



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0109272
(43) 공개일자 2017년09월29일

(51) 국제특허분류(Int. Cl.)
G06F 21/62 (2013.01) G06F 17/30 (2006.01)
G06F 21/60 (2013.01)
(52) CPC특허분류
G06F 21/6227 (2013.01)
G06F 17/30171 (2013.01)
(21) 출원번호 10-2016-0033152
(22) 출원일자 2016년03월21일
심사청구일자 2016년03월21일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
윤지원
서울특별시 용산구 한강대로43길 8, 102동 1401호
(한강로2가, 벽산메가트리움)
정지수
대구광역시 수성구 달구벌대로522길 33, 104동 803호 (만촌동, 수성2차 e 편한세상아파트)
(74) 대리인
특허법인엠에이피에스

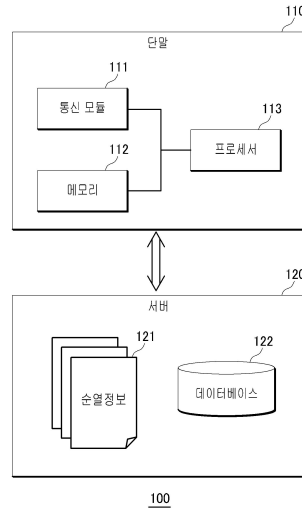
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 데이터 검색 단말 및 그의 검색 방법

(57) 요약

본 발명은 통신 모듈, 데이터 검색 프로그램이 저장된 메모리 및 프로그램을 실행하는 프로세서를 포함한다. 이 때, 프로세서는 프로그램의 실행에 따라, 범위 및 키에 기초하여, 서버에 저장된 데이터 중 검색 대상 데이터에 대응하는 인덱스를 산출하며, 산출된 인덱스를 서버로 전송하고, 서버로부터 산출된 인덱스에 대응하는 데이터를 수신한다. 그리고 범위는 서버로부터 수신한 순열 정보에 기초하여 산출된 것이며, 키는 서버에 저장된 데이터에 대응하여 설정된다.

대표도 - 도1



(52) CPC특허분류

G06F 17/30386 (2013.01)

G06F 21/602 (2013.01)

G06F 2211/007 (2013.01)

명세서

청구범위

청구항 1

서버에 저장된 데이터를 검색하는 단말에 있어서,

통신 모듈,

데이터 검색 프로그램이 저장된 메모리 및

상기 프로그램을 실행하는 프로세서를 포함하고,

상기 프로세서는 상기 프로그램의 실행에 따라, 범위 및 키에 기초하여, 상기 서버에 저장된 데이터 중 검색 대상 데이터에 대응하는 인덱스를 산출하며, 상기 산출된 인덱스를 상기 서버로 전송하고, 상기 서버로부터 상기 산출된 인덱스에 대응하는 데이터를 수신하되,

상기 범위는 상기 서버로부터 수신한 순열 정보에 기초하여 산출된 것이며,

상기 키는 상기 서버에 저장된 데이터에 대응하여 설정된 것인, 데이터 검색 단말.

청구항 2

제 1 항에 있어서,

상기 프로세서는 상기 서버로부터 수신한 데이터 및 상기 검색 대상 데이터를 비교하고, 상기 서버로부터 수신한 데이터 및 상기 검색 대상 데이터가 매칭되지 않을 경우, 상기 범위를 재산출하고,

상기 서버로부터 수신한 데이터 및 상기 검색 대상 데이터가 매칭될 때까지 상기 재산출된 범위 및 상기 키에 기초하여, 인덱스를 재산출하며, 상기 재산출된 인덱스를 상기 서버로 전송하고, 상기 서버로부터 상기 재산출된 인덱스에 대응하는 데이터를 재수신하는 것을 반복하는, 데이터 검색 단말.

청구항 3

제 1 항에 있어서,

상기 프로세서는 이진 검색 방법에 기초하여, 상기 검색 대상 데이터를 검색하는 것인, 데이터 검색 단말.

청구항 4

제 1 항에 있어서,

상기 프로세서는 상기 서버로부터 순열 정보를 수신하고, 상기 순열 정보에 기초하여, 상기 키를 선택하며, 상기 선택된 키 및 상기 순열 정보에 기초하여, 원시 데이터를 재정렬하고, 상기 재정렬된 원시 데이터를 상기 서버의 데이터베이스에 저장되도록 상기 서버로 전송하되,

상기 순열 정보는 유계 행렬(bounded matrix) 및 하나 이상의 균형 행렬(balanced matrix)을 포함하는, 데이터 검색 단말.

청구항 5

제 4 항에 있어서,

상기 프로세서는 상기 유계 행렬의 차원 및 상기 하나 이상의 균형 행렬의 차원의 곱을 산출하고, 상기 산출된 곱이 상기 데이터의 수보다 크도록 상기 유계 행렬 및 상기 하나 이상의 균형 행렬을 선택하며,

상기 산출된 곱보다 작거나 같은 값을 상기 키로 선택하는 것인, 데이터 검색 단말.

청구항 6

제 5 항에 있어서,

상기 프로세서는 상기 선택된 키 및 상기 수신된 순열 정보에 기초하여, 상기 서버에 저장할 데이터를 순차적으로 선택하여, 상기 서버로 전송하는 것인, 데이터 검색 단말.

청구항 7

제 4 항에 있어서,

상기 유계 행렬은 동일한 열에 동일한 값의 원소가 미리 정해진 값 미만으로 포함되는 행렬이며,

상기 균형 행렬은 동일한 열에 동일한 원소가 포함되지 않는 행렬인, 데이터 검색 단말.

청구항 8

제 4 항에 있어서,

상기 원시 데이터는 상기 원시 데이터에 포함된 특정 속성에 대하여 정렬된 것인, 데이터 검색 단말.

청구항 9

제 4 항에 있어서,

상기 프로세서는 유계 행렬을 생성하고, 상기 생성된 유계 행렬 및 상기 서버로부터 수신한 하나 이상의 균형 행렬에 기초하여, 상기 원시 데이터를 재정렬하고, 상기 유계 행렬을 상기 메모리에 저장하는, 데이터 검색 단말.

청구항 10

제 4 항에 있어서,

상기 프로세서는 데이터가 추가되면, 상기 선택된 키 및 상기 순열 정보에 기초하여, 추가 데이터를 재정렬하고, 상기 재정렬된 추가 데이터가 상기 서버에 저장되도록 상기 재정렬된 추가 데이터를 상기 서버로 전송하되,

상기 순열 정보는 상기 원시 데이터 및 상기 추가 데이터의 크기에 기초하여 선택된 것인, 데이터 검색 단말.

청구항 11

단말에서의 서버에 저장된 데이터 검색 방법에 있어서,

범위 및 키에 기초하여, 상기 서버에 저장된 데이터 중 검색 대상 데이터에 대응하는 인덱스를 산출하는 단계;

상기 산출된 인덱스를 상기 서버로 전송하는 단계; 및

상기 서버로부터 상기 산출된 인덱스에 대응하는 데이터를 수신하는 단계를 포함하되,

상기 범위는 상기 서버로부터 수신한 순열 정보에 기초하여 산출된 것이며,

상기 키는 상기 서버에 저장된 데이터에 대응하여 설정된 것인, 데이터 검색 방법.

청구항 12

제 11 항에 있어서,

상기 데이터를 수신하는 단계 이후에, 상기 수신된 데이터와 상기 검색 대상 데이터를 비교하는 단계; 및

상기 서버로부터 수신한 데이터가 상기 검색 대상 데이터와 매칭되지 않을 경우, 상기 범위를 재산출하는 단계를 포함하되,

상기 서버로부터 수신한 데이터가 상기 검색 대상 데이터와 매칭될 때까지 상기 재산출된 범위 및 상기 키에 기초하여, 상기 인덱스를 산출하는 단계, 상기 산출된 인덱스를 상기 서버로 전송하는 단계, 상기 데이터를 수신하는 단계 및 상기 비교하는 단계를 반복 수행하는, 데이터 검색 방법.

