(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0067601
(43) 공개일자 2011년06월22일

(51) Int. Cl.
H04L 9/14 (2006.01) *G06F 21/24* (2006.01)
G06F 12/14 (2006.01) *G06F 9/06* (2006.01)
(21) 출원번호 10-2009-0124256
(22) 출원일자 2009년12월14일
심사청구일자 2009년12월14일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가 1
(72) 발명자
이동훈
서울특별시 종로구 내수동 72 경희궁의아침3단지 아파트 1404호
정익래
서울특별시 노원구 공릉동 81번지 태강아파트 1014동 1303호
노건태
서울특별시 강남구 개포3동 주공아파트 707동 1304호
(74) 대리인
특허법인충현

전체 청구항 수 : 총 11 항

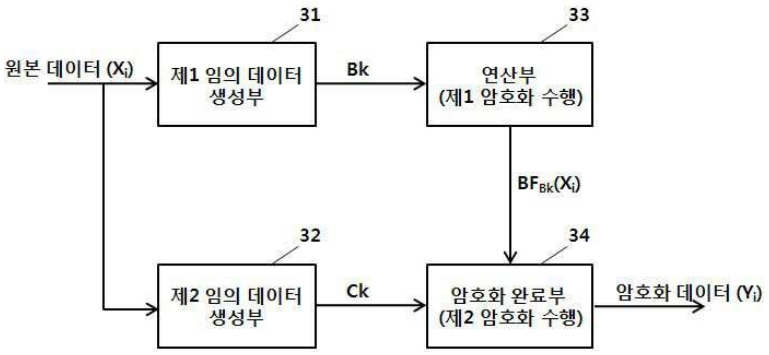
(54) 이진 데이터의 암호화 시스템 및 그 방법

(57) 요 약

본 발명은 이진 데이터의 암호화 방법에 관한 것으로, 보다 상세하게는 암호화 대상 이진 데이터(binary data)의 특성을 보존할 수 있는 암호화 방안에 관한 것이며, 특히 이진 데이터들 사이의 거리를 보존할 수 있는 거리 보존 암호화(distance preserving encryption) 방안에 관한 것이다.

본 명세서에서 개시하는 암호화 시스템은 n-비트(n-bit) 데이터의 암호화를 위한 제1 임의 데이터를 생성하는 제1 임의 데이터 생성부; 상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터를 다차원 벡터로 생성하는 제2 임의 데이터 생성부; 상기 n-비트 데이터와 상기 제1 임의 데이터를 비트 대 비트 연산(bit by bit operation)하는 연산부; 및 상기 비트 대 비트 연산의 결과와 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 완료하는 암호화 완료부를 포함하여 상기한 과제를 해결한다.

대 표 도 - 도3a



이 발명을 지원한 국가연구개발사업
과제고유번호 ITAA1100090100480001000100100
부처명 정보통신연구진흥원
연구관리전문기관
연구사업명 지식경제기술혁신사업
연구과제명 Car-헬스케어 보안 기술개발
기여율
주관기관 고려대학교
연구기간 2009년 03월 01일 ~ 2010년 02월 28일

특허청구의 범위

청구항 1

n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 생성하는 임의 데이터 생성부; 및

상기 n-비트 데이터와 상기 임의 데이터를 비트 대 비트 연산(bit by bit operation)하여 상기 암호화를 수행하는 암호화 수행부를 포함하는 이진 데이터의 암호화 시스템.

청구항 2

n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 다차원 벡터로 생성하는 임의 데이터 생성부; 및

상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 수행하는 암호화 수행부를 포함하는 이진 데이터의 암호화 시스템.

청구항 3

n-비트(n-bit) 데이터의 암호화를 위한 제1 임의 데이터를 생성하는 제1 임의 데이터 생성부;

상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터를 다차원 벡터로 생성하는 제2 임의 데이터 생성부;

상기 n-비트 데이터와 상기 제1 임의 데이터를 비트 대 비트 연산(bit by bit operation)하는 연산부; 및

상기 비트 대 비트 연산의 결과와 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 완료하는 암호화 완료부를 포함하는 이진 데이터의 암호화 시스템.

청구항 4

제 1 항 또는 제 3 항에 있어서,

상기 비트 대 비트 연산은 XOR(eXclusive-OR)인 것을 특징으로 하는 이진 데이터의 암호화 시스템.

청구항 5

제 2 항 또는 제 3 항에 있어서,

상기 다차원 벡터는 n차 치환 행렬(n-th order permutation matrix)인 것을 특징으로 하는 이진 데이터의 암호화 시스템.

