



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0031079
(43) 공개일자 2009년03월25일

(51) Int. Cl.

G06F 17/30 (2006.01)

(21) 출원번호 10-2007-0096916

(22) 출원일자 2007년09월21일

심사청구일자 2007년09월21일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

박현아

서울 성북구 안암동5가 고려대학교 정보경영공학
전문대학원정보보호프로토콜연구실

김범한

서울 성북구 안암동5가 고려대학교 정보경영공학
전문대학원정보보호프로토콜연구실

(뒷면에 계속)

(74) 대리인

현종철

전체 청구항 수 : 총 14 항

(54) 암호화된 문서의 근사 스트링 매칭 방법, 이를 기록한기록매체 및 그 시스템

(57) 요약

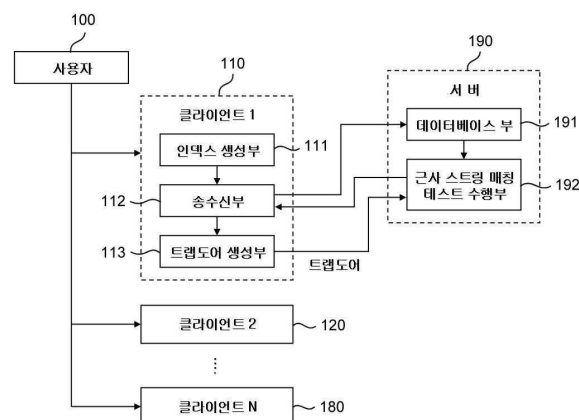
암호화된 문서의 근사 스트링 매칭 방법, 이를 기록한 기록매체 및 그 시스템이 개시된다.

본 발명에 따른 암호화된 문서의 근사 스트링 매칭 방법은,

인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 상기 인증된 사용자의 검색키로 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성하는 단계; 상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하면, 상기 서버에서 상기 전송된 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장하는 단계; 상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의하는 단계; 및 상기 서버에서 상기 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는 단계를 포함한다.

본 발명에 의하면, 데이터베이스 테이블에 저장된 인덱스와 클라이언트가 질의한 트랩도어로 패턴스트링을 생성하여 암호문에서 유사 문서 검색을 위해 스트링 매칭 테스트를 수행함으로써, 유사 검색에 따른 검색 실패 확률을 감소시켜 데이터의 가용성을 높일 수 있으며, 평문 검색 시스템과 암호문 검색 시스템의 괴리를 좁혀 암호 검색 시스템의 상용화를 실현할 수 있어 데이터 베이스의 일원화를 이룰 수 있으며, 이를 통하여 효율적이고 안전한 암호화된 데이터 검색을 수행할 수 있으며, 암호화된 문서의 체계적인 관리 뿐만 아니라 안전한 정보 체계화 시스템 구축을 실현할 수 있어 데이터 검색의 신뢰도와 공신력을 이룰 수 있는 효과가 있다.

대표도 - 도1



(72) 발명자

이동훈

서울 성북구 안암동5가 고려대학교 정보경영공학전
문대학원창의관 510호

임종인

서울 성북구 안암동5가 고려대학교 정보경영공학전
문대학원창의관 510호

특허청구의 범위

청구항 1

인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 상기 인증된 사용자의 검색기로 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성하는 단계;

상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하면, 상기 서버에서 상기 전송된 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장하는 단계;

상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의하는 단계; 및

상기 서버에서 상기 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는 단계를 포함하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 2

제 1 항에 있어서,

상기 근사 스트링 매칭 테스트를 수행하는 단계는,

상기 근사 매칭 테스트를 만족하는 문서가 존재하는 경우 상기 문서를 클라이언트에게 리턴하는 단계; 및

상기 클라이언트에서 상기 암호화된 문서를 복호화하여 상기 인증된 사용자에게 전송하는 단계를 더 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 3

제 1 항에 있어서,

상기 문서에 대한 인덱스 스트링은 상기 데이터베이스 테이블의 각 행의 각 셀에 저장되며,

상기 인덱스 스트링은

상기 키워드, 상기 인증된 사용자의 식별자, 상기 문서의 식별자 및 상기 문서의 속성 정보에 따른 필드 식별자를 이용하여 암호화된 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 4

제 3 항에 있어서,

상기 문서에 대한 인덱스를 생성하는 단계는

상기 클라이언트에서 설정된 암호화 파라미터에 따라 암호화 시스템 구축을 위한 그룹 파라미터, 일방향 해쉬 함수 및 슈도 랜덤 함수를 포함하는 암호화 함수를 생성하는 단계;

상기 암호화 파라미터를 이용하여 상기 인증된 사용자의 검색키를 생성하는 단계; 및

상기 생성된 검색키, 상기 문서의 키워드, 상기 문서의 식별자, 상기 문서의 필드 식별자를 이용하여 상기 문서의 인덱스를 생성하는 단계를 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 5

제 4 항에 있어서,

상기 문서의 인덱스를 생성하는 단계는,

상기 생성된 검색키 및 상기 문서의 필드 식별자를 이용하여 상기 문서의 키워드의 철자별 제 1 암호값을 생성하는 단계;

상기 생성된 제 1 암호값에 상기 문서의 식별자를 이용하여 상기 문서의 키워드의 철자별 제 2 암호값을 생성하

는 단계; 및

상기 생성된 제 2 암호값에 상기 문서의 식별자를 부가하여 상기 문서의 인덱스를 생성하는 단계를 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 6

제 1 항에 있어서,

상기 트랩도어는

상기 인증된 사용자의 질의시마다 랜덤하게 생성되는 랜덤값을 이용하여 생성되며,

상기 트랩도어는 모듈 연산에 의해 생성되는 것을 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 7

제 1 항에 있어서,

상기 근사 스트링 매칭 테스트를 수행하는 단계는

생성된 패턴 스트링 P 와 i 번째 문서의 j 번째 인덱스 스트링 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 연산하는 단계; 및

상기 연산된 해밍 디스턴스 값 $H(P, I_{ij})$ 이 미리 설정된 유사도 k 와 $H(P, I_{ij}) \leq k$ 의 관계가 성립하면, 상기 서버에서 클라이언트에게 상기 암호화된 문서를 전송하는 단계를 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 8

제 3 항에 있어서,

상기 문서에 대한 인덱스를 생성하는 단계는

상기 클라이언트에서 설정된 암호화 파라미터에 따라 암호화 시스템 구축을 위한 그룹 파라미터, 일방향 해쉬 함수 및 슈도 랜덤 함수를 포함하는 암호화 함수를 생성하는 단계;

상기 암호화 파라미터를 이용하여 상기 인증된 사용자의 검색키를 생성하는 단계; 및

상기 생성된 검색키, 상기 문서의 키워드, 상기 문서의 키워드 스트링의 철자 순서값, 상기 문서의 식별자 및 상기 문서의 필드 식별자를 이용하여 상기 문서의 인덱스를 생성하는 단계를 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 9

제 8 항에 있어서,

상기 문서의 인덱스를 생성하는 단계는,

상기 생성된 검색키, 상기 문서의 필드 식별자 및 상기 문서의 키워드 스트링의 철자 순서값을 이용하여 상기 문서의 키워드의 철자별 제 1 암호값을 생성하는 단계;

상기 생성된 제 1 암호값에 상기 문서의 식별자를 이용하여 상기 문서의 키워드의 철자별 제 2 암호값을 생성하는 단계; 및

상기 생성된 제 2 암호값에 상기 문서의 식별자를 부가하여 상기 문서의 인덱스를 생성하는 단계를 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 방법.

청구항 10

제 1 항 내지 제 9 항 중 어느 하나의 방법을 컴퓨터에서 수행하도록 프로그램으로 기록된 기록매체.

청구항 11

인증된 사용자, 복수의 클라이언트 및 서버를 포함하는 암호화된 문서의 근사 스트링 매칭 시스템에 있어서,

상기 복수의 클라이언트는

상기 인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 상기 인증된 사용자의 검색키를 이용하여 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성하는 인덱스 생성부;

상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하는 송수신부; 및

상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의하는 트랩도어 생성부를 포함하고,

상기 서버는

데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장하는 데이터 베이스부; 및

상기 클라이언트에서 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는 근사 스트링 매칭 테스트 수행부를 포함하는 암호화된 문서의 근사 스트링 매칭 시스템.