청구항 13

제 11 항에 있어서,

상기 인덱스를 산출하는 단계 이전에,

상기 서버로부터 순열 정보를 수신하는 단계;

상기 순열 정보에 기초하여, 상기 키를 선택하는 단계;

상기 선택된 키 및 상기 순열 정보에 기초하여, 원시 데이터를 재정렬하는 단계; 및

상기 재정렬된 원시 데이터를 상기 서버의 데이터베이스에 저장되도록 상기 서버로 전송하는 단계를 더 포함하
되,

상기 순열 정보는 유계 행렬(bounded matrix) 및 하나 이상의 균형 행렬(balanced matrix)을 포함하는, 데이터
검색 방법.

청구항 14

제 11 항 내지 제 13 항 중 어느 한 항에 기재된 방법을 컴퓨터 상에서 수행하기 위한 프로그램을 기록한 컴퓨
터 판독 가능한 기록 매체.

발명의 설명

기술 분야

[0001] 본 발명은 데이터 검색 단말 및 그의 검색 방법에 관한 것이다.

배경 기술

[0002] 일반적인 데이터 검색 기술은 암호화되지 않은 정보에 대한 검색을 수행한다. 그러나 클라우드(cloud) 환경에
서 데이터를 저장하는 기술이 발달함에 따라, 암호화된 데이터에 대한 검색에 대한 요구가 증가하고 있다.

[0003] 순서보존 암호화(order preserving encryption; OPE) 방법은 데이터를 암호화하되, 결과 값의 순서를 보존하는
방법이다. 순서보존 암호화 방법은 데이터 암호화를 통하여, 안정성 및 검색에 대한 효율성을 동시에 만족시킬
수 있다. 그러나 순서 보존 암호화 방법은 순서가 보존되므로 순서로부터 데이터 정보가 유출될 수 있는 위험
이 있다.

[0004] 이와 관련되어, 한국 공개특허공보 제1584220호(발명의 명칭: "암호화 데이터 정렬 유지를 위한 인코딩 방법")
는 카운터 모드 블록 암호화에 의한 암호화된 데이터를 평문 데이터 정렬 순서를 유지할 수 있게 인코딩하는 방
법을 개시하고 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 서버에 저장된 암호화된 데이터에 대한 검색
을 수행할 수 있는 데이터 검색 단말 및 그의 검색 방법에 관한 것이다.

[0006] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기
술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0007] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 제 1 측면에 따른 서버에 저장된 데이터를
검색하는 단말은 통신 모듈, 데이터 검색 프로그램이 저장된 메모리 및 프로그램을 실행하는 프로세서를 포함하
다. 이때, 프로세서는 프로그램의 실행에 따라, 범위 및 키에 기초하여, 서버에 저장된 데이터 중 검색 대상
데이터에 대응하는 인덱스를 산출하며, 산출된 인덱스를 서버로 전송하고, 서버로부터 산출된 인덱스에 대응하
는 데이터를 수신한다. 그리고 범위는 서버로부터 수신한 순열 정보에 기초하여 산출된 것이며, 키는 서버에
저장된 데이터에 대응하여 설정된다.

[0008] 또한, 본 발명의 제 2 측면에 따른 단말에서의 서버에 저장된 데이터 검색 방법은 범위 및 키에 기초하여, 서버
에 저장된 데이터 중 검색 대상 데이터에 대응하는 인덱스를 산출하는 단계; 산출된 인덱스를 서버로 전송하는

단계; 및 서버로부터 산출된 인덱스에 대응하는 데이터를 수신하는 단계를 포함한다. 이때, 범위는 서버로부터 수신한 순열 정보에 기초하여 산출된 것이며, 키는 서버에 저장된 데이터에 대응하여 설정된 것이다.

발명의 효과

[0009] 본 발명은 단말에서 데이터를 재정렬하고, 서버로 전송하여 서버의 데이터베이스에 저장할 수 있다. 그러므로 본 발명은 서버에 저장된 데이터의 안정성을 보장할 수 있다. 또한, 본 발명은 이진 검색을 통하여, 서버에 저장된 암호화된 데이터에 대한 효율적인 검색을 수행할 수 있다. 그러므로 본 발명은 데이터 유출을 방지하고, 암호화된 데이터의 신속한 검색을 제공할 수 있다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 일 실시예에 따른 데이터 검색 시스템의 블록도이다.
 도 2는 본 발명의 일 실시예에 따른 유계 행렬 및 균형 행렬의 예시도이다.
 도 3은 본 발명의 일 실시예에 따른 데이터 암호화 행렬의 예시도이다.
 도 4는 본 발명의 일 실시예에 따른 데이터의 예시도이다.
 도 5는 본 발명의 일 실시예에 따른 데이터 검색의 예시도이다.
 도 6은 본 발명의 일 실시예에 따른 데이터 검색 단말의 검색 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0011] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0012] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0013] 다음은 도 1 내지 도 5를 참조하여, 본 발명의 일 실시예에 따른 데이터 검색 시스템(100)을 설명한다.

[0014] 도 1은 본 발명의 일 실시예에 따른 데이터 검색 시스템(100)의 블록도이다.

[0015] 데이터 검색 시스템(100)은 단말(110)이 서버(120)에 저장된 암호화된 데이터를 검색한다. 이때, 서버(120)에 저장된 데이터는 단말(110)에 의해 암호화된 후 서버(120)에 포함된 데이터베이스(122)에 저장된 것일 수 있다.

[0016] 데이터 검색 시스템(100)은 서버(120) 및 복수의 단말(110)을 포함할 수 있다.

[0017] 서버(120)는 단말(110)로부터 데이터를 수신하여 저장할 수 있다. 또한, 서버(120)는 단말(110)의 요청에 따라, 단말(110)에 단말(110)의 요청에 대응하는 데이터를 제공할 수 있다. 서버(120)는 데이터를 저장할 수 있는 데이터베이스(122)를 포함할 수 있다.

[0018] 이때, 데이터베이스(122)는 서버(120)에 응용 프로그램(application) 또는 서비스(service)의 형태로 포함된 것일 수 있다. 또한, 데이터베이스(122)는 서버(120)와 연결된 데이터베이스 서버일 수 있다. 그리고 서버(120)는 클라우드 서버일 수 있다.

[0019] 서버(120)는 단말(110)에서 암호화를 위하여 사용할 수 있는 순열 정보를 포함할 수 있다. 이때, 순열 정보는 행렬의 모든 행이 순열(permutation)로 이루어진 유계 행렬(bounded matrix) 및 균형 행렬(balanced matrix)을 포함할 수 있다. 유계 행렬 및 균형 행렬은 행 및 열의 수가 동일한 정방 행렬(square matrix)일 수 있다.

[0020] 도 2는 본 발명의 일 실시예에 따른 유계 행렬 및 균형 행렬의 예시도이다.

[0021] 균형 행렬은 도 2의 (a)와 같이, 동일한 열에 동일한 원소가 한 번도 중복될 수 없는 행렬을 의미한다. 균형 행렬은 "1-균형 행렬" 또는 "1-balanced"로 표기할 수 있다.