청구항 6

(a)n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 생성하는 단계; 및

(b)상기 n-비트 데이터와 상기 임의 데이터를 비트 대 비트 연산(bit by bit operation)하여 상기 암호화를 수행하는 단계를 포함하는 이진 데이터의 암호화 방법.

청구항 7

(a)n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 다차원 벡터로 생성하는 단계; 및

(b)상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 수행하는 단계를 포함하는 이진 데이터의 암호화 방법.

청구항 8

(a)n-비트(n-bit) 데이터의 암호화를 위한 제1 임의 데이터를 생성하는 단계;

(b)상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터를 다차원 벡터로 생성하는 단계;

(c)상기 n-비트 데이터와 상기 제1 임의 데이터를 비트 대 비트 연산(bit by bit operation)하는 단계; 및

(d)상기 비트 대 비트 연산의 결과와 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 완료하는 단계를 포함하는 이진 데이터의 암호화 방법.

청구항 9

제 6 항 또는 제 8 항에 있어서,

상기 비트 대 비트 연산은 XOR(eXclusive-OR)인 것을 특징으로 하는 이진 데이터의 암호화 방법.

청구항 10

제 7 항 또는 제 8 항에 있어서,

상기 다차원 벡터는 n차 치환 행렬(n-th order permutation matrix)인 것을 특징으로 하는 이진 데이터의 암호화 방법.

청구항 11

제 6 항 내지 제 8 항 중 어느 한 항의 방법을 컴퓨터에서 실행하기 위한 프로그램을 기록하는 컴퓨터 판독 가능한 기록 매체.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 이진 데이터의 암호화 시스템 및 그 방법에 관한 것으로, 보다 상세하게는 암호화 대상 이진 데이터(binary data)의 특성을 보존할 수 있는 암호화 방안에 관한 것이며, 특히 이진 데이터들 사이의 거리를 보존할 수 있는 거리 보존 암호화(distance preserving encryption) 방안에 관한 것이다.

배 경 기 술

[0002] 원본 데이터(raw data)가 지닌 특성을 보존하면서 암호화하는 방안에는 여러 가지가 있다. 그 중에서도 원본 데이터들 사이의 거리를 보존하면서 암호화하는 거리 보존 암호화에 대한 연구들이 활발히 진행 중인데, 거리 보존 암호화는 프라이버시 보존 클러스터링(privacy preserving clustering) 분야에서 데이터의 프라이버시를 노출하지 않으면서 효과적으로 클러스터링을 할 수 있는 유용한 방안으로 주로 사용된다. 하지만, 기존의 거리 보존 암호화 기법들에서는 유클리디안 거리(Euclidean distance)를 보존하는 방안에 대해서만 개시하고 있으며, 유클리디안 거리 아닌 다른 거리를 보존하는 암호화 방안은 실시되거나 제안된 바가 없다. 유클리디안 거리만을 고려할 경우 발생되는 제반 문제 및 거리 보존 암호화의 활용 범위의 협소성을 고려해 볼 때, 다른 거리를 보존하는 암호화 내지는 복호화 방안이 필요하다.

발명의 내용

해결 하고자하는 과제

[0003] 본 발명은 상기와 같은 필요성에 부응하기 위해 창안된 것으로, 본 발명이 해결하고자 하는 과제는 기존 거리 보존 암호화 방안의 문제점을 극복하고 아울러 거리 보존 암호화 방안의 활용성을 배가시킬 수 있는 새로운 거리 보존 암호화 방안을 제안하는 것이다.

과제 해결수단

[0004] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 암호화 시스템은

[0005] n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 생성하는 임의 데이터 생성부; 및 상기 n-비트 데이터와 상기 임의 데이터를 비트 대 비트 연산(bit by bit operation)하여 상기 암호화를 수행하는 암호화 수행부를 포함하여 상기한 과제를 해결한다.

[0006] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 또 다른 암호화 시스템은

[0007] n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 다차원 벡터로 생성하는 임의 데이터 생성부; 및 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 수행하는 암호화 수행부를 포함하여 상기한 과제를 해결

한다.

[0008] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 또 다른 암호화 시스템은

[0009] n-비트(n-bit) 데이터의 암호화를 위한 제1 임의 데이터를 생성하는 제1 임의 데이터 생성부; 상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터를 다차원 벡터로 생성하는 제2 임의 데이터 생성부; 상기 n-비트 데이터와 상기 제1 임의 데이터를 비트 대 비트 연산(bit by bit operation)하는 연산부; 및 상기 비트 대 비트 연산의 결과와 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 완료하는 암호화 완료부를 포함하여 상기한 과제를 해결한다.