청구항 12

제 11 항에 있어서,

상기 문서에 대한 인덱스 스트링은 상기 데이터베이스 테이블의 각 행의 각 셀에 저장되며,

상기 인덱스 스트링은

상기 키워드, 상기 인증된 사용자의 식별자, 상기 문서의 식별자 및 상기 문서의 속성 정보에 따른 필드 식별자를 이용하여 암호화된 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 시스템.

청구항 13

제 11 항에 있어서,

상기 트랩도어는

상기 인증된 사용자의 질의시마다 랜덤하게 생성되는 랜덤값을 이용하여 생성되며,

상기 트랩도어는 모듈 연산에 의해 생성되는 것을 포함하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 시스템.

청구항 14

제 11 항에 있어서,

상기 근사 스트링 매칭 테스트 수행부는

생성된 패턴 스트링 P 와 i 번째 문서의 J 번째 인덱스 스트링 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 연산하고, 상기 연산된 해밍 디스턴스 값 $H(P, I_{ij})$ 이 미리 설정된 유사도 k 와

$$H(P, I_{ij}) \leq k$$

의 관계가 성립하면, 상기 클라이언트에게 상기 암호화된 문서를 전송하는 것을 특징으로 하는 암호화된 문서의 근사 스트링 매칭 시스템.

명세서

발명의 상세한 설명

기술 분야

- <1> 본 발명은 데이터 암호화에 관한 것으로, 더욱 상세하게는, 암호화된 문서에 대하여 신뢰도 및 공신력이 우수한 유사 검색을 수행할 수 있는 암호화된 문서의 근사 스트링 매칭 방법, 이를 기록한 기록매체 및 그 시스템에 관한 것이다.

배경 기술

- <2> 암호화는 시큐리티에 대처하는 가장 강력한 수단으로써, 정보화 사회에서 정보들의 저장소인 데이터베이스의 관리의 무엇보다 중요하다. 중요 정보 및 개인정보를 보호하기 위한 가장 현실적인 방법은 이러한 정보들을 저장하는 데이터베이스의 암호화이다.
- <3> 현재까지의 상용 모듈이 제공하고 있는 암호화 방법은 암호화된 문서를 읽은 후에 복호화하여 검색하거나 인덱스 부분은 평문으로 남겨두고 문서만 암호화하여 검색하는 형식이 대부분이다.
- <4> 이러한 과정들은 내부 관리자에 의한 정보 유출을 막지 못할 뿐만 아니라 복호화 순간의 외부 공격과 인덱스 정보 노출로 인한 상관성 공격 등이 큰 문제점으로 지적되고 있다.
- <5> 현재, 암호화 방법이 현실성이 없었던 가장 큰 이유는 안전성 위주의 이론적 연구에 치중한 나머지 암호화된 데이터베이스 상에서는 효율적인 자료 검색을 보장 받지 못했다는 것과 암호화된 검색 시스템 구현을 위해서는 기존 데이터 베이스 관리 시스템(DataBase Management System:DBMS)을 그대로 사용하지 못하고 구조를 변경하거나 새로 생성해야만 했고, 기존 데이터 베이스 관리 시스템이 제공하는 기능과 스키마를 사용할 수 없었기 때문이다.
- <6> 즉, 종래의 암호화된 문서에 대한 검색 시스템에 관한 연구는 지금까지의 연구는 크게 두 가지 분류로 나누어져 진행되어 왔다.
- <7> 첫 번째 분류는, 암호학자들에 의한 것으로 2000년 Song et al.의 대칭키 기반의 문서 전체를 검색하는 시스템에 관한 연구를 필두로 하여, 그 이후에는 문서의 키워드를 암호화하여 만든 인덱스를 사용하여 검색하는 시스템에 관한 연구가 주류를 이루면서 그것의 안전성 증명을 주목적으로 해왔다.
- <8> 두 번째 분류는, 컴퓨터 사이언스(Computer science) 분야에서 암호학의 안전성 관점보다는 보안의 관점에 보다 치중하여 기존의 데이터 베이스 관리 시스템에 변화를 주지 않고 암호화를 적용할 수 있는 효율적인 암호화 시스템 개발을 목적으로 연구가 진행되어져 왔으나, 안전성 측면에서 많은 문제점을 드러내고 있다.
- <9> 그러나, 암호화된 문서에 대한 효율적인 검색은 민감한 정보 보호뿐만 아니라, 완벽한 프라이버시 보호에 가장 필요한 기술이다. 따라서, 현재 절실히 요구되는 것은 암호화된 문서를 복호화하지 않은 상태에서 검색 가능한 시스템의 구축으로, 이는 보안성 뿐만 아니라 상용 가능하여야 하며, 개인의 비밀 문서를 보관하는 데이터 저장소의 개념보다는 데이터베이스 시스템을 서비스로 사용하는 보다 현실적인 환경을 위한 시스템이어야 하며, 기존의 데이터 베이스의 구조를 변경시키지 않으면서, 그것이 지원하는 스키마의 모든 기능을 사용할 수 있는 시스템의 개발이다.
- <10> 그러나, 사용자의 프라이버시 보호를 위해 사용한 암호화는 비효율성 및 데이터베이스 스키마(Database schema)와 그것이 지원하는 기능의 적용 불가능 등 여러 가지 문제를 발생시킨다. 이런 문제들 중 하나로 유사 문서 검색 역시 암호화된 데이터에 대하여 불가능한 것으로 알려져 왔다.
- <11> 평문에서 1비트 차이가 암호문에서는 엄청난 비트 차이가 될 수 있기 때문에, 검색어 질의시 발생한 오타 하나가 암호문에서는 엄청난 차이를 유발할 수 있어 평문 유사 검색에서 사용되는 텍스트 스트링(text string)과 패턴 스트링(pattern string)에 대해 매칭 테스트를 실시하여 미리 정의되어진 유사도를 만족하는지의 여부를 테스트하는 유사 스트링 매칭 테스트 등과 같은 기법이 암호문에서는 불가능할 것으로 여겨져 왔다.

- <12> 암호문에서 널리 사용되는 검색 방법은 동일성 테스트(equality test)이다. 즉, 질의자가 원하는 검색어를 암호화한 트랩도어(trapdoor)를 보내면 서버는 이것을 가지고 정해진 프로토콜대로 소정의 연산을 수행하여 이것과 일치하는 인덱스를 찾는 것이다.
- <13> 그렇기 때문에 한 비트의 차이라도 있으면, 이것은 검증식에 따른 테스트를 만족할 수 없어 검색에 실패하게 된다.
- <14> 그런데, 상술한 바와 같이 평문에서의 한 비트의 차이는 암호문에서는 엄청난 차이가 될 수 있기 때문에, 평문에서 사용되는 유사 문서 검색 방법인 근사 스트링 매칭(approximate string matching) 방법을 암호문에 적용한다는 것은 아무 의미 없는 일이다.
- <15> 특히, 동일성 테스트로 검색되는 암호문 검색 시스템에서는 키워드 단위로 암호화한 인덱스를 사용하여 테스트하기 때문에 오타로 인한 에러에 대해서는 검색 실패나 엉뚱한 결과를 얻게 된다.
- <16> 따라서, 암호문에서도 유사 문서 검색을 가능하게 하기 위해서는 인덱스를 생성할 때 키워드를 단어별로 암호화할 것이 아니라 그 단어를 이루고 있는 철자 하나하나를 따로 암호화해야 한다.
- <17> 이렇게 하면, 평문의 유사 검색 시스템과 거의 동일한 조건을 갖추게 되어 평문에서 유사 검색을 위해 사용하는 근사 스트링 매칭 방법들, 예를 들어, edit distance, automata, bit-parallelism, filtering 모두가 적용 가능하게 된다.
- <18> 그러나, 철자별 암호화 방법은, 암호화 도메인이 너무 작으며, 영문 알파벳의 갯수는 총 26자이기 때문에 아무리 복잡한 암호 함수를 사용한다 하더라도 암호화되어 생성되는 형태는 단지 26개에 불과하므로, 전수 조사 공격이 충분히 가능하며, 이는 전체 시스템의 붕괴를 초래할 수 있는 문제점이 있다.

발명의 내용

해결 하고자하는 과제

- <19> 따라서, 본 발명이 해결하고자 하는 첫 번째 과제는 기존의 키워드 인덱스 검색을 기반으로 키워드의 철자별 단위로 암호화된 문서의 검색을 수행할 수 있고, 공격자에 대한 전수 조사 공격을 차단하여 시스템을 안전하게 유지할 수 있는 암호화된 문서의 근사 스트링 매칭 방법을 제공하는 것이다.
- <20> 또한, 본 발명이 해결하고자 하는 두 번째 과제는 상기 암호화된 문서의 근사 스트링 매칭 방법을 이용한 암호화된 문서의 근사 스트링 매칭 시스템을 제공하는 것이다.
- <21> 그리고, 본 발명이 해결하고자 하는 세 번째 과제는 상기 암호화된 문서의 근사 스트링 매칭 방법을 컴퓨터에서 수행할 수 있는 프로그램으로 기록된 기록매체를 제공하는 것이다.