- [0022] 유계 행렬은 도 2의 (b)와 같이, 동일한 열에 동일한 원소가 특정 개수 이하 포함된 행렬을 의미한다. 유계 행렬은 열에 포함된 동일한 원소 개수(k)에 따라, " k -유계 행렬" 또는 " k -Bounded"로 표기할 수 있다.
- [0023] 단말(110)은 데이터를 암호화하여, 서버(120)에 저장한다. 그리고 단말(110)은 서버(120)에 저장된 암호화된 데이터를 검색할 수 있다. 단말(110)은 통신 모듈(111), 메모리(112) 및 프로세서(113)을 포함한다.
- [0024] 통신 모듈(111)은 서버(120)와의 데이터 통신을 수행할 수 있다.
- [0025] 메모리(112)는 데이터 검색 프로그램이 저장된다. 이때, 메모리(112)는 전원이 공급되지 않아도 저장된 정보를 계속 유지하는 비휘발성 저장장치 및 저장된 정보를 유지하기 위하여 전력이 필요한 휘발성 저장장치를 통칭하는 것이다.
- [0026] 프로세서(113)는 데이터를 서버(120)에 포함된 데이터베이스(122)에 저장할 수 있다. 이때, 프로세서(113)는 저장되는 데이터를 재정렬하여, 서버(120)로 전송할 수 있다.
- [0027] 구체적으로 프로세서(113)는 서버(120)의 데이터베이스(122)에 원시 데이터(original data)를 저장한다. 이때, 원시 데이터는 특정 속성에 따라 정렬된 데이터일 수 있다. 예를 들어, 원시 데이터는 주 키(primary key)에 따라 정렬된 데이터일 수 있다.
- [0028] 프로세서(113)는 원시 데이터를 저장하기 위하여, 서버(120)에 저장된 순열 정보 중 하나의 유계 행렬 및 하나 이상의 균형 행렬을 선택할 수 있다. 이때, 선택된 유계 행렬 및 균형 행렬의 크기는 저장하고자 하는 원시 데이터의 수에 따라, 결정될 수 있다.
- [0029] 또한, 본 명세서에서 원시 데이터, 유계 행렬 및 균형 행렬의 인덱스(index)는 0부터 시작할 수 있다. 즉 원시 데이터의 인덱스는 0보다 같거나 크며, '원시 데이터 크기 - 1'보다 작은 값을 가질 수 있다. 또한, 행렬에서 행의 인덱스 및 열의 인덱스는 같거나 0보다 크며, '행렬의 차원 수 - 1'보다 작은 값을 가질 수 있다.
- [0030] 한편, 프로세서(113)는 유계 행렬 및 균형 행렬 각각의 차원의 곱이 원시 데이터의 수보다 크도록 유계 행렬 및 균형 행렬의 차원의 수를 선택할 수 있다. 예를 들어, 원시 데이터 수가 10개라면, 프로세서(113)는 유계 행렬 및 균형 행렬의 차원 수의 곱이 10 이상이 되도록 유계 행렬 및 균형 행렬을 선택할 수 있다.
- [0031] 그리고 프로세서(113)는 선택된 차원의 수에 대응하는 유계 행렬 및 균형 행렬을 선택할 수 있다. 프로세서(113)가 유계 행렬 및 균형 행렬을 선택하면, 서버(120)는 선택된 유계 행렬 및 균형 행렬 정보를 저장할 수 있다.
- [0032] 프로세서(113)는 유계 행렬 및 균형 행렬의 곱을 통하여, 데이터 암호화 행렬을 산출할 수 있다.
- [0033] 예를 들어, $N-1$ 개의 균형 행렬($\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{N-1}$) 및 1개의 유계 행렬($\mathbf{M}_N$)이 선택되는 경우, 데이터 암호화 행렬($\mathbf{M}$)은 $N-1$ 개의 균형 행렬 및 1개의 유계 행렬의 곱을 통하여 산출될 수 있다. 데이터 암호화 행렬은 먼저 ($\mathbf{M}_N \times \mathbf{M}_{N-1}$)을 산출하고, 산출된 값에 다시 $\mathbf{M}_{N-2}$ 를 곱하는 연산 순서로 산출될 수 있다. 즉, 데이터 암호화 행렬 $\mathbf{M}$ 는 $\{((\mathbf{M}_N \times \mathbf{M}_{N-1}) \times \dots \times \mathbf{M}_2) \times \mathbf{M}_1\}$ 을 통하여 산출될 수 있다.
- [0034] 예를 들어, [수학식 1]에 기초하여, 1 개의 균형 행렬($\mathbf{M}_1$) 및 1개의 유계 행렬($\mathbf{M}_2$)로부터 데이터 암호화 행렬 $\mathbf{M}$ 을 산출할 수 있다. [수학식 1]에서 m_1 은 균형 행렬($\mathbf{M}_1$)의 차원 수이며, m_2 은 유계 행렬($\mathbf{M}_2$)의 차원 수이다. $\mathbf{J}$ 는 $m_1 \times m_1$ 의 원소가 모두 1인 정방 행렬이다. 또한, 데이터 암호화 행렬 $\mathbf{M}$ 은 행 및 열의 크기가 $(m_1 \times m_2)$ 인 정방 행렬이다.

수학식 1

$$\mathbf{M} = \mathbf{M}_1 \times \mathbf{M}_2$$

$$= \begin{bmatrix} \mathbf{M}_1 + m_1 \times \mathbf{J} \times \mathbf{M}_2(0, 0) & \cdots & \mathbf{M}_1 + m_1 \times \mathbf{J} \times \mathbf{M}_2(0, m_2 - 1) \\ \vdots & \ddots & \vdots \\ \mathbf{M}_1 + m_1 \times \mathbf{J} \times \mathbf{M}_2(m_1 - 1, 0) & \cdots & \mathbf{M}_1 + m_1 \times \mathbf{J} \times \mathbf{M}_2(m_1 - 1, m_2 - 1) \end{bmatrix}$$

[0035]