[0010] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 암호화 방법은

[0011] (a)n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 생성하는 단계; 및 (b)상기 n-비트 데이터와 상기 임의 데이터를 비트 대 비트 연산(bit by bit operation)하여 상기 암호화를 수행하는 단계를 포함하여 상기한 과제를 해결한다.

[0012] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 또 다른 암호화 방법은

[0013] (a)n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 다차원 벡터로 생성하는 단계; 및 (b)상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 수행하는 단계를 포함하여 상기한 과제를 해결한다.

[0014] 상기와 같은 과제를 해결하기 위해 본 명세서에서 개시하는 또 다른 암호화 방법은

[0015] (a)n-비트(n-bit) 데이터의 암호화를 위한 제1 임의 데이터를 생성하는 단계; (b)상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터를 다차원 벡터로 생성하는 단계; (c)상기 n-비트 데이터와 상기 제1 임의 데이터를 비트 대 비트 연산(bit by bit operation)하는 단계; 및 (d)상기 비트 대 비트 연산의 결과와 상기 다차원 벡터를 이용하여 상기 n-비트 데이터의 암호화를 완료하는 단계를 포함하여 상기한 과제를 해결한다.

효 과

[0016] 본 발명에 의한 거리 보존 암호화 기법에 의하면 기존의 거리 보존 암호화 기법과는 달리 해밍 거리(Hamming distance) 또한 보존할 수 있다. 따라서 기존의 거리 보존 암호화 방안의 활용 범위가 매우 협소한 문제점을 해결할 수 있다. 아울러 암호화에 요구되는 연산이 매우 간단하고 연산량이 현저히 줄어들며, 암호화 데이터에 수반되었던 길이 확장(length expansion)의 발생 염려가 없어 보다 효율적이고 구현이 용이한 암호화 시스템의 제작을 가능하게 한다.

[0017] 본 발명에 의한 거리 보존 암호화 기법에 의하면 기존의 암호화 기법의 적용이 가능하지 아니한 새로운 응용이 가능하다. 예를 들어 포털 사이트 사용자가 키워드 검색을 수행할 때 자신이 검색하는 키워드의 프라이버시를 노출하고 싶지 않은 경우, 본 발명에 의한 암호화 기법을 사용한다면 키워드에 대한 프라이버시의 노출 없이도 유사 키워드 검색이 가능하게 된다. 또한, 의료 서비스를 제공하는 곳에서 환자의 민감한 유전자에 대해 변형 여부를 판단하는 경우, 환자의 유전자에 대한 프라이버시의 노출 없이도 유전자 유사 변형 여부 판단이 가능하다.

발명의 실시를 위한 구체적인 내용

[0018] 본 발명에 관한 구체적인 내용의 설명에 앞서 이해의 편의를 위해 본 발명이 해결하고자 하는 과제의 해결 방안의 개요를 우선 제시한다.

[0019] 위에서 언급한 바와 같이, 이제까지 제안되거나 실시된 거리 보존 암호화 방안은 유클리디안 거리만을 고려한 방안이었다. 하지만 유클리디안 거리만을 고려하는 기존 내지는 현재의 거리 보존 암호화 방안은 거리 보존 제약 때문에 확률론적 암호화 기법을 사용할 수 없으며, 결과적으로 암호화 기법에 일반적으로 적용되는 안전성 모델을 그대로 사용할 수 없다는 단점이 존재한다. 아울러 거리 보존 암호화 방안의 활용 범위가 매우 협소한 문제점 및 암호화에 따른 길이 확장(length expantion)이 수반되어 암호화에 대한 부하가 커지는 문제점도 갖는다.

[0020] 따라서 본 발명은 상기한 거리 보존 암호화 방안의 제반 문제를 해결하기 위해 이진 데이터에 대한 개선된 거리 보존 암호화 방안을 제공하는데, 이는 해밍 거리(Hamming distance)를 보존하는 것으로서 개선된 거리 보존 암호화 방안을 제공한다.

[0021] 이하, 본 발명이 해결하고자 하는 과제의 해결 방안을 명확하게 하기 위한 발명의 구성을 본 발명의 바람직한 실시예에 근거하여 첨부 도면을 참조하여 상세히 설명하되, 도면의 구성요소들에 참조번호를 부여함에 있어서 동일 구성요소에 대해서는 비록 다른 도면상에 있더라도 동일 참조번호를 부여하였으며 당해 도면에 대한 설명시 필요한 경우 다른 도면의 구성요소를 인용할 수 있음을 미리 밝혀둔다. 아울러 본 발명과 관련된 공지 기능 혹은 구성에 대한 구체적인 설명 그리고 그 이외의 제반 사항이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그 상세한 설명을 생략한다.