과제 해결수단

- <22> 상기 첫 번째 과제를 해결하기 위하여 본 발명은,
- <23> 인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성하는 단계; 상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하면, 상기 서버에서 상기 전송된 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장하는 단계; 상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의하는 단계; 및 상기 서버에서 상기 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는 단계를 포함하는 암호화된 문서의 근사 스트링 매칭 방법을 제공한다.
- <24> 한편, 상기 근사 스트링 매칭 테스트를 수행하는 단계는, 상기 근사 매칭 테스트를 만족하는 문서가 존재하는 경우 상기 문서를 클라이언트에게 리턴하여, 상기 클라이언트에서 상기 인증된 사용자에게 상기 암호화된 문서를 복호화하여 상기 인증된 사용자에게 전송하는 단계를 더 포함하는 것을 특징으로 한다.
- <25> 그리고, 상기 문서에 대한 인덱스 스트링은 상기 데이터베이스 테이블의 각 행의 각 셀에 저장되며, 상기 인덱스 스트링은 상기 키워드, 상기 인증된 사용자의 식별자, 상기 문서의 식별자 및 상기 문서의 속성 정보에 따른

필드 식별자를 이용하여 암호화된 것을 특징으로 한다.

- <26> 또한, 상기 문서에 대한 인덱스를 생성하는 단계는 상기 클라이언트에서 암호화 검색 시스템 구축을 위한 시스템 파라미터, 일방향 해쉬 함수 및 슈도 랜덤 함수를 포함하는 암호화 함수를 생성하는 단계, 상기 생성된 암호화 함수를 이용하여 상기 인증된 사용자의 검색키를 생성하는 단계, 및 상기 생성된 검색키, 상기 문서의 키워드, 상기 문서의 식별자, 상기 문서의 필드 식별자를 이용하여 상기 문서의 인덱스를 생성하는 단계를 포함한다.
- <27> 아울러, 상기 문서의 인덱스를 생성하는 단계는, 상기 생성된 검색키 및 상기 문서의 필드 식별자를 이용하여 상기 문서의 키워드의 철자별 제 1 암호값을 생성하는 단계; 상기 생성된 제 1 암호값에 상기 문서의 식별자를 이용하여 상기 문서의 키워드의 철자별 제 2 암호값을 생성하는 단계; 및 상기 생성된 제 2 암호값에 상기 문서의 식별자를 붙여 상기 문서의 인덱스를 생성하는 단계를 포함한다.
- <28> 그리고, 상기 트랩도어는 상기 인증된 사용자의 질의시마다 랜덤하게 생성되는 랜덤값을 이용하여 생성되며, 상기 트랩도어는 모듈 연산에 의해 생성되는 것을 포함한다.
- <29> 또한, 근사 스트링 매칭 테스트를 수행하는 단계는 생성된 패턴 스트링 P 와 i 번째 문서의 j 번째 인덱스 스트링 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 연산하는 단계; 및 상기 연산된 해밍 디스턴스 값 $H(P, I_{ij})$ 이 미리 설정된 유사도 k 와 $H(P, I_{ij}) \leq k$ 의 관계가 성립하면, 상기 서버에서 클라이언트에게 해당 문서를 전송하는 단계를 포함하는 것을 특징으로 한다.
- <30> 그리고, 상기 문서에 대한 인덱스를 생성하는 단계는 상기 클라이언트에서 암호화 검색을 수행하기 위한 시스템 파라미터, 일방향 해쉬 함수 및 슈도 랜덤 함수를 포함하는 암호화 함수를 생성하는 단계; 상기 클라이언트에서 상기 일방향 해쉬 함수 및 슈도 랜덤 함수를 이용하여 암호화 파라미터를 생성하는 단계; 상기 암호화 파라미터를 이용하여 상기 인증된 사용자의 검색키를 생성하는 단계; 및 상기 생성된 검색키, 상기 문서의 키워드, 상기 문서의 키워드 스트링의 철자 순서값, 상기 문서의 식별자 및 상기 문서의 필드 식별자를 이용하여 상기 문서의 인덱스를 생성하는 단계를 포함한다.
- <31> 또한, 상기 문서의 인덱스를 생성하는 단계는, 상기 생성된 검색키, 상기 문서의 필드 식별자 및 상기 문서의 키워드 스트링의 철자 순서값을 이용하여 상기 문서의 키워드의 철자별 제 1 암호값을 생성하는 단계; 상기 생성된 제 1 암호값에 상기 문서의 식별자를 이용하여 상기 문서의 키워드의 철자별 제 2 암호값을 생성하는 단계; 및 상기 생성된 제 2 암호값에 상기 문서의 식별자를 붙여 상기 문서의 인덱스를 생성하는 단계를 포함한다.
- <32> 상기 두 번째 과제를 달성하기 위하여 본 발명은,
- <33> 인증된 사용자, 복수의 클라이언트 및 서버를 포함하는 암호화된 문서의 근사 스트링 매칭 시스템에 있어서, 상기 복수의 클라이언트는 상기 인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성하는 인덱스 생성부; 상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하는 송수신부; 및 상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의하는 트랩도어 생성부를 포함하고, 상기 서버는 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장하는 데이터 베이스부; 및 상기 클라이언트에서 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는 근사 스트링 매칭 테스트 수행부를 포함하는 암호화된 문서의 근사 스트링 매칭 시스템을 제공한다;
- <34> 그리고, 상기 문서에 대한 인덱스 스트링은 상기 데이터베이스 테이블의 각 행의 각 셀에 저장되며, 상기 인덱스 스트링은 상기 키워드, 상기 인증된 사용자의 식별자, 상기 문서의 식별자 및 상기 문서의 속성 정보에 따른 필드 식별자를 이용하여 암호화된 것을 특징으로 한다.
- <35> 아울러, 상기 트랩도어는 상기 인증된 사용자의 질의시마다 랜덤하게 생성되는 랜덤값을 이용하여 생성되며,

상기 트랩도어는 모듈 연산에 의해 생성되는 것을 포함하는 것을 특징으로 한다.

<36> 또한, 상기 근사 스트링 매칭 테스트를 수행부는 생성된 패턴 스트링 P 와 i 번째 문서의 J 번째 인덱스 스트링 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 연산하고, 상기 연산된 해밍 디스턴스 값

$H(P, I_{ij})$ 이 미리 설정된 유사도 k 와 $H(P, I_{ij}) \leq k$ 의 관계가 성립하면, 상기 클라이언트에게 해당 문서를 전송하는 것을 특징으로 한다.

<37> 상기 세 번째 과제를 해결하기 위하여 본 발명은,

<38> 상기 암호화된 문서의 근사 스트링 매칭 방법을 컴퓨터에서 수행하도록 프로그램으로 기록된 기록매체를 제공한다.

효과

<39> 본 발명에 의하면, 유사 검색에 따른 검색 실패 확률을 감소시켜 데이터의 가용성을 높일 수 있으며, 평문 검색 시스템과 암호문 검색 시스템의 괴리를 좁혀 암호 검색 시스템의 상용화를 실현할 수 있어 데이터 베이스의 일원화를 이룰 수 있으며, 이를 통하여 효율적이고 안전한 암호화된 데이터 검색을 수행할 수 있으며, 암호화된 문서의 체계적인 관리 뿐만 아니라 안전한 정보 체계화 시스템 구축을 실현할 수 있어 데이터 검색의 신뢰도와 공신력을 이룰 수 있는 효과가 있다.

발명의 실시를 위한 구체적인 내용

<40> 본 발명은 암호화된 문서에 대해서도 유사 검색을 가능하게 하는 암호화된 문서의 근사 스트링 매칭 방법을 제공하는 것이다. 이를 위하여 본 발명은 기존의 키워드 인덱스 검색 기법을 기반으로 키워드 단위별 암호화가 아닌 철자별 단위의 암호화 방법을 적용한다.

<41> 본 발명은 유사 검색을 가능하게 하기 위해 일반 평문에서 사용하는 근사 스트링 매칭 기법(approximate string matching scheme)을 적용한다. 비록 암호화되었지만, 철자별 암호화이므로 평문에서 근사 스트링 매칭 기법을 적용하는 상황이 되므로, 어떠한 종류의 근사 스트링 매칭 기법이라도 적용할 수 있게 된다.