- [0036] 예를 들어, [수학식 1]에서는 1행 1열부터 m_1 행 m_1 열까지의 원소를 산출하기 위한 식이다. 즉, 앞에서 설명한
- 예제에 따라, 1행 1열부터 3행 3열까지의 원소를 산출하는 식은 $M_1 + m_1 \times J \times M_2(0,0)$ 이다. 이때,
- $$M_1 + m_1 \times J \times M_2(0,0) = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} + 3 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times 0$$
- 는 $\begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ 이므로, 1행 1열부터 3행 3열까지의 원소는 $\begin{bmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ 이 될 수 있다.
- [0037] 즉, 데이터 암호화 행렬 M 에서 1행 1열의 값은 0, 1행 3열의 값은 1이되며, 3행 3열의 값은 2가 될 수 있다.
- [0038] 또한, 1행 10열부터 3행 12열까지의 원소를 산출하는 식은 $M_1 + m_1 \times J \times M_2(0,3)$ 이므로, 1행 10열부터 3행 12열까지의 원소는 $\begin{bmatrix} 9 & 11 & 10 \\ 11 & 10 & 9 \\ 10 & 9 & 11 \end{bmatrix}$ 가 될 수 있다.
- [0039] 데이터 암호화 행렬은 단말(110)에서 서버(120)로부터 유계 행렬 및 균형 행렬을 수신한 후, 계산된 것일 수 있다. 또한, 데이터 암호화 행렬은 서버(120)에서 계산된 후 단말(110)로 전달된 것일 수 있다.
- [0040] 예를 들어, 데이터 암호화 행렬을 단말(110)에서 계산되는 경우, 프로세서(113)는 선택한 유계 행렬 및 하나의 균형 행렬을 서버(120)로부터 수신하여, 데이터 암호화 행렬을 산출할 수 있다.
- [0041] 또한, 서버(120)에서 데이터 암호화 행렬이 계산되는 경우, 프로세서(113)는 선택한 유계 행렬 및 하나 이상의 균형 행렬의 정보를 서버(120)로 전송할 수 있다. 그리고 서버(120)에서 데이터 암호화 행렬을 산출하면, 프로세서(113)는 통신 모듈(111)을 통하여, 산출된 데이터 암호화 행렬을 수신할 수 있다. 또는, 프로세서(113)는 통신 모듈(111)을 통하여, 서버(120)로 키를 전송하고, 서버(120)로부터 키에 대응하는 행을 수신할 수 있다.
- [0042] 그리고 프로세서(113)는 산출된 데이터 암호화 행렬에 기초하여, 원시 데이터를 재정렬할 수 있다. 이때, 프로세서(113)는 키에 기초하여, 재정렬에 사용될 행을 선택할 수 있다. 그리고 사용자가 데이터 암호화를 위한 키를 선택하면, 프로세서(113)는 키를 메모리(112)에 저장할 수 있다.
- [0043] 이때, 키는 단말(110)을 통하여, 원시 데이터를 저장하는 사용자가 선택한 것일 수 있다. 키는 1보다 크거나 같은 값이며, 유계 행렬 및 하나 이상의 균형 행렬 각각의 차원의 곱보다 작거나 같은 값일 수 있다.
- [0044] 도 3은 본 발명의 일 실시예에 따른 데이터 암호화 행렬의 예시도이다.
- [0045] 예를 들어, 도 3을 참조하면 프로세서(113)는 하나의 유계 행렬(M_2) 및 하나의 균형 행렬(M_1)의 곱을 통하여, 데이터 암호화 행렬을 산출할 수 있다. 그리고 사용자가 키로 7번째 행의 인덱스인 "6"을 선택하면, 프로세서(113)는 데이터 암호화 행렬의 7번째 행을 선택할 수 있다.
- [0046] 프로세서(113)는 데이터 암호화 행렬에서 선택한 7번째 행을 이용하여, 저장할 데이터를 재정렬 할 수 있다.
- [0047] 도 4는 본 발명의 일 실시예에 따른 데이터의 예시도이다.
- [0048] 도 4를 참조하면, 프로세서(113)는 도 4의 (a)와 같은 원시 데이터를 재정렬하여, 도 4의 (b)와 같이 서버(120)의 데이터베이스(122)에 저장할 수 있다. 이때, 원시 데이터는 주 키인 "No" 속성에 따라 내림차순 정렬될 수 있다.
- [0049] 프로세서(113)는 먼저 키를 이용하여, 원시 데이터 중 서버(120)에 저장될 데이터를 선택할 수 있다. 그리고 프로세서(113)는 선택된 데이터를 서버(120)로 전송하여, 서버(120)의 데이터베이스(122)에 저장할 수 있다.

- [0050] 예를 들어, 프로세서(113)는 " $M[y] = h$ "를 만족하는 y 값에 기초하여, 서버(120)에 저장할 데이터를 선택할 수 있다. 이때, M 은 데이터 암호화 행렬을 의미하며, y 는 서버(120)에 저장할 데이터의 현재 인덱스가 될 수 있다. 또한, h 는 서버(120)에 저장될 데이터를 선택하는 순서가 될 수 있다.
- [0051] 그리고 프로세서(113)는 원시 데이터 중 1번째로 서버(120)에 저장할 데이터를 선택할 수 있다. 이때, 프로세서(113)는 1번째 데이터를 선택하기 위하여, 변수 h 를 0으로 설정할 수 있다. 그리고 프로세서(113)는 데이터 암호화 행렬에서 " $M[6][y] = 0$ "을 만족하는 y 값인 "6"을 찾을 수 있다. 그러므로 프로세서(113)는 데이터의 인덱스가 "6"인 7번째 데이터 "<'Green', 901212, 'M', 'AAA'>"을 선택할 수 있다. 프로세서(113)는 선택한 데이터를 통신 모듈(111)을 통하여 서버(120)로 전송할 수 있다.
- [0052] 서버(120)는 단말(110)로부터 수신한 데이터를 순서대로 데이터베이스(122)에 저장할 수 있다. 그러므로 단말(110)이 전송한 7번째 데이터는 데이터베이스(122)에 1번째 데이터로 저장되게 된다.
- [0053] 이때, 프로세서(113)가 서버(120)로 전송하는 데이터는 암호화된 것일 수 있다. 즉, 프로세서(113)는 서버(120)로 "<'Green', 901212, 'M', 'AAA'>" 데이터를 전송하기 전에 암호화를 수행할 수 있다. 프로세서(113)는 "<'Green', 901212, 'M', 'AAA'>"에 대응하여 암호화된 데이터를 서버(120)로 전송할 수 있다. 그리고 서버(120)는 암호화된 데이터를 서버(120)의 데이터베이스(122)에 저장할 수 있다.
- [0054] 또한, 프로세서(113)는 원시 데이터 중 2번째 데이터를 선택할 수 있다. 이를 위하여, 프로세서(113)는 h 를 1 증가시켜, 1로 설정할 수 있다. 프로세서(113)는 데이터 암호화 행렬에서 " $M[6][y] = 1$ "을 만족하는 y 값인 "8"을 찾을 수 있다. 그러므로 프로세서(113)는 데이터 인덱스가 "8"인 9번째 데이터 "<'Isabella', 910815, 'F', 'CCC'>"를 선택하여, 통신 모듈(111)을 통하여 서버(120)로 전송할 수 있다.
- [0055] 단말(110)로부터 데이터를 수신한 서버(120)는 서버(120)의 데이터베이스(122)에 수신한 데이터를 저장할 수 있다. 이때, 서버(120)는 단말(110)로부터 수신한 순서대로 데이터를 저장할 수 있다. 그러므로 서버(120)의 데이터베이스(122)에는 "<'Green', 901212, 'M', 'AAA'>" 및 "<'Isabella', 910815, 'F', 'CCC'>"순으로 데이터가 저장되게 된다.
- [0056] 이와 같이, 프로세서(113)는 h 값을 1씩 증가시키면서, 원시 데이터 중 서버(120)에 저장될 데이터를 선택할 수 있다. 그리고 프로세서(113)는 통신 모듈(111)을 통하여 선택된 데이터를 서버(120)로 전송할 수 있다.
- [0057] 다시 도 4를 참조하면 도 4의 (b)와 같이, 서버(120)의 데이터베이스(122)는 재정렬된 데이터가 저장될 수 있다. 즉, 서버(120)의 데이터베이스(122)에 저장되는 데이터는 단말(110)이 전송하는 순서에 따라 저장될 수 있다. 그러므로 도 4의 (a) 및 (b)와 같이, 단말(110)의 원시 데이터와 서버(120)의 데이터베이스(122)에 저장되는 데이터는 순서가 상이할 수 있다.
- [0058] 한편, 프로세서(113)는 사용자의 검색 요청에 따라, 서버(120)로부터 사용자의 검색 요청에 대응하는 데이터를 수신할 수 있다. 이때, 사용자의 검색 요청은 데이터베이스(122)에 대한 쿼리(query) 일 수 있다.
- [0059] 앞에서 설명한 바와 같이, 프로세서(113)는 서버(120)에 원시 데이터를 저장할 때, 저장된 원시 데이터에 대한 재정렬 정보를 저장하지 않으므로, 사용자의 검색 요청에 대응하는 데이터가 저장된 위치를 알 수 없다. 또한, 서버(120)가 저장하는 데이터는 암호화된 데이터이므로 서버(120)는 사용자의 검색 요청에 대응하는 데이터가 정확히 무엇인지 확인할 수 없다.
- [0060] 그러므로 프로세서(113)는 원시 데이터를 저장할 때, 선택된 순열 정보 및 키에 기초하여, 검색 범위를 설정할 수 있다. 그리고 프로세서(113)는 설정된 범위를 줄이는 방향으로 서버(120)의 데이터베이스(122)에 저장된 데이터를 반복적으로 확인하여, 사용자의 검색 요청에 대응하는 데이터를 검색할 수 있다.
- [0061] 이때, 원시 데이터는 특정 속성에 대하여 정렬된 데이터이다. 그러므로 이러한 특징에 기초하여, 프로세서(113)는 이진 검색(binary search)을 수행할 수 있다.
- [0062] 이진 검색 기법은 정렬된 데이터에서 특정 값을 위치의 찾는 검색 기법이다. 구체적으로 이진 검색 기법은 전체 데이터를 대상으로 범위 설정하고, 설정된 범위 내에서 중앙값을 선택한다. 그리고 이진 검색 기법은 선택된 중앙값에 대응하는 데이터가 검색 대상 데이터보다 큰지 작은지를 비교하는 방식으로 검색을 수행한다.
- [0063] 만약, 중앙값에 대응하는 데이터가 검색 대상 데이터보다 큰 경우, 이진 검색 기법은 중앙값을 범위의 새로운 최대값으로 설정하고, 다시 중앙값을 선택하여, 비교를 반복한다. 또한, 중앙값에 대응하는 데이터가 검색 대상 데이터보다 작은 경우, 이진 검색 기법은 중앙값을 범위의 새로운 최소값으로 설정하고, 다시 중앙값을 선택

