송한다.

[0085] 제3 단계: 검증 서버(130)는 저장 서버(120)로부터 수신한 암호화된 저장 서버 ID를 시스템 마스터 키인 SM_key로 복호화한다. 그리고 저장 서버(120)에서 수신한 nonce와 저장 서버 ID를 연접해 해시한 값과, 검증 서버(130)가 전송한 nonce값과 복호화된 저장 서버 ID와 연접해 해시한 값이 일치하는지를 먼저 검사한다. 만약에 일치한다면 검증 서버(130)가 nonce를 전송한 저장 서버(120)로부터 응답 메시지를 받은 것으로 간주하고, 저장 서버(120)가 보낸 암호화된 데이터의 해시값과 검증 서버(130)가 갖고 있는 데이터의 메타(meta) 정보를 비교함으로써 저장 서버(120)에 저장된 데이터의 변조 및 삭제 여부를 검증한다.

[0086] 제4 단계: 검증 서버(130)는 위의 제1, 2, 3 단계의 검증 과정을 주기적으로 실시하여 그 결과를 데이터 소유자(사용자 컴퓨터(110))에게 통보한다.

[0087] 여기서, 이상에 대해 조금 더 상세히 설명해 보기로 한다.

[0088] 도 8은 DVM(Data Verification Module)에 의한 데이터 무결성 검증 절차 과정을 설명하는 도면이다. 도 8을 참조하면서 각 단계별 처리 과정을 살펴보기로 한다.

[0089] - step 1: 검증 서버(130)는 검증 서버 파일 할당 테이블인 VFAT에서 검증을 위임받은 파일의 저장 서버(120)에 검증 요청 메시지를 전송한다.

[0090] - step 2: 이때, 검증 서버(130)는 랜덤 함수를 이용해 랜덤한 수인 nonce를 생성하고, 검증 서버의 ID를 시스템 마스터 키인 SM_key로 암호화하여 저장 서버 (120)로 전송한다.

[0091] nonce, E_IDv=Encode(IDv, SM_key)

[0092] - step 3: 저장 서버(120)는 시스템 마스터 키인 SM_key로 step 2에서 전송받은 E_IDv를 복호화한다.

[0093] VID=Decode(E_IDv, SM_key)

[0094] - step 4: 저장 서버(120)는 VID를 저장 서버 파일 할당 테이블(VFAT)에서 검색한다.

[0095] - step 5: 저장 서버(120)는 저장 서버 ID를 시스템 마스터 키인 SM_key로 암호화 한다.

[0096] E_IDs = Encode(SID, SM_key)

[0097] - step 6: 저장 서버는 step 2에서 전송받은 nonce와 저장 서버 ID를 연접한 후 해시함수로 해시한다.

[0098] E_nonce = Hash(nonce || SID)

[0099] - step 7: 저장 서버(120)는 저장서버 파일 할당 테이블(SFAT)에서 해당 파일을 찾아서 해시 함수로 해시한다.

[0100] H_Datas = Hash(E_Data1)

[0101] - step 8: 저장 서버(120)는 step 5, 6, 7의 결과를 검증 서버(130)로 전송한다.

[0102] - step 9: 검증 서버(130)는 시스템 마스터 키인 SM_key로 E_IDs를 복호화한다.

[0103] SID = Decode(E_IDS, SM_key)

[0104] - step 10: 검증 서버(130)는 step 9의 결과와 step 2에서 생성한 nonce를 연접하여 해시함수로 해시한다.

[0105] VE_nonce=Hash(nonce || SID)

[0106] - step 11: 검증 서버(130)는 step 10의 결과와 step 8에서 전송받은 E_nonce를 비교한다. 비교 결과가 참이면, 검증을 요청한 저장 서버(120)가 응답하는 것으로 간주하고, 후속되는 step 12를 실행한다.

[0107] 그리고, 비교 결과가 거짓이면 검증을 요청한 저장 서버(120)가 응답한 것이 아니라, 제3자가 응답한 것이므로 검증 절차를 강제로 종료한다.