<42> 그리고, 본 발명은 서버-클라이언트 모델인 DAS(Database As a Server) 모델을 이용한다. 이 모델에서 클라이언트는 제한된 저장 공간과 연산 능력을 가진 반면, 서버는 무한한 저장 공간과 연산 능력을 가지고 있다.

<43> 클라이언트는 이러한 서버의 능력을 이용하여 사용자들의 데이터를 서버에 저장하는데, 이때 클라이언트는 서버가 사용자들의 데이터의 내용을 알 수 없도록 사용자들의 키를 이용하여 암호화하여 서버에 저장한다.

<44> 그리고, 복호화 역시 서버에 저장해 놓은 암호화된 데이터를 가져와 사용자들의 키를 이용하여 복호화한다. 클라이언트는 데이터를 생성하고, 데이터를 암호화, 복호화를 하며, 그 데이터를 관리할 수 있다.

<45> 이에 반하여, 서버는 단지 클라이언트로부터 전송된 데이터를 저장하고, 클라이언트가 요청한 쿼리(query)를 처리하여 그 결과값을 전달해 준다.

<46> 이러한 시스템 모델은 서버 관리자, 즉 내부공격자에 안전한 암호화된 검색 시스템 설계를 가능하게 한다.

<47> 이와 같이, 본 발명은 서버-클라이언트 모델의 일종인 DAS 모델을 기반으로 하여 근사 스트링 매칭의 방법으로 해밍 디스턴스(Hamming distance)를 적용한다.

<48> 이는 스트링 매칭에 있어서 두 스트링 S 와 T 에 대하여 두 스트링의 위치가 다른 철자의 갯수를 말하는

것으로, 두 스트링의 유사도를 k 라고 설정해 놓았을때, i 번째 문서의 J 번째 필드의 인덱스인 I_{ij} 를 이용하여, $H(P, I_{ij}) \leq k$ 를 만족하는 문서들을 출력하게 된다. 암호화된 문서에서 유사 문서 검색을 가능하게 하기 위한 방법으로 키워드 인덱스 검색 기법으로 인한 인덱스를 유사 문서 검색이 가능한 형태가 되도록 암호화

한다.

<49> 이는 각 문서마다 주요한 키워드를 뽑아 철자별로 암호화 방법을 적용하여 암호화하고, 이 암호화된 문서에 대한 인덱스를 생성하여 클라이언트가 생성된 인덱스를 서버에 전송하면 서버는 데이터베이스 테이블의 각 행에 하나의 문서에 대한 인덱스를 저장한다.

<50> 그런 후, 사용자가 클라이언트에게 사용자가 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트는 상기 전송된 키워드를 암호화하여 트랩도어를 생성하여 서버에게 전송한다. 서버는 트랩도어가 생성된 암호화된 키워드를 수신하여 미리 정해진 프로토콜에 의해 패턴 스트링을 생성하고, 생성된 패턴 스트링과 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하는데, 이는 미리 설정된 유사도 k 와 $H(P, I_{ij}) \leq k$ 의 관계가 성립하면, 상기 클라이언트에게 해당 문서를 전송하고, 클라이언트는 이를 복호화하여 사용자에게 문서를 전송하게 된다.

<51> 이하, 본 발명의 바람직한 실시예를 첨부도면에 의거하여 상세히 설명하기로 한다.

<52> 그리고, 하기에 상술되는 사용자는 인증된 사용자를 의미하는 것으로 한정하여 상술하기로 한다.

<53> 한편, 다음에 예시하는 본 발명의 실시예는 여러 가지 다른 형태로 변형할 수 있으며, 본 발명의 범위가 다음에 상술하는 실시예에 한정되는 것은 아니다. 본 발명의 실시예는 당 업계에서 평균적인 지식을 가진 자에게 본 발명을 더욱 완전하게 설명하기 위하여 제공된다.

<54> 도 1은 본 발명에 따른 암호화된 문서의 근사 스트링 매칭 시스템의 블록도이다.

<55> 본 발명에 따른 암호화된 문서의 근사 스트링 매칭 시스템은 사용자(100), 복수의 클라이언트(110 내지 180) 및 서버(190)를 포함한다.

<56> 그리고 복수의 클라이언트(110 내지 180) 각각은 인덱스 생성부(111), 송수신부(112) 및 트랩도어 생성부(113)를 포함하며, 서버(190)는 데이터 베이스부(191) 및 근사 스트링 매칭 테스트 수행부(192)를 포함할 수 있다.

<57> 인덱스 생성부(111)는 인증된 사용자(100)가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기 문서를 문서 암호용 키로 암호화하고, 상기 키워드를 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성한다.

<58> 그리고, 송수신부(112)는 상기 클라이언트(110 내지 180)에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버(190)에 전송한다.

<59> 또한, 트랩도어 생성부(113)는 상기 인증된 사용자(100)가 클라이언트(110 내지 180)에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의한다.

<60> 한편, 서버(190)의 데이터 베이스부(191)는 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장한다.

<61> 그리고, 근사 스트링 매칭 테스트 수행부(192)는 상기 클라이언트에서 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행한다.

<62> 여기서, 상기 트랩도어는 상기 인증된 사용자의 질의시마다 랜덤하게 생성되는 랜덤값을 이용하여 생성되며, 상기 트랩도어는 모듈 연산에 의해 생성될 수 있다.

<63> 한편, 상기 근사 스트링 매칭 테스트 수행부(192)는 생성된 패턴 스트링 P 와 i 번째 문서의 j 번째 인덱스

스트링 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 연산하고, 상기 연산된 해밍 디스턴스 값 $H(P, I_{ij})$ 이 미리 설정된 유사도 k 와 $H(P, I_{ij}) \leq k$ 의 관계가 성립하면, 상기 클라이언트에게 해

당 문서를 전송한다.

- <64> 이와 같이 실제로, 클라이언트(110 내지 180)는 제한된 저장 공간과 연산 능력을 가진 반면, 서버(190)는 무한한 저장 공간과 연산 능력을 가지고 있다. 여기서 클라이언트는 이런 서버의 능력을 이용하여 사용자(100)의 데이터를 서버에 저장하는데, 이때, 클라이언트는 서버가 사용자의 데이터의 내용을 알 수 없도록 사용자만의 키를 이용하여 암호화하여 서버에 저장한다. 그리고, 복호화 역시 서버에 저장해 놓은 암호화된 데이터를 가져와 사용자의 키를 이용하여 복호화한다. 클라이언트는 데이터를 생성하고, 데이터를 암호화 및 복호화를 수행하여 그 데이터를 관리할 수 있다. 이에 반해 서버는 단지 클라이언트로부터 전송받은 데이터를 저장하고, 클라이언트가 요청한 질의를 처리하여 그 결과값을 전송한다.
- <65> 본 발명에서는 암호화된 문서에 대한 근사 스트링 매칭 테스트를 수행하기 위하여 위하여 하기의 6가지의 알고리즘을 사용한다.
- <66> 첫 번째는, 시스템 파라미터 생성 알고리즘 $SysParam(1^k)$ 으로 인덱스의 암호화 시스템 구축을 위한 각종 파라미터 λ 를 생성한다.
- <67> 두 번째는, 키 생성 알고리즘 $KeyGen(\lambda)$ 로 인증된 사용자들의 인덱스 암호화에 쓰여질 인증된 사용자의 검색키 K 를 생성한다.
- <68> 세 번째는, 인덱스 생성 알고리즘 $IndGen(K, W, d, D)$ 으로, 인증된 사용자의 검색키 K , 문서의 키워드 w 들의 리스트 W , 문서 식별자(ID) d 및 필드 식별자 J 를 입력 받아 암호화하여 그 문서의 인덱스를 생성한다.
- <69> 네 번째는, 트랩도어 생성 알고리즘 $Trapdoor(K, w)$ 로 사용자가 검색하기를 원하는 키워드 w 를 클라이언트가 상기 사용자의 검색키 K 를 이용하여 암호화하여 트랩도어를 생성하는 알고리즘이다. 이것은 사용자가 원하는 문서를 서버가 탐색할 수 있도록 하기 위한 검색 능력을 부여하는 것이다.
- <70> 다섯 번째는, 패턴 생성 알고리즘 $PattGen(T, d)$ 로 클라이언트가 질의한 트랩도어 T 를 사용하여 문서 식별자(ID)가 d 인 문서의 스트링 매칭을 하기 위한 패턴 스트링을 만드는 알고리즘이다.
- <71> 여섯 번째는, 유사 스트링 매칭 테스트 알고리즘 $SimMatch(P, I)$ 으로, 생성한 패턴 스트링 P 와 각 문서의 인덱스 I 에 있는 특정 혹은 모든 키워드 필드의 스트링과 매칭 테스트를 수행한다. 이는 검증 요건인 유사도를 만족하면 1을 출력하고, 그렇지 않으면 0을 출력하게 된다.
- <72> 상술한 여섯 개의 알고리즘에 따라 본 발명의 암호화된 문서의 근사 스트링 매칭 방법을 상술하기로 한다.
- <73> 도 2는 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 흐름도를 도시한 것이다.
- <74> 각 문서에 대한 인덱스를 데이터베이스에 저장하는 형태로 키워드 필드(Keyword field)를 사용한다. 테이블은 m 개의 필드로 구성되며, 각 필드에는 각자의 속성이 부여된다.
- <75> 예를 들어, 메일 서버의 데이터베이스의 경우, 첫 번째 필드는 보낸 이, 두 번째 필드는 받는 이, 세 번째 필드는 날짜 등의 형태이다. 따라서, 추후에 검색어 질의를 위하여 클라이언트는 이 필드와 필드의 속성의 매칭 정보를 저장하고 있어야 한다.
- <76> 도 2를 참조하면, 우선, 인증된 사용자가 문서 및 상기 문서에 대한 키워드를 입력하면, 클라이언트에서 상기