하여 비교를 반복한다. 이러한 방법을 통하여, 이진 검색 기법은 검색 범위를 줄여가며 원하는 정보를 검색할 수 있다.

- [0064] 본 발명의 일 실시예에 따른 프로세서(113)는 이진 검색 방법에 따라, 사용자 요청에 대응하는 검색 대상 데이터를 검색할 수 있다. 이를 위하여, 프로세서(113)는 먼저, 순열정보, 범위 및 키에 기초하여, 검색할 서버(120)에 저장된 데이터의 인덱스를 산출한다.
- [0065] 이때, 순열정보는 앞에서 설명한 바와 같이, 원시 데이터의 저장 시 단말(110)이 선택하여 서버(120)에 저장된 것일 수 있다. 또한, 범위는 서버(120)에 단말(110)이 원시 데이터 저장 시 선택한 순열 정보에 기초하여 산출된 것일 수 있다. 키는 서버(120)의 데이터베이스(122)에 원시 데이터를 저장할 때 사용한 후 단말(110)의 메모리(112)에 저장된 것일 수 있다.
- [0066] 프로세서(113)는 통신 모듈(111)을 통하여, 산출된 데이터의 인덱스를 서버(120)로 전송한다. 그리고 프로세서(113)는 서버(120)로부터 데이터의 인덱스에 대응하는 데이터를 수신한다. 프로세서(113)는 서버(120)로부터 수신된 데이터와 검색 대상 데이터를 비교할 수 있다. 수신된 데이터와 검색 대상 데이터가 상이한 경우, 프로세서(113)는 산출된 데이터의 인덱스에 기초하여, 새로운 범위를 산출할 수 있다.
- [0067] 만약, 수신된 데이터가 검색 대상 데이터보다 큰 경우, 프로세서(113)는 데이터의 인덱스에 기초하여 새로운 최대값을 설정할 수 있다. 그리고 프로세서(113)는 이전 범위의 최소값 및 새로운 최대값을 이용하여 새로운 범위를 산출할 수 있다.
- [0068] 또한, 수신된 데이터가 검색 대상 데이터보다 작을 경우, 프로세서(113)는 데이터의 인덱스에 기초하여 새로운 최소값을 설정할 수 있다. 그리고 프로세서(113)는 새로운 최소값 및 이전 범위의 최대값에 기초하여 새로운 범위를 산출할 수 있다.
- [0069] 프로세서(113)는 산출된 새로운 범위 및 키에 기초하여, 데이터의 인덱스를 재산출할 수 있다. 그리고 프로세서(113)는 재산출된 데이터의 인덱스를 서버(120)에 전송하고, 데이터의 인덱스에 대응하는 데이터를 다시 수신할 수 있다.
- [0070] 프로세서(113)는 수신된 데이터가 검색 대상 데이터와 동일할 때까지 범위를 재설정하고 검색하는 과정을 반복할 수 있다. 만약, 수신된 데이터와 검색 대상 데이터가 동일한 경우, 프로세서(113)는 사용자에게 수신된 데이터를 제공할 수 있다.
- [0071] 도 5는 본 발명의 일 실시예에 따른 데이터 검색의 예시도이다.
- [0072] 예를 들어, 도 5를 참조하면, 서버(120)는 원시 데이터 저장 시 단말(110)의 프로세서(113)가 선택한 순열 정보를 저장할 수 있다. 또한, 서버(120)는 데이터베이스(122)에 단말(110)의 프로세서(113)로부터 수신한 데이터를 저장할 수 있다.
- [0073] 프로세서(113)는 사용자로부터 "No 속성"의 값이 "920311"인 데이터에 대한 검색 요청을 수신할 수 있다. 즉, 검색 대상 데이터는 "No 속성"의 값이 "920311"인 데이터가 될 수 있다.
- [0074] 먼저, 프로세서(113)는 검색 대상 데이터에 대한 검색 범위를 설정할 수 있다. 앞에서 설명한 바와 같이, 데이터의 크기는 유계 행렬 및 균형 행렬의 차원의 곱보다 작거나 같을 수 있다. 그러므로 프로세서(113)는 기저장된 순열 정보에 기초하여, 데이터의 크기를 예측할 수 있다.
- [0075] 순열 정보에 저장된 유계 행렬이 "4 x 4 행렬"(4차원)이고, 균형 행렬이 "3 x 3"(3차원)이므로 프로세서(113)는 데이터의 크기가 두 행렬의 차원의 곱인 12보다 작을 것으로 예측할 수 있다. 그러므로 프로세서(113)는 데이터베이스(122)에 저장된 데이터에 대한 검색 범위를 인덱스 "0"이상 인덱스 "11" 이하로 설정할 수 있다.
- [0076] 프로세서(113)는 설정된 범위의 중앙값을 선택하여, 검색을 위하여 사용될 데이터 암호화 행렬의 열을 선택할 수 있다. 이때, 프로세서(113)는 [수학식 2]에 기초하여, 중앙값을 선택할 수 있다. [수학식 2]에서 low 는 범위의 최소값, $high$ 는 범위의 최대값이다. 또한, y 는 검색에 사용될 열이며, $floor()$ 함수는 소수점 이하를 버리는 함수이다.

수학적식 2

$$y = \text{floor}\left(\frac{\text{low} + \text{high}}{2}\right)$$

[0077]

[0078]

[수학적식 2]에서 현재 설정된 범위를 통하여, $\text{low} = 0$, $\text{high} = 11$ 이므로, y 는 5가 된다. 즉, 프로세서(113)는 검색에 사용될 데이터 암호화 행렬의 열 인덱스로 "5"를 산출할 수 있다. 그리고 데이터 저장 시 사용한 키가 "6"이므로 프로세서(113)는 순열 정보를 통하여, 산출된 데이터 암호화 행렬 "M(6, 5)"의 값을 서버(120)로 전송할 데이터의 인덱스로 사용할 수 있다. 즉, 데이터 암호화 행렬의 7행 6열의 값이 10이므로, 프로세서(113)는 데이터의 인덱스로 "10"을 설정할 수 있다. 그리고 프로세서(113)는 서버(120)로 "10"을 전달할 수 있다.