[0022] 도 1a는 본 시스템 발명의 바람직한 제1 구성을 제시한 것이며, 도 1b는 본 방법 발명의 바람직한 제1 흐름을 제시한 것이다.

[0023] 임의 데이터 생성부(11)는 암호화 대상인 n-비트(n-bit) 원본 데이터(raw data)의 암호화를 위한 임의 데이터를 생성시킨다(s11). 구체적으로는 다음과 같다.

[0024] 암호화 대상 n-비트 데이터 X_i를 행 벡터(row vector)로 표현한다.

[0025] $X_i = (X_{i,1}, X_{i,2}, \cdots, X_{i,n})$.

[0026] 임의 데이터 Bk는 다음과 같이 n-비트 행 벡터 형태로 생성될 수 있다.

[0027] $Bk = (Bk_1, Bk_2, \cdots, Bk_n)$.

[0028] 암호화 수행부(12)는 상기 X_i를 암호화한 암호화 데이터 Y_i를 생성하는데(s12), 암호화(Y_i의 생성)는 X_i와 Bk를 비트 대 비트 연산(bit by bit operation)을 통해 이루어진다. 이때 비트 대 비트 연산은 배타적 논리합(XOR: eXclusiveOR) 연산을 통해 이루어진다.

[0029] $Y_i = BF_{Bk}(X_i) = X_i \oplus Bk = (X_{i,1} \oplus Bk_1, X_{i,2} \oplus Bk_2, \cdots, X_{i,n} \oplus Bk_n) = (Y_{i,1}, Y_{i,2}, \cdots, Y_{i,n})$.

[0030] 여기서 $BF_{Bk}(X_i)$는 Bk를 인자로 하여 원본 데이터(raw data) X_i의 암호화 연산을 나타내는 기호이며, $\oplus$는 XOR 연산 기호이다. 그리고 이 암호화 연산의 의미는 Bk_j(jn)의 비트값이 0인 경우에는 원본 데이터의 j번째 비트값을 그대로 두는 것이고, Bk_j의 비트값이 1인 경우에는 원본 데이터의 j번째 비트값이 0이면 1로, 1이면 0으로 바꾸어 암호화를 하는 것이다.

[0031] 한편 이렇게 암호화된 데이터 Y_i의 복호화는 암호화 연산의 역 연산(inverse operation)으로 이루어질 수 있다. 역 연산(복호화)의 경우에도 상기 암호화를 위한 임의 데이터를 그대로 이용할 수 있다.

[0032] $X_i = BF_{Bk}^{-1}(Y_i) = Y_i \oplus Bk = (Y_{i,1} \oplus Bk_1, Y_{i,2} \oplus Bk_2, \cdots, Y_{i,n} \oplus Bk_n) = (X_{i,1}, X_{i,2}, \cdots, X_{i,n})$.

[0033] 도 1의 경우에 의한 암호화 기법의 적용 실제를 예를 들어 살펴본다.

[0034] $X_i = (1,0,1)$이고 $Bk = (1,1,0)$이라 하자.

[0035] $$Y_i = BF_{Bk}(X_i) = X_i \oplus Bk = (101) \oplus (110) = (011)$$.

[0036] 따라서 원본 데이터의 각 비트값이 유지되거나 바뀌어 X_i에 대한 암호화가 이루어짐을 알 수 있다.

[0037] 복호화는 $$X_i = BF_{Bk}^{-1}(Y_i) = Y_i \oplus Bk = (011) \oplus (110) = (101)$$.

[0038] 따라서 원본 데이터 X_i가 완벽하게 복호화됨을 알 수 있다.

[0039] 도 1에 의한 암호화 기법을 사용하면 원본 데이터들 사이의 비트 차이, 즉 해밍 거리(hamming distance)를 보존할 수 있음을 알 수 있다. 만약 Bk의 j번째 비트가 0이라면 원본 데이터의 값은 변경되지 않기 때문에 원본 데이터들 사이의 해밍 거리는 보존되며, Bk의 j번째 비트가 1이라면 원본 데이터들의 j번째 비트가 모두 변경되기 때문에 여전히 데이터들 사이의 해밍 거리는 그대로 보존된다. 그리고 데이터들이 모두 비트 문자열이기 때문에 해밍 거리가 보존되는 것은 유클리디안 거리도 보존되는 것을 의미한다. 아울러 본 발명에 의한 암호화 기법은

암호화/복호화 시에 XOR(eXclusive OR) 연산으로만 이루어지기 때문에 암호화/복호화 하드웨어 또는 소프트웨어의 구현에 매우 효율적이다.