문서를 문서 암호용 키로 암호화하고, 상기 키워드를 검색키로 철자별로 암호화하여 상기 문서에 대한 인덱스를 생성한다(210 과정).

<77> 그 다음, 상기 클라이언트에서 상기 암호화된 문서 및 상기 철자별로 암호화된 문서에 대한 인덱스를 서버에 전송하면, 상기 서버에서 상기 전송된 데이터베이스 테이블의 각 행에 상기 문서에 대한 인덱스를 저장한다(220 과정).

<78> 본 발명은 철자별 암호화의 문제점인 전수 조사 공격을 차단하기 위하여 데이터베이스의 각 셀마다 다른 암호 형태가 나오게 하는 셀 프라이버시(Cell Privacy)를 보장할 수 있는 암호화 방법을 사용한다. 이것은 각 셀에 들어갈 하나의 키워드를 암호화할 때, 사용자의 검색키, 문서의 문서 식별자(ID) 및 필드 식별자(ID) 모두를 함께 암호화함으로써 가능하다.

<79> 그러면, 첫 번째 행 첫 번째 열의 'a'를 암호화한 값과 첫 번째 행 두 번째 열의 'a', 그리고 두 번째 행 첫 번째 열의 'a'를 암호화한 값이 모두 다르게 되어 전수 조사 공격이 불가능하다.

<80> 그 다음, 상기 인증된 사용자가 클라이언트에게 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트에서 상기 전송된 키워드 및 상기 인증된 사용자의 검색키를 이용한 트랩도어를 생성하여 상기 서버에 질의한다(230 과정).

<81> 마지막으로, 상기 서버에서 상기 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링을 생성하고, 상기 생성된 패턴 스트링과 상기 인덱스의 인덱스 스트링을 이용하여 근사 스트링 매칭 테스트를 수행한다(240 과정).

<82> 이는, 패턴 스트링 P 와 각 문서의 j 번째 스트링 I_{ij} 를 가지고, 근사 스트링 매칭 테스트를 실시한다. 즉, 패턴 스트링 P 와 각 인덱스 I_{ij} 사이의 해밍 디스턴스 $H(P, I_{ij})$ 를 구하여, 그것이 미리 설정해 놓은 유사도 k 의 범주 내에 들면, 즉 $H(P, I_{ij}) \leq k$ 가 성립하면, "true"로 해당 문서를 리턴하고, 그렇지 않을 경우 "false"를 출력한다.

<83> 이와 같이 도 2의 본 발명의 암호화된 문서의 근사 스트링 매칭 방법에 따른 전체 데이터 흐름은 도 3에 도시된 바와 같다.

<84> 도 3를 참조하면, 각 문서마다 주요한 키워드를 뽑아 철자별로 암호화 방법을 적용하여 암호화하고, 이 암호화된 문서에 대한 인덱스를 생성하여 클라이언트가 생성된 인덱스를 서버에 전송하면 서버는 데이터베이스 테이블의 각 행에 하나의 문서에 대한 인덱스를 저장한다.

<85> 그런 후, 사용자가 클라이언트에게 사용자가 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트는 상기 전송된 키워드를 암호화하여 트랩도어를 생성하여 서버에게 전송한다. 서버는 트랩도어가 생성된 암호화된 키워드를 수신하여 미리 정해진 프로토콜에 의해 패턴 스트링을 생성하고, 생성된 패턴 스트링을 이용하여 근사 스트링 매칭 테스트를 수행하여, 상기 근사 스트링 매칭 테스트의 결과값이 미리 설정해 놓은 유사도 k 의 범주 내에 들면, 즉 $H(P, I_{ij}) \leq k$ 가 성립하면, "true"로 해당 문서를 클라이언트에게 리턴하고, 클라이언트는 이를 복호화하여 사용자에게 전송하게 된다.

<86> 본 발명에 따른 암호화된 문서의 근사 스트링 매칭 방법은 2가지로 구성될 수 있다.

<87> 우선, 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예는 하기와 같다.

<88> 도 4는 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예의 흐름도이다.

<89> 우선, 시스템 파라미터인 암호화 파라미터 $\lambda = (G, g, Gr, h(\cdot), HS(\cdot))$ 에 따라 암호화 시스템 구축을 위한 그룹 파라미터, 일방향 해쉬함수 및 슈도 랜덤 함수를 포함하는 암호화 함수를 생성한다(410 과정).

<90> 여기서 G 는 가장 큰 소수인 q 를 위수로 하는 그룹 파라미터이며, g 는 그룹 파라미터 G 의 생성 인자이다. Gr 은 슈도 랜덤 생성기이고, $h : \{0, 1\}^* \rightarrow Z_q$ 와 $HS : \{0, 1\}^* \rightarrow \{0, 1\}^k$ 는 일방향 해쉬 함수이다.

<91> 그 다음, 키 생성 알고리즘 $KeyGen(\lambda)$ 은 암호화 검색 시스템 사용을 위한 인증된 사용자의 검색키 $K \in \{0, 1\}^k$ 를 생성한다(420 과정).

<92> 그 다음, 본 발명에 따른 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예에서 클라이언트는 사용자의 검색키 k_u , 문서의 키워드들의 리스트 $W_i = \{W_{i,1}, W_{i,2}, \dots, W_{i,m}\}$ 을 이용하여 문서 아이디 d_i , 필드 아이디 집합 $J = \{j_1, j_2, \dots, j_m\}$ 와 함께 암호화하여 상기 문서의 인덱스를 생성한다(430 과정).

<93> 여기서, W_{ij} 는 i 번째 문서의 j 번째 필드의 키워드로서 $W_{ij} = w_{ij}^1, w_{ij}^2, \dots, w_{ij}^s$ 이며, w_{ij}^s 는 W_{ij} 의 s 번째 철자이다.

<94> 인덱스를 생성하는 상세한 과정은 하기와 같다.

<95> 첫 번째로, 키워드 W_{ij} 의 제 1 차 암호화 과정으로 키워드와 필드 아이디를 사용자의 검색키로 철자별로 암호화하여 하기의 수학적 식 1에 의해 제 1 암호화 값 $F_{k_u}(W_{1,j})$ 를 생성한다.

수학적 식 1

$$\begin{aligned} F_{k_u}(W_{1,1}) &= f_{k_u}(w_{1,1}^1|1) \| f_{k_u}(w_{1,1}^2|1) \| \dots \| f_{k_u}(w_{1,1}^s|1) \\ F_{k_u}(W_{1,2}) &= f_{k_u}(w_{1,2}^1|2) \| f_{k_u}(w_{1,2}^2|2) \| \dots \| f_{k_u}(w_{1,2}^s|2) \\ &\dots \\ F_{k_u}(W_{1,j}) &= f_{k_u}(w_{1,j}^1|j) \| f_{k_u}(w_{1,j}^2|j) \| \dots \| f_{k_u}(w_{1,j}^s|j) \end{aligned}$$

<96> 두 번째로, 상기 수학적 식 1에 따른 제 1 암호화 값을 이용하여 철자별로 문서 식별자 d_i 를 추가하여 하기의 수학적 식 2에 따라 제 2 암호화 값 I_{ij} 을 생성한다.