[0079]

서버(120)는 단말(110)로부터 "10"을 수신하면, 데이터베이스(122)에 저장된 데이터 중 데이터의 인덱스가 "10"인 11번째 데이터를 검색할 수 있다. 그리고 서버(120)는 11번째 데이터인 "<'Frank', 901018, 'M', 'GGG'>"를 단말(110)로 전달할 수 있다.

[0080]

단말(110)의 프로세서(113)는 통신 모듈(111)을 통하여, 서버(120)로부터 "<'Frank', 901018, 'M', 'GGG'>"를 수신할 수 있다. 그리고 검색 대상 데이터와 서버(120)로부터 수신한 데이터를 비교할 수 있다. 검색 대상 데이터의 "No 속성" 값 "920311"보다 수신한 데이터의 "No 속성" 값 "901018"은 작다. 그러므로 프로세서(113)는 검색에 사용된 데이터 암호화 행렬의 열 인덱스 "5"보다 1 큰 값인 "6"을 새로운 low 로 다시 설정할 수 있다. 그러므로 새로운 범위에서 low 는 6이며, high 는 11이 될 수 있다.

[0081]

[수학적식 1]에 기초하여, 프로세서(113)는 새로운 범위에 대한 중앙값인 "8"을 선택할 수 있다. 즉, 프로세서(113)는 데이터 암호화 행렬 "M(6, 8)"의 값인 "1"을 데이터의 인덱스로 선택할 수 있다. 프로세서(113)는 선택된 검색에 사용될 데이터의 인덱스 "1"을 서버(120)로 전송하고 서버(120)로부터 인덱스에 대응하는 데이터베이스(122)에 2번째 데이터인 "<'Isabella', 910815, 'F', 'CCC'>"를 수신할 수 있다.

[0082]

프로세서(113)는 수신한 데이터에 포함된 "No 속성"의 값 "910815"와 검색 대상 데이터의 "No 속성" 값 "920311"을 비교할 수 있다. 서버(120)로부터 수신한 값이 작으므로, 프로세서(113)는 검색에 사용된 열인 8보다 1 큰 값인 9를 새로운 low 로 다시 설정할 수 있다. 그러므로 새로운 low 는 9이며, high 는 11이 될 수 있다.

[0083]

위에서 설명한 방법에 따라, 프로세서(113)는 새로운 범위에 따라, 데이터 암호화 행렬 M의 6행 10열의 값인 8을 검색에 사용될 인덱스로 사용할 인덱스로 선택할 수 있다. 프로세서(113)는 서버(120)로 8을 전달하고, 서버(120)로부터 인덱스 8에 대응하는 데이터인 "<'Kelly', 930208, 'F', 'AAA'>"를 수신할 수 있다.

[0084]

프로세서(113)는 수신한 데이터에 포함된 No 속성 값 "930208"과 사용자의 검색 요청인 No 속성 값 "920311"을 비교할 수 있다. 서버(120)로부터 수신한 값이 크므로, 프로세서(113)는 검색에 사용된 데이터 암호화 행렬의 열 인덱스인 "10"보다 1 작은 값인 "9"를 범위의 새로운 high 로 다시 설정할 수 있다. 그러므로 범위의 새로운 low 및 high 는 모두 9가 될 수 있다.

[0085]

다시, 프로세서(113)는 새로운 범위에 따라, 데이터 암호화 행렬 M(6, 9)의 값인 "6"을 검색에 사용될 데이터의 인덱스로 선택할 수 있다. 프로세서(113)는 서버(120)로 데이터 인덱스 "6"을 전달하고, 서버(120)로부터 데이터 인덱스 "6"에 대응하는 데이터인 "<'John', 920311, 'M', 'AAA'>"를 수신할 수 있다.

[0086]

프로세서(113)는 수신한 데이터에 포함된 "No 속성" 값 "920311"과 검색 대상 데이터의 "No 속성" 값 "920311"을 비교할 수 있다. 서버(120)로부터 수신한 "No 속성" 값과 검색 대상 데이터의 "No 속성" 값이 같으므로, 프로세서(113)는 검색을 완료하고 검색 결과인 "<'John', 920311, 'M', 'AAA'>"를 사용자에게 제공할 수 있다.

[0087]

한편, 본 발명의 일 실시예에 따른 단말(110)은 검색을 수행할 때마다 데이터 암호화 행렬을 생성하거나, 서버로부터 데이터 암호화 행렬을 수신하여 사용할 수 있다. 구체적으로 단말(110)은 검색을 수행할 때, 데이터 암호화 행렬에 대응하는 행과 열 인덱스를 서버(120)로 전달할 수 있다. 그리고 단말(110)은 행과 열의 인덱스에 대응하는 데이터 암호화 행렬의 값을 서버(120)로부터 수신하여 검색에 사용될 데이터의 인덱스를 설정할 수 있다.

[0088]

그러므로 서버(120)는 단말(110)이 검색을 수행할 때 또는 단말(110)의 요청에 따라, 미리 단말(110)이 선택한

순열 정보에 기초하여, 데이터 암호화 행렬을 생성할 수 있다. 그리고 서버(120)는 단말(110)로부터 행과 열을 수신하면 생성된 데이터 암호화 행렬로부터 인덱스를 산출하여, 단말(110)로 전송할 수 있다.

[0089] 또한, 본 발명의 다른 실시예에 따른 단말(110)은 [수학식 3] 내지 [수학식 5]를 통하여, 검색을 수행할 데이터의 인덱스를 산출할 수 있다.

[0090] 구체적으로 단말(110)은 [수학식 3]을 통하여, 데이터 암호화 행렬의 행(R)에 따른 유계 행렬의 행(q_x) 및 균형 행렬의 행(r_x)을 산출할 수 있다. 예를 들어, 행의 인덱스가 "6"이고, 균형 행렬의 차원의 크기가 "3"이므로, [수학식 3]은 " $6 = q_x \times 3 + r_x$ "가 될 수 있다. 이때, " r_x "은 균형 행렬의 차원의 수보다 작아야 하므로, " q_x "는 "2"가 되고, " r_x "는 "0"이 될 수 있다.

수학식 3

$$R = q_x \times m_1 + q_y$$

[0091]

[0092] [수학식 4]를 통하여, 데이터 암호화 행렬의 열(C)에 따른 유계 행렬의 열(q_y) 및 균형 행렬의 열(r_y)을 산출할 수 있다. 예를 들어, 열의 인덱스가 "5"이고, 균형 행렬의 차원의 크기가 "3"이므로, [수학식 4]는 " $5 = q_y \times 3 + r_y$ "가 될 수 있다. 이때, " r_y "은 균형 행렬의 차원의 수보다 작아야 하므로, " q_y "는 "1"이 되고, " r_y "는 "2"가 될 수 있다.

수학식 4

$$C = r_y \times m_1 + r_y$$

[0093]

[0094] 즉, 단말(110)의 프로세서(113)는 [수학식 3] 및 [수학식 4]를 통하여, 데이터 암호화 행렬의 행(R) 및 열(C)에 따른 유계 행렬에서의 행(q_x) 및 열(q_y)과 균형 행렬에서의 행(r_x) 및 열(r_y)을 산출할 수 있다. 그리고 [수학식 5] 및 각각의 행렬의 값에 기초하여, 프로세서(113)는 데이터의 인덱스(i)를 산출할 수 있다.