[0108] - step 12: 검증 서버(130)는 step 8에서 전송받은 H_Datas와 검증 서버 파일 할당 테이블(VFAT)에 저장되어 있던 H_Data1을 비교한다. 비교 결과가 참이면 저장 서버(120)에 저장되어 있는 데이터가 변조 없이 저장되어 있는 것으로 판단하고, 비교 결과가 거짓이면 저장 서버(120)에 저장되어 있는 데이터에 문제가 발생한 것으로 판단한다. 그리고, 이러한 결과를 데이터 소유자(사용자 컴퓨터)(110)에게 전송한다.

[0109] 한편, 데이터 소유자(사용자 컴퓨터(110))는 저장 서버(120)에 분산되어 있던 암호화된 데이터를 수집하여 복호화한 후, 완성된 데이터로 병합한다. 이때 데이터 소유자(사용자 컴퓨터(110))는 분할된 데이터의 무결성에 대한 정보를 검증 서버(130)로부터 주기적으로 수신한다. 그러나, 바람직하게는 데이터 소유자(사용자 컴퓨터(110))가 병합된 데이터의 무결성을 한 번 더 검증하도록 시스템적으로 구성한다. 이는 데이터의 신뢰성을 한층 더 향상시키기 위한 것이다.

[0110] 이상의 설명에서와 같이 본 발명에 따른 자가조직 저장매체의 보안 시스템 및 그 방법은 인프라 클라우딩 환경의 저장매체에 데이터를 저장함에 있어서, 3자간 협상 프로토콜을 통해 데이터를 분할하여 분산 저장하고, 저장된 데이터를 권한을 위임받은 제3자가 검증하도록 함으로써 데이터에 대한 안전성, 무결성 및 신뢰도를 한층 향상시킬 수 있는 효과가 있다.

[0111] 이상, 바람직한 실시예를 통하여 본 발명에 관하여 상세히 설명하였으나, 본 발명은 이에 한정되는 것은 아니며, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 다양하게 변경, 응용될 수 있음은 당해 기술분야의 통상의 기술자에게 자명하다. 따라서, 본 발명의 진정한 보호 범위는 다음의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술적 사상은 본 발명의 권리 범위에 포함되는 것으로 해석되어야 할 것이다.