수학적 식 2

$$\begin{aligned} I_{i,j} &= HS \left(g^{d_i h(k_u) F_{k_u}(w_{i,j}^1|j)} \right) \| HS \left(g^{d_i h(k_u) F_{k_u}(w_{i,j}^2|j)} \right) \\ &\| \dots \| HS \left(g^{d_i h(k_u) F_{k_u}(w_{i,j}^s|j)} \right) \end{aligned}$$

<98>

<99> 세 번째로, 인덱스 I_i 를 생성하는데, 일종의 문서 식별자 역할을 하는 $id_{1,1}$ 과 $id_{1,2}$ 와 상기 수학식 2에서 생성된 $I_{i,1}$ 내지 $I_{i,m}$ 까지를 붙여서 하기의 수학식 3에 따라 인덱스를 생성한다.

수학식 3

$$I_i = id_{i,1}, id_{i,2}, I_{i,1}, I_{i,2} \dots, I_{i,m}$$

$$where, id_{i,1} = g^{d_i h(k_u)}, id_{i,2} = g^{-d_i}$$

<101> 그 다음, 사용자가 검색하기를 원하는 키워드 $W = w^1 w^2 \dots w^s$ 를 사용자의 검색키를 가지고 클라이언트는 하기의 수학식 4에 따라 트랩도어(trapdoor)를 생성한다(440 과정).

<102> 트랩도어는 사용자가 검색하기를 원하는 키워드를 암호화함으로써 서버는 사용자가 어떠한 검색어를 질의했는지, 또 검색 결과에 해당하는 문서가 어떠한 내용인지 전혀 추측 불가능하게 하는 것으로, 서버가 데이터베이스에 저장된 데이터를 복호화하지 않은 채로 주어진 프로토콜대로 검색 가능하게 한다.

수학식 4

$$T = (T_1, T_2, T_3) \\ = ((f_{k_{i,j}}(w^1 | j) + \alpha) \parallel \dots \parallel (f_{k_{i,j}}(w^s | j) + \alpha), \\ h(f_{k_{i,j}})\alpha, j(\text{키워드 식별자})) \pmod{q}$$

$$where, T_1 = (f_{k_{i,j}}(w^1 | j) + \alpha) \parallel \dots \parallel (f_{k_{i,j}}(w^s | j) + \alpha) \\ = t^1 \parallel \dots \parallel t^s$$

<104> 그리고, 상기 수학식 4에서 α 는 매 쿼리(query)마다 새롭게 랜덤 하게 생성 되는 값이고, 트랩도어는 모듈 연산으로 계산된다.

<105> 따라서, 서버는 비록 같은 사용자가 같은 키워드를 검색하였다고 하더라도 서버는 그 사실을 알 수 없다. 이것을 쿼리 프라이버시라고 하며, 이것으로 인하여 서버는 축적된 결과들로부터 연관성을 전혀 찾을수 없게 되어 어떠한 사실도 알아낼 수 없는 것이다.

<106> 마지막으로, 상기 서버에서 상기 생성된 트랩도어를 수신하여 미리 정해진 프로토콜에 의하여 패턴 스트링 P 을 생성하고, 생성된 패턴 스트링 P 와 각 문서의 j 번째 스트링 $I_{i,j}$ 를 가지고, 근사 스트링 매칭 테스트를 실시한다(350 과정).

<107> 즉, 패턴 스트링 P 와 각 인덱스 $I_{i,j}$ 사이의 해밍 디스턴스 $H(P, I_{i,j})$ 를 구하여, 그것이 미리 설정해 놓은 유사도 k 의 범주 내에 들면, 즉 $H(P, I_{i,j}) \leq k$ 가 성립하면, "true"로 해당 문서를 리턴하고, 그렇지 않을 경우 "false"를 출력한다.

<108> 이와 같이, 도 4의 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예에 따른 데이터 흐름은 도 5

에 도시된 바와 같다.

<109> 도 5에 따라, 각 문서마다 주요한 키워드를 뽑아 철자별로 암호화 방법을 적용하여 암호화하고, 이 암호화된 문서에 대한 인덱스를 생성하여 클라이언트가 생성된 인덱스를 서버에 전송하면 서버는 데이터베이스 테이블의 각 행에 하나의 문서에 대한 인덱스를 저장한다.

<110> 그런 후, 사용자가 클라이언트에게 사용자가 검색하고자 하는 문서에 대한 키워드를 전송하면, 클라이언트는 상기 전송된 키워드를 암호화하여 하나의 트랩도어를 생성하여 서버에게 전송한다. 서버는 트랩도어가 생성된 암호화된 키워드를 수신하여 미리 정해진 프로토콜에 의해 패턴 스트링을 생성하고, 생성된 패턴 스트링을 이용하여 근사 스트링 매칭 테스트를 수행한다.

<111> 한편, 도 6은 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예의 흐름도를 도시한 것이다.

<112> 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예에서는 하나의 키워드에 대하여 트랩도어는 모든 필드 즉, m 개의 필드에 대해 생성하여야 한다.

<113> 우선, 클라이언트는 사용자의 검색키 k_u , 문서의 키워드 W_i 들의 리스트의 집합인 $W_i = \{W_{i,1}, W_{i,2}, \dots, W_{i,m}\}$ 를 가지고 문서 식별자 d_i , 필드 아이디 집합 $J = \{j_1, j_2, \dots, j_m\}$ 과 함께 암호화하여 그 문서의 인덱스를 생성한다(610 과정).

<114> 여기서, W_{ij} 는 i 번째 문서의 j 번째 필드의 키워드로서 $W_{ij} = w_{ij}^1, w_{ij}^2, \dots, w_{ij}^s$ 이며, w_{ij}^s 는 W_{ij} 의 s 번째 철자이다. 인덱스를 생성하는 상세한 과정은 하기와 같다.

<115> 첫 번째로, 키워드 W_{ij} 의 제 1 차 암호화 값 F_{k_u} 을 연산하는 과정으로, 키워드를 필드 아이디와 키워드 스트링의 철자의 순서번호도 같이 사용자의 검색키로 철자별로 암호화하여 붙여(concatenation) 놓는다. 이는 하기의 수학적식 5와 같다.

수학적식 5

$$\begin{aligned} F_{k_u}(W_{1,1}) &= f_{k_u}(w_{1,1}^1 | 1 + 1) \| f_{k_u}(w_{1,1}^2 | 2 + 1) \| \dots \\ &\quad \dots \| f_{k_u}(w_{1,1}^s | s + 1) \\ &= f_{1,1}^1 \| f_{1,1}^2 \| \dots \| f_{1,1}^s \end{aligned}$$

$$\begin{aligned} F_{k_u}(W_{1,2}) &= f_{k_u}(w_{1,2}^1 | 1 + 2) \| f_{k_u}(w_{1,2}^2 | 2 + 2) \| \dots \\ &\quad \dots \| f_{k_u}(w_{1,2}^s | s + 2) \\ &= f_{1,2}^1 \| f_{1,2}^2 \| \dots \| f_{1,2}^s \end{aligned}$$

<118> ...

$$\begin{aligned} F_{k_u}(W_{1,j}) &= f_{k_u}(w_{1,j}^1 | 1+j) \| f_{k_u}(w_{1,j}^2 | 2+j) \| \dots \\ &\quad \dots \| f_{k_u}(w_{1,j}^s | s+j) \\ &= f_{1,j}^1 \| f_{1,j}^2 \| \dots \| f_{1,j}^s \end{aligned}$$

<119>

<120> 상기 수학식 5에 따라 연산된 키워드 W_{ij} 의 1차 암호화 값 F_{k_u} 에 문서 아이디 d_i 를 추가하여 하기의 수학식 6에 의거하여 제 2차 암호화 값 I_{ij} 를 연산한다.

수학식 6

$$I_{i,j} = HS(f_{i,j}^1 | d_i) \| HS(f_{i,j}^2 | d_i) \| \dots \| HS(f_{i,j}^s | d_i)$$

<121>

<122> 그 다음, 문서 아이디 d_i 와 상기 수학식 6에서 생성한 $I_{i,1} \sim I_{i,m}$ 까지를 붙인다.

<123>

그러면, 사용자가 검색하기를 원하는 키워드 $W = w^1 w^2 \dots w^s$ 를 사용자의 검색키를 가지고 클라이언트는 하기의 수학식 7과 같은 트랩도어를 생성한다(620 과정).