수학식 5

$$i = M_2(q_x, q_y) \times m_1 + M_1(r_x, r_y)$$

[0095]

[0096] 한편, 본 발명의 일 실시예에 따른 단말(110)의 프로세서(113)는 원시 데이터에 데이터가 추가되는 경우를 대비하여, 초기 원시 데이터 저장 시 충분한 데이터 크기를 고려하여, 순열 정보를 선택할 수 있다. 그리고 데이터가 추가되면, 프로세서(113)는 기 선택된 순열 정보에 기초하여, 추가 데이터에 대한 재정렬을 수행하여, 서버(120)로 전송할 수 있다.

[0097] 서버(120)는 추가 데이터가 수신되면, 데이터베이스(122)의 기존 데이터가 저장된 데이터 공간에 수신된 추가 데이터를 저장할 수 있다.

[0098] 또한, 본 발명의 다른 실시예에 따른 프로세서(113)는 원시 데이터에 데이터가 추가되면, 서버(120)로 전송하여, 데이터베이스(122)에 기존 저장된 데이터에 링크드 리스트(linked-list) 형태로 추가 데이터를 저장할 수 있다.

[0099] 구체적으로 단말(110)의 프로세서(113)는 추가 데이터가 발생하면, 순열 정보를 선택하고, 선택된 순열 정보에

기초하여, 추가 데이터를 재정렬할 수 있다. 그리고 재정렬된 추가 데이터를 서버(120)로 전송할 수 있다.

- [0100] 서버(120)는 수신한 추가 데이터를 데이터베이스(122)에 저장할 수 있다. 이때, 서버(120)는 수신한 추가 데이터를 기존에 저장된 데이터에 링크드 리스트 형태로 연결할 수 있다. 그러므로 서버(120)에 저장되는 추가 데이터는 기존 데이터와 연결될 수 있다.
- [0101] 그리고 본 발명의 또 다른 실시예에 따른 프로세서(113)는 원시 데이터에 데이터가 추가되면, 기존에 저장된 데이터와 추가 데이터를 병합하여 재정렬할 수 있다. 이를 위하여, 프로세서(113)는 병합된 데이터를 위한 순열 정보를 다시 선택할 수 있다. 그리고 프로세서(113)는 다시 선택된 순열 정보에 따라, 재정렬된 병합 데이터를 서버(120)로 전송하여 저장되도록 할 수 있다.
- [0102] 다음은 도 6을 참조하여, 본 발명의 일 실시예에 따른 데이터 검색 단말(110)의 검색 방법을 설명한다.
- [0103] 도 6은 본 발명의 일 실시예에 따른 데이터 검색 단말(110)의 검색 방법의 순서도이다.
- [0104] 데이터 검색 단말(110)은 범위 및 키에 기초하여, 상기 서버(120)에 저장된 데이터 중 검색 대상 데이터에 대응하는 인덱스를 산출한다(S600). 이때, 범위는 서버(120)로부터 수신한 순열 정보에 기초하여 산출된 것이다. 또한, 키는 서버(120)에 저장된 데이터에 대응하여 설정된 것이다.
- [0105] 또한, 데이터 검색 단말(110)은 산출된 인덱스를 상기 서버(120)로 전송한다(S610).
- [0106] 서버(120)는 서버(120)의 데이터베이스(122)를 통하여, 데이터 검색 단말(110)로부터 산출된 인덱스에 대응하는 데이터를 검색할 수 있다. 서버(120)는 검색된 데이터를 데이터 검색 단말(110)로 전송할 수 있다.
- [0107] 데이터 검색 단말(110)은 서버(120)로부터 산출된 인덱스에 대응하는 데이터를 수신한다(S620).
- [0108] 데이터 검색 단말(110)은 수신된 데이터와 검색 대상 데이터를 비교할 수 있다(S630).
- [0109] 그리고 데이터 검색 단말(110)은 서버(120)로부터 수신한 데이터가 검색 대상 데이터와 매칭되지 않은 경우, 데이터 검색 단말(110)은 범위를 재산출할 수 있다(S640).
- [0110] 그리고 재산출된 범위 및 키에 기초하여, 서버(120)로부터 수신한 데이터가 검색 대상 데이터와 매칭될 때, 인덱스를 산출하고(S600), 산출된 인덱스를 서버(120)로 전송하며(S610), 전송한 인덱스에 대응하는 데이터를 수신하고(S620), 수신된 데이터와 검색 대상 데이터를 비교하는(S630) 과정을 반복하여 수행할 수 있다.
- [0111] 한편, 데이터 검색 단말(110)은 인덱스를 산출하기 이전에 서버(120)의 데이터베이스(122)에 원시 데이터를 저장할 수 있다.
- [0112] 구체적으로 데이터 검색 단말(110)은 서버(120)로부터 순열 정보를 수신할 수 있다.
- [0113] 데이터 검색 단말(110)은 순열 정보에 기초하여, 키를 선택할 수 있다. 그리고 데이터 검색 단말(110)은 선택된 키 및 순열 정보에 기초하여, 원시 데이터를 재정렬할 수 있다.
- [0114] 데이터 검색 단말(110)은 재정렬된 원시 데이터를 서버(120)의 데이터베이스(122)에 저장되도록 서버(120)로 전송할 수 있다.
- [0115] 본 발명의 일 실시예에 따른 데이터 검색 단말(110) 및 그의 검색 방법은 단말(110)에서 데이터를 재정렬하여, 서버(120)로 전송하여 서버(120)의 데이터베이스(122)에 저장할 수 있다. 그러므로 데이터 검색 단말(110) 및 그의 검색 방법은 서버(120)에 저장된 데이터의 안정성을 보장할 수 있다. 또한, 데이터 검색 단말(110) 및 그의 검색 방법은 이진 검색을 통하여, 서버(120)에 저장된 암호화된 데이터에 대한 효율적인 검색을 수행할 수 있다. 그러므로 데이터 검색 단말(110) 및 그의 검색 방법은 데이터 유출을 방지하고, 암호화된 데이터의 신속한 검색을 제공할 수 있다.
- [0116] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전

송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

[0117] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.

[0118] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0119] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0120] 100: 데이터 검색 시스템