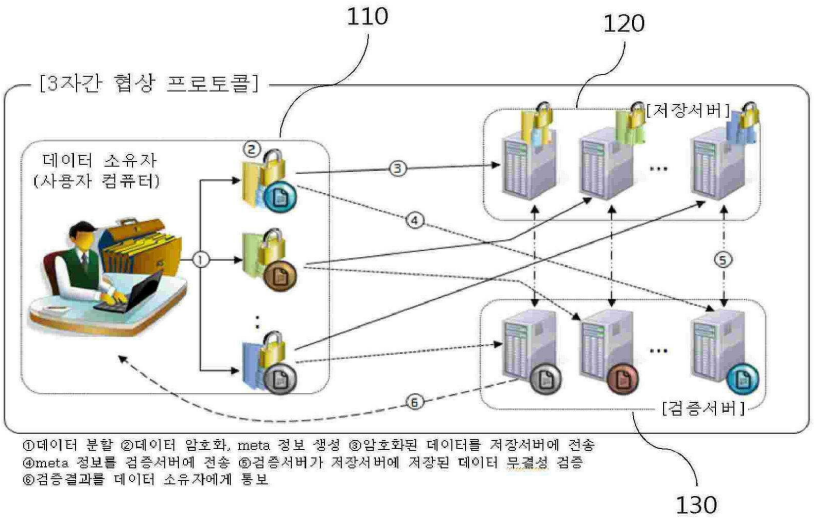
부호의 설명

[0112] 110...데이터 소유자(사용자 컴퓨터) 120...저장 서버

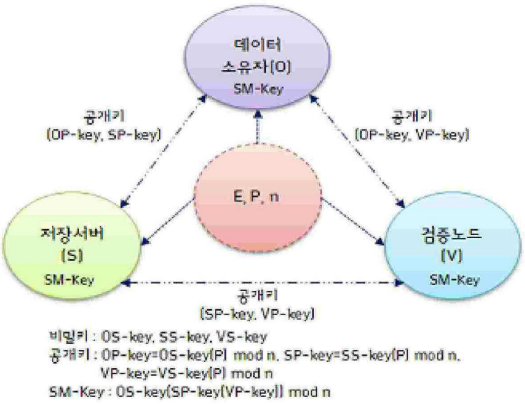
130...검증 서버

도면

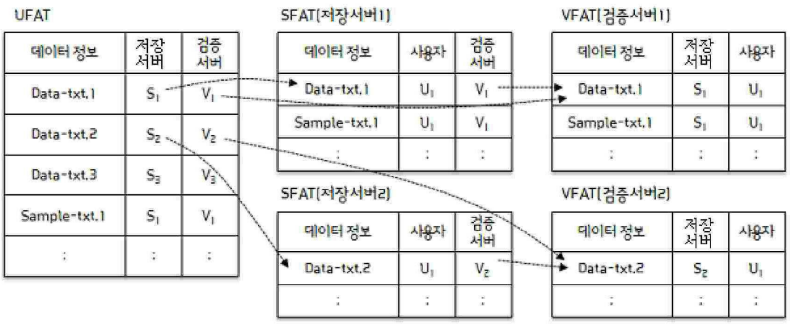
도면1



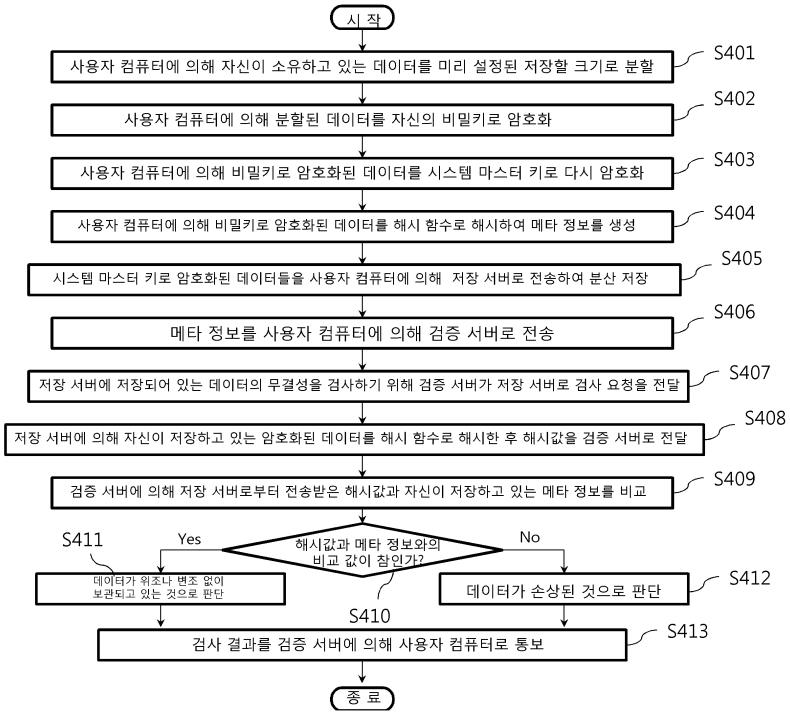
도면2



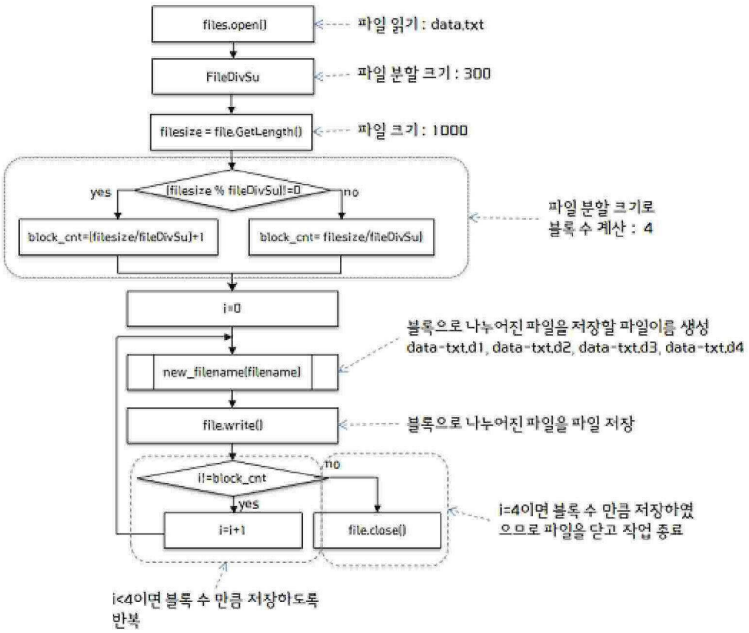
도면3



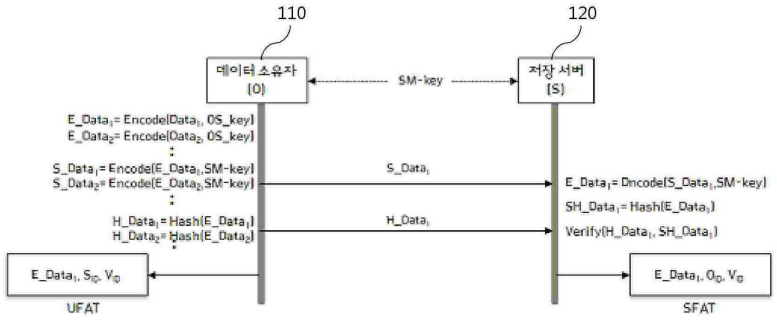
도면4


시 작
사용자 컴퓨터에 의해 자신이 소유하고 있는 데이터를 미리 설정된 저장할 크기로 분할
S401
사용자 컴퓨터에 의해 분할된 데이터를 자신의 비밀키로 암호화
S402
사용자 컴퓨터에 의해 비밀키로 암호화된 데이터를 시스템 마스터 키로 다시 암호화
S403
사용자 컴퓨터에 의해 비밀키로 암호화된 데이터를 해시 함수로 해시하여 메타 정보를 생성
S404
시스템 마스터 키로 암호화된 데이터들을 사용자 컴퓨터에 의해 저장 서버로 전송하여 분산 저장
S405
메타 정보를 사용자 컴퓨터에 의해 검증 서버로 전송
S406
저장 서버에 저장되어 있는 데이터의 무결성을 검사하기 위해 검증 서버가 저장 서버로 검사 요청을 전달
S407
저장 서버에 의해 자신이 저장하고 있는 암호화된 데이터를 해시 함수로 해시한 후 해시값을 검증 서버로 전달
S408
검증 서버에 의해 저장 서버로부터 전송받은 해시값과 자신이 저장하고 있는 메타 정보를 비교
S409
S411
Yes
해시값과 메타 정보와의 비교 값이 참인가?
No
데이터가 위조나 변조 없이 보관되고 있는 것으로 판단
데이터가 손상된 것으로 판단
S412
S410
검사 결과를 검증 서버에 의해 사용자 컴퓨터로 통보
S413
종 료