수학식 7

$$\begin{aligned} T &= (T_1, T_2, \dots, T_m) \\ T_1 &= f_{k_u}(w^1 | 1+1) \| f_{k_u}(w^2 | 2+1) \| \dots \\ &\quad \dots \| f_{k_u}(w^s | s+1) = t_1^1 \| t_1^2 \| \dots \| t_1^s \\ T_2 &= f_{k_u}(w^1 | 1+2) \| f_{k_u}(w^2 | 2+2) \| \dots \\ &\quad \dots \| f_{k_u}(w^s | s+2) = t_2^1 \| t_2^2 \| \dots \| t_2^s \\ &\dots \\ T_m &= f_{k_u}(w^1 | 1+m) \| f_{k_u}(w^2 | 2+m) \| \dots \\ &\quad \dots \| f_{k_u}(w^s | s+m) = t_m^1 \| t_m^2 \| \dots \| t_m^s \end{aligned}$$

<124>

<125> 그 다음, 상기 수학식 7에 의해서 생성된 m 개의 트랩도어와 각 문서의 식별자를 가지고, 서버는 각 문서의 스트링 매칭 테스트를 위한 패턴 스트링을 하기의 수학식 8에 의거하여 m 개 생성한다(630 과정).

수학식 8

$$T_1: HS(t_1^1 | d_i) \parallel HS(t_1^2 | d_i) \parallel \dots \parallel HS(t_1^s | d_i) = P_1$$

$$T_2: HS(t_2^1 | d_i) \parallel HS(t_2^2 | d_i) \parallel \dots \parallel HS(t_2^s | d_i) = P_2$$

.....

$$T_m: HS(t_m^1 | d_i) \parallel HS(t_m^2 | d_i) \parallel \dots \parallel HS(t_m^s | d_i) = P_m$$

<126>

<127> 그 다음, 패턴 스트링 $P = (P_1, P_2, \dots, P_m)$ 과 각 문서의 m 개의 인덱스 스트링 $I_{i,j}$ 를 가지고 근사 스트링 매칭 테스트를 실시한다(640 과정).

<128> 즉, 패턴 스트링 $P = (P_1, P_2, \dots, P_m)$ 과 각 인덱스 $I_{i,j}$ 사이의 해밍 디스턴스 $H(P_1, I_{i,1}), H(P_2, I_{i,2}), \dots, H(P_m, I_{i,m})$ 를 연산하여 그것이 미리 설정해 놓은 유사도 k 의 범주 내에 들면 'true'로 해당 문서를 리턴하고, 그렇지 않을 경우에는 'false'가 된다.

<129> 상기 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예의 구체적인 동작 과정은 도 7에 도시된 바와 같다.

<130> 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예의 구체적인 동작 과정은 도 5와 유사하지만, 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예에서는 사용자가 질의하는 키워드가 어느 필드에 속한 것 인지를 알 수 없기 때문에 하나의 키워드에 대하여 트랩도어는 모든 필드 즉, m 개의 필드에 대하여 모두 생성하여야 한다는 점에서 차별이 있다.

<131> 본 발명은 소프트웨어를 통해 실행될 수 있다. 소프트웨어로 실행될 때, 본 발명의 구성 수단들은 필요한 작업을 실행하는 코드 세그먼트들이다. 프로그램 또는 코드 세그먼트들은 프로세서 판독 가능 매체에 저장되거나 전송 매체 또는 통신망에서 반송파와 결합된 컴퓨터 데이터 신호에 의하여 전송될 수 있다.

<132> 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, DVD±ROM, DVD-RAM, 자기 테이프, 플로피 디스크, 하드 디스크(hard disk), 광데이터 저장장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 장치에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

<133> 본 발명은 도면에 도시된 일 실시예를 참고로 하여 설명하였으나 이는 예시적인 것에 불과하며 당해 분야에서 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 실시예의 변형이 가능하다는 점을 이해할 것이다. 그러나, 이와 같은 변형은 본 발명의 기술적 보호범위내에 있다고 보아야 한다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해서 정해져야 할 것이다.

도면의 간단한 설명

<134> 도 1은 본 발명에 따른 암호화된 문서의 근사 스트링 매칭 시스템의 블록도이다.

<135> 도 2는 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 흐름도를 도시한 것이다.

<136> 도 3은 도 2의 본 발명의 암호화된 문서의 근사 스트링 매칭 방법에 따른 전체 데이터 흐름을 도시한 것이다.

<137> 도 4는 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예의 흐름도이다.

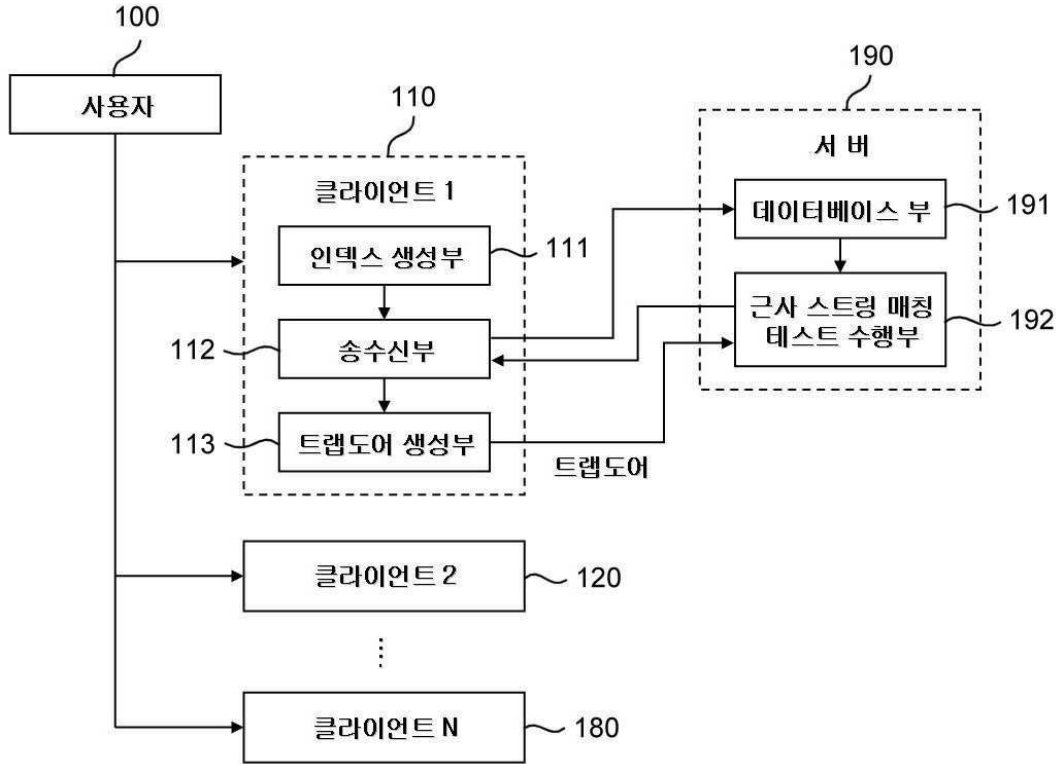
<138> 도 5는 도 4의 암호화된 문서의 근사 스트링 매칭 방법의 일 실시예에 따른 데이터 흐름을 도시한 것이다.

<139> 도 6은 본 발명의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예의 흐름도를 도시한 것이다.

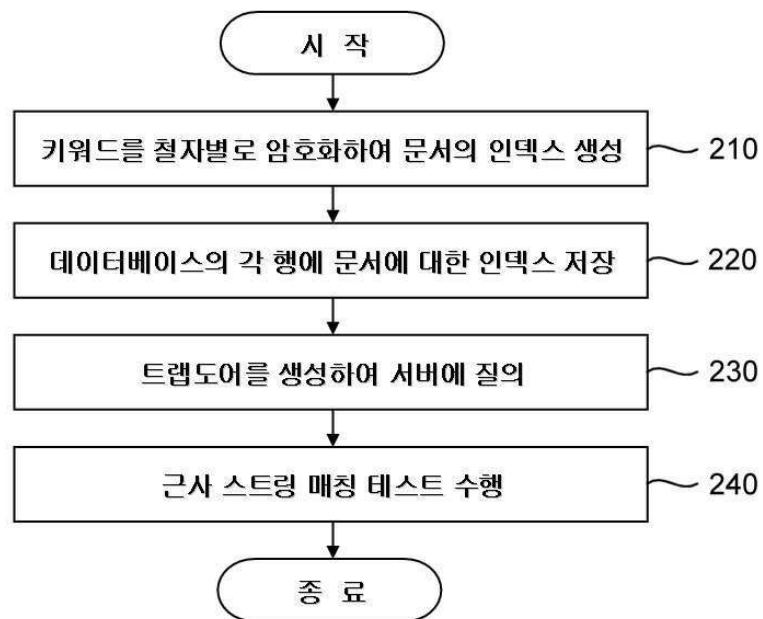
<140> 도 7은 도 6의 암호화된 문서의 근사 스트링 매칭 방법의 다른 실시예에 따른 데이터 흐름을 도시한 것이다.