110: 단말

111: 통신 모듈

112: 메모리

113: 프로세서

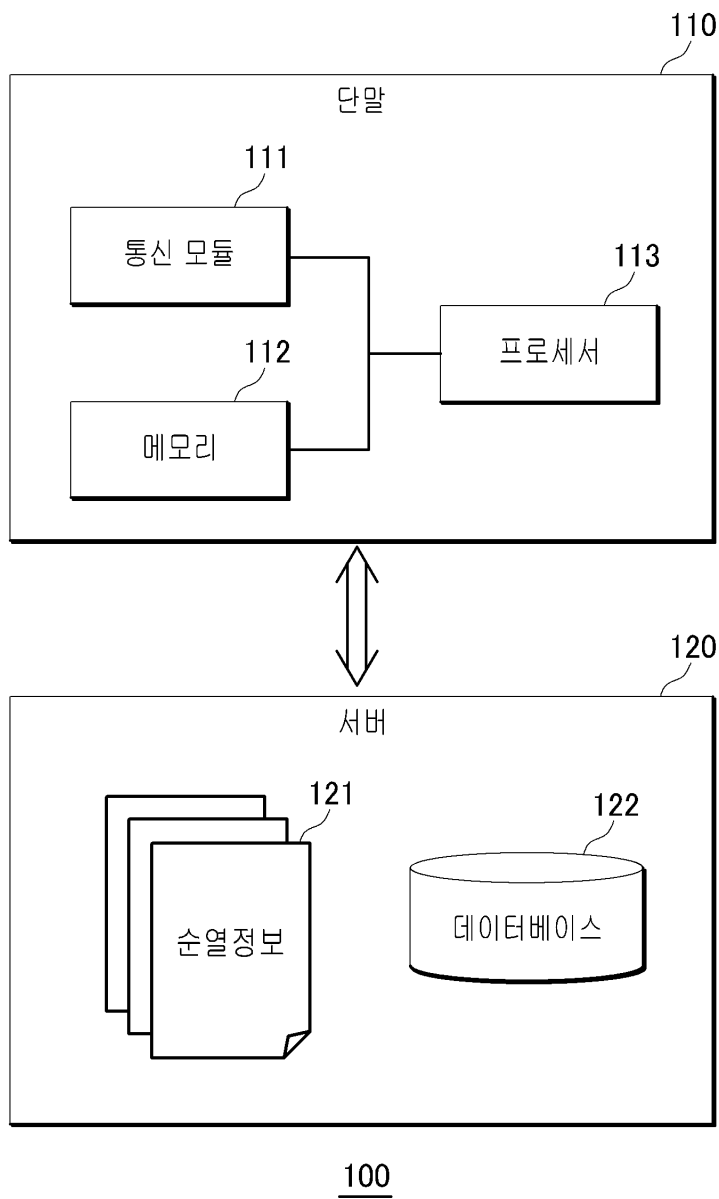
120: 서버

121: 순열 정보

122: 데이터베이스

도면

도면1



도면2

$$M_1 = \begin{array}{|c|c|c|} \hline 0 & 2 & 1 \\ \hline 2 & 1 & 0 \\ \hline 1 & 0 & 2 \\ \hline \end{array} \quad M_2 = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 2 & 3 \\ \hline 0 & 2 & 3 & 1 \\ \hline 1 & 3 & 0 & 2 \\ \hline 1 & 2 & 0 & 3 \\ \hline \end{array}$$

(a) (b)

도면3

$$M = M_2 \times M_1 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 2 & 1 & 3 & 5 & 4 & 6 & 8 & 7 & 9 & 11 & 10 \\ \hline 2 & 1 & 0 & 5 & 4 & 3 & 8 & 7 & 6 & 11 & 10 & 9 \\ \hline 1 & 0 & 2 & 4 & 3 & 5 & 7 & 6 & 8 & 10 & 9 & 11 \\ \hline 0 & 2 & 1 & 6 & 8 & 7 & 9 & 11 & 10 & 3 & 5 & 4 \\ \hline 2 & 1 & 0 & 8 & 7 & 6 & 11 & 10 & 9 & 5 & 4 & 3 \\ \hline 1 & 0 & 2 & 7 & 6 & 8 & 10 & 9 & 11 & 4 & 3 & 5 \\ \hline 3 & 5 & 4 & 9 & 11 & 10 & 0 & 2 & 1 & 6 & 8 & 7 \\ \hline 5 & 4 & 3 & 11 & 10 & 9 & 2 & 1 & 0 & 8 & 7 & 6 \\ \hline 4 & 3 & 5 & 10 & 9 & 11 & 1 & 0 & 2 & 7 & 6 & 8 \\ \hline 3 & 5 & 4 & 6 & 8 & 7 & 0 & 2 & 1 & 9 & 11 & 10 \\ \hline 5 & 4 & 3 & 8 & 7 & 6 & 2 & 1 & 0 & 11 & 10 & 9 \\ \hline 4 & 3 & 5 & 7 & 6 & 8 & 1 & 0 & 2 & 10 & 9 & 11 \\ \hline \end{array}$$

도면4

abc	Name	1234	No	abc	Gender	abc	Disease
	'Green'		901212		'M'		'AAA'
	'Isabella'		910815		'F'		'CCC'
	'Honey'		910418		'M'		'DDD'
	'Alice'		870302		'F'		'AAA'
	'Carol'		881220		'M'		'BBB'
	'Bob'		870625		'M'		'BBB'
	'John'		920311		'M'		'AAA'
	'Lily'		950923		'F'		'BBB'
	'Kelly'		930208		'F'		'AAA'
	'Dave'		890523		'M'		'CCC'
	'Frank'		901018		'M'		'GGG'
	'Erin'		900715		'M'		'CCC'

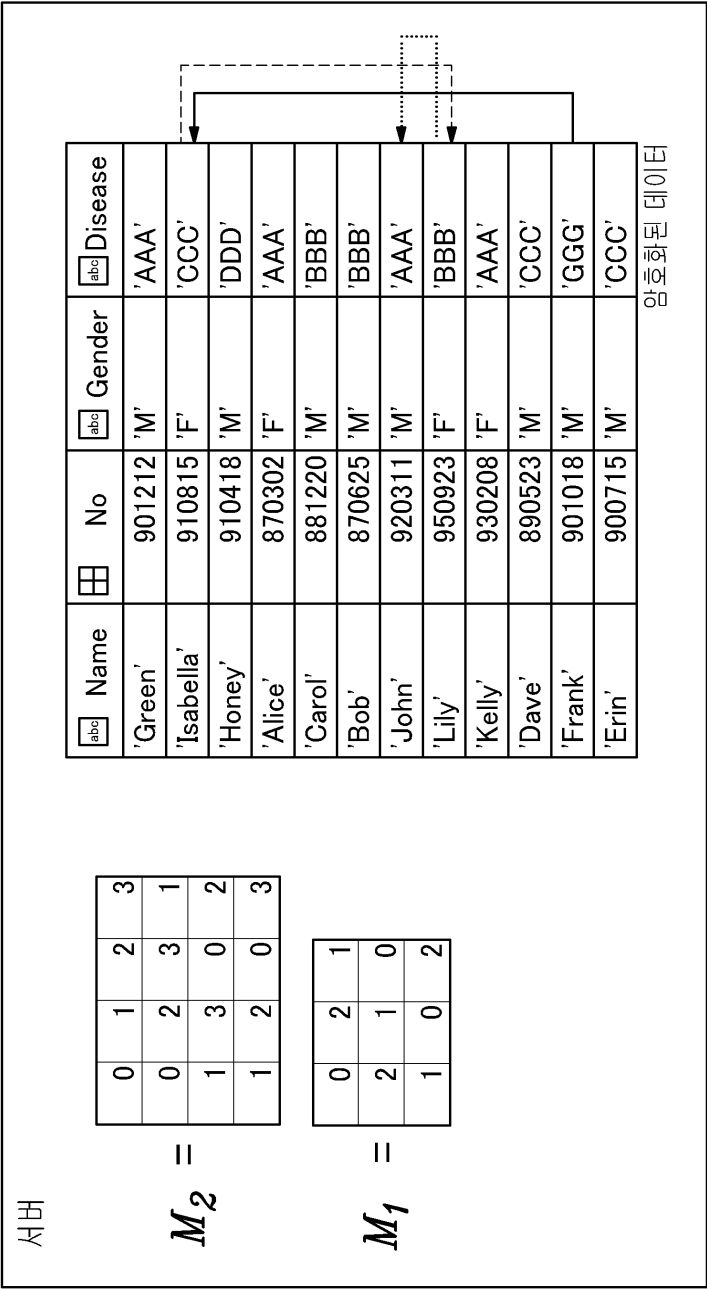
암호화 및 재정렬

(b)

abc	Name	1234	No	abc	Gender	abc	Disease
	'Alice'		870302		'F'		'AAA'
	'Bob'		870625		'M'		'BBB'
	'Carol'		881220		'M'		'BBB'
	'Dave'		890523		'M'		'CCC'
	'Erin'		900715		'M'		'CCC'
	'Frank'		901018		'M'		'GGG'
	'Green'		901212		'M'		'AAA'
	'Honey'		910418		'M'		'DDD'
	'Isabella'		910815		'F'		'CCC'
	'John'		920311		'M'		'AAA'
	'Kelly'		930208		'F'		'AAA'
	'Lily'		950923		'F'		'BBB'

(a)

도면5



도면6