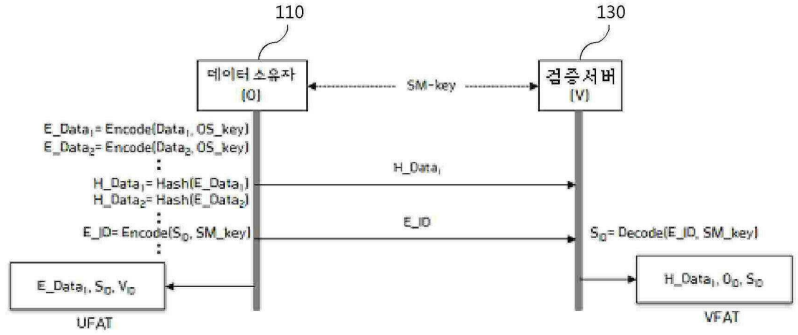
도면5


files.open()
파일 읽기 : data.txt
FileDivSu
파일 분할 크기 : 300
filesize = file.GetLength()
파일 크기 : 1000
yes
if(filesize % fileDivSu!=0)
no
block_cnt=(filesize/fileDivSu)+1
block_cnt= filesize/fileDivSu
파일 분할 크기로
블록 수 계산 : 4
i=0
new_filename(filename)
블록으로 나누어진 파일을 저장할 파일이름 생성
data-txt.d1, data-txt.d2, data-txt.d3, data-txt.d4
file.write()
블록으로 나누어진 파일을 파일 저장
i!=block_cnt
no
yes
i=i+1
file.close()
i=4이면 블록 수 만큼 저장하였으므로 파일을 닫고 작업 종료
i<4이면 블록 수 만큼 저장하도록 반복

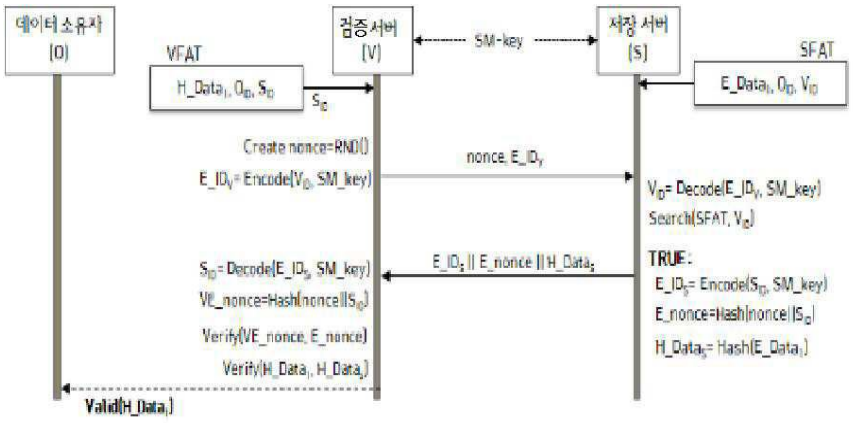
도면6


110
120
데이터 소유자
(O)
SM-key
저장 서버
(S)
E_Data1= Encode(Data1, OS_key)
E_Data2= Encode(Data2, OS_key)
S_Data1= Encode(E_Data1,SM-key)
S_Data2= Encode(E_Data2,SM-key)
S_Data1
E_Data1= Dncode(S_Data1,SM-key)
SH_Data1= Hash(E_Data1)
H_Data1= Hash(E_Data1)
H_Data2= Hash(E_Data2)
H_Data1
Verify(H_Data1, SH_Data1)
E_Data1, SID, VID
E_Data1, OID, VID
UFAT
SFAT

도면7


110
130
데이터 소유자
(O)
SM-key
검증서버
(V)
E_Data1= Encode(Data1, OS_key)
E_Data2= Encode(Data2, OS_key)
H_Data1
H_Data1= Hash(E_Data1)
H_Data2= Hash(E_Data2)
E_ID
E_ID= Encode(SID, SM_key)
SID= Decode(E_ID, SM_key)
E_Data1, SID, VID
H_Data1, OID, SID
UFAT
VFAT

도면8


데이터 소유자
(O)
VFAT
H_Data1, OID, SID
SID
검증서버
(V)
SM-key
저장 서버
(S)
SFAT
E_Data1, OID, VID
Create nonce=RND()
E_IDV= Encode(VID, SM_key)
nonce, E_IDV
VID= Decode(E_IDV, SM_key)
Search(SFAT, VID)
TRUE:
E_IDS= Encode(SID, SM_key)
E_nonce=Hash(nonce||SID)
H_Data2= Hash(E_Data1)
SID= Decode(E_IDS, SM_key)
VE_nonce=Hash(nonce||SID)
E_IDS || E_nonce || H_Data2
Verify(VE_nonce, E_nonce)
Verify(H_Data1, H_Data2)
Valid(H_Data1)