도면

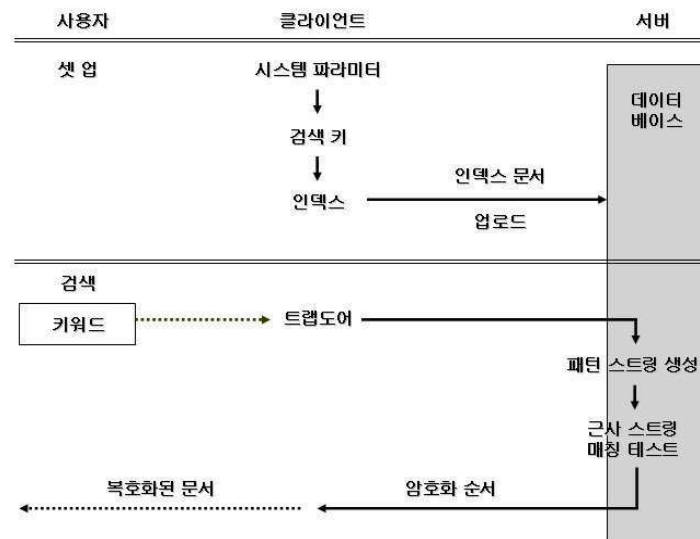
도면1



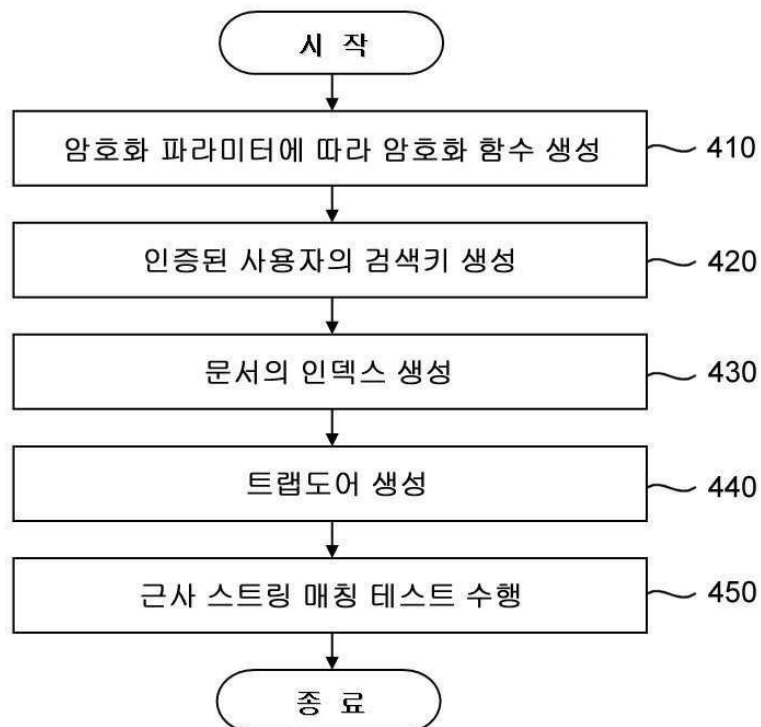
도면2



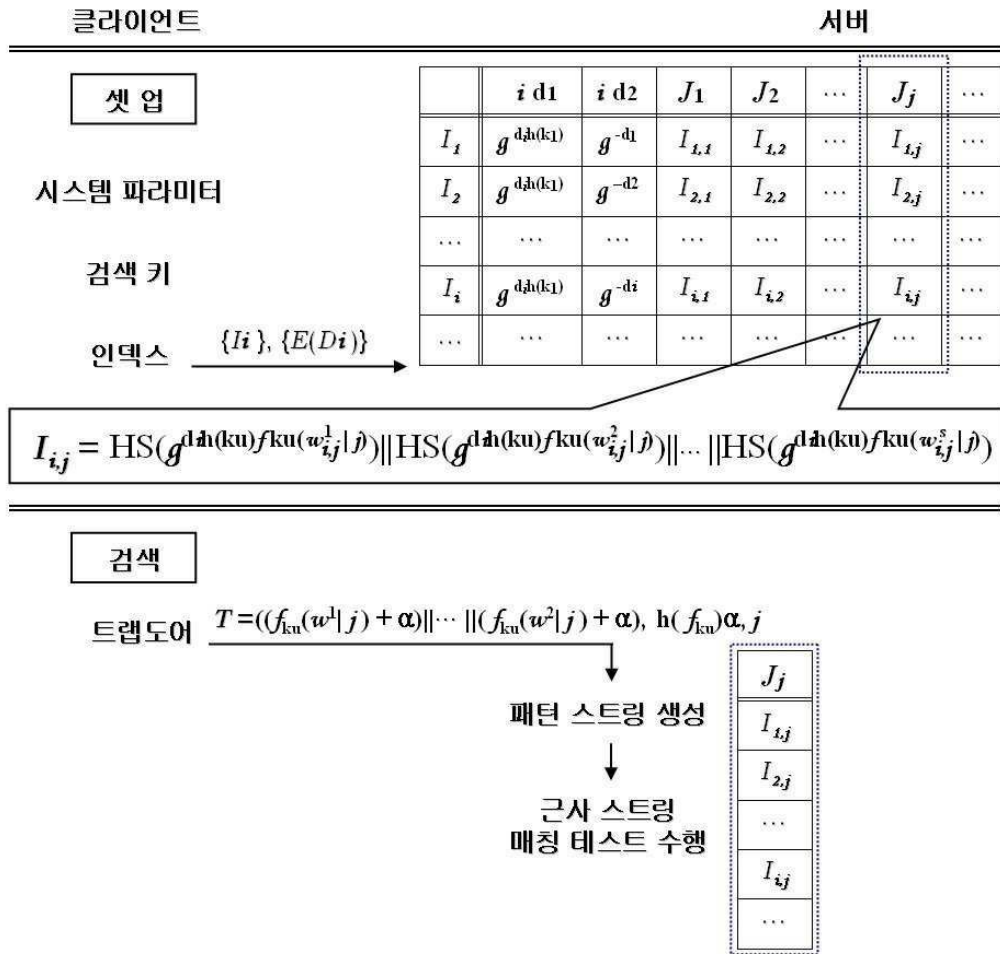
도면3



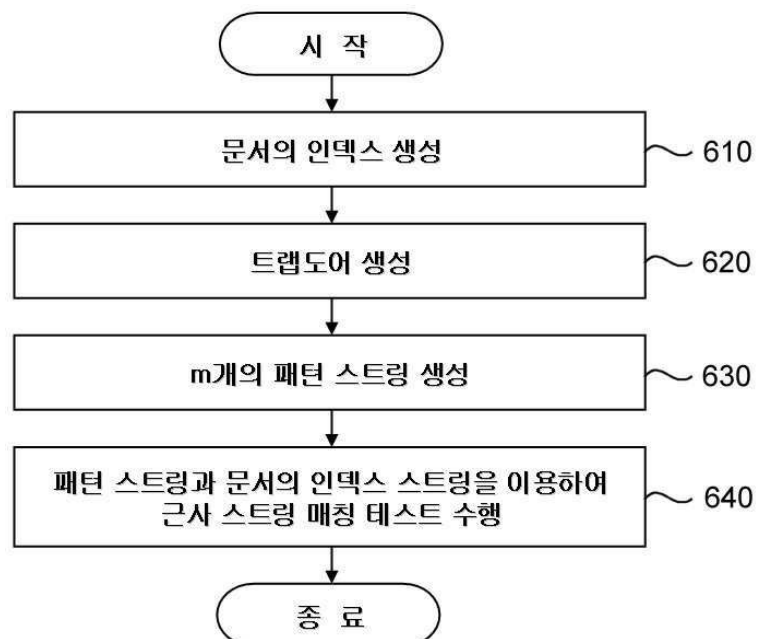
도면4



도면5



도면6



도면7