[0040] 도 2a는 본 시스템 발명의 바람직한 제2 구성을 제시한 것이며, 도 2b는 본 방법 발명의 바람직한 제2 흐름을 제시한 것이다.

[0041] 임의 데이터 생성부(21)는 암호화 대상인 n-비트(n-bit) 데이터의 암호화를 위한 임의 데이터를 생성한다(s21). 구체적으로는 다음과 같다.

[0042] 암호화 대상 n-비트 데이터 X_i를 행 벡터(row vector)로 표현한다.

[0043] $X_i = (X_{i,1}, X_{i,2}, \cdots, X_{i,n})$.

[0044] 임의 데이터 Ck는 다음과 같이 다차원 벡터(multi-dimensional vector)로 생성되는데, 본 발명에서는 n차원 치환 행렬(n-th order permutation matrix)의 형태로 생성된다.

[0045] 치환 행렬(permutation matrix)은 각각의 행과 열에 대해 정확히 하나의 원소만 1이고, 나머지 원소들은 모두 0으로 구성된 n차 정방 행렬(n-th order square matrix)이다. 치환 행렬은 n개의 원소들을 행(row) 또는 열(column)에 대해 치환(permutation)하기 위해 사용된다.

[0046] n개의 원소로 이루어진 열 벡터 X에 대해 치환 행렬 P를 적용한 행 치환(row permutation) $RP_p(X)$는 다음과 같다.

[0047] $RP_p(X) = PX$.

[0048] 예를 들어, $P=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}$ 이고 $X=\begin{pmatrix}a\\b\\c\end{pmatrix}$ 이면, $RP_p(X)=PX=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}\begin{pmatrix}a\\b\\c\end{pmatrix}=\begin{pmatrix}b\\c\\a\end{pmatrix}$ 이다. 이것은 원래의 데이터 X에서 첫 번째 행이 세 번째 행으로, 두 번째 행이 첫 번째 행으로, 세 번째 행이 두 번째 행으로 치환됨을 의미한다.

[0049] n개의 원소로 이루어진 행 벡터 X에 대해 치환 행렬 P를 사용한 열 치환(column permutation) $CP_p(X)$는 다음과 같다.

[0050] $CP_p(X) = XP$.

[0051] 예를 들어, $P=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}$ 이고 X = (a b c)이면,

[0052] $CP_p(X)=XP=(a\ b\ c)\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}=(c\ a\ b)$ 이다. 이것은 원래의 데이터 X에서 첫 번째 열이 두 번째 열로, 두 번째 열이 세 번째 열로, 세 번째 열이 첫 번째 열로 치환됨을 의미한다.

[0053] 그리고 치환 행렬의 역행렬(inverse matrix)은 그 치환 행렬의 전치 행렬(transpose matrix)과 동일하다. 이러한 사실은 n의 크기가 증가하더라도 치환 행렬의 역행렬을 찾기가 매우 용이하다는 특성을 가진다.

[0054] 임의 데이터 Ck는 다음과 같이 생성된다.

[0055] $Ck=\begin{pmatrix}ck_{1,1}&\cdots&ck_{1,n}\\\vdots&\ddots&\vdots\\ck_{n,1}&\cdots&ck_{n,n}\end{pmatrix}$. Ck는 n차 치환 행렬이다.

[0056] 암호화 수행부(22)는 상기 X_i를 암호화한 암호화 데이터 Y_i를 생성하는데(s22), 암호화(Y_i의 생성)는 X_i에 Ck를

곱하여 이루어진다.

[0057] $$Y_i = (Y_{i,1} \cdots Y_{i,n}) = CP_{Ck}(X_i) = (X_{i,1} \cdots X_{i,n}) \begin{pmatrix} ck_{1,1} & \cdots & ck_{1,n} \\ \vdots & \ddots & \vdots \\ ck_{n,1} & \cdots & ck_{n,n} \end{pmatrix}$$.

[0058] 여기서 $CP_{Ck}(X_i)$는 Ck를 인자로 하여 원본 데이터(raw data) X_i의 암호화 연산을 나타내는 기호이다. 그리고 이 암호화 연산의 의미는 행 벡터로 이루어진 X_i의 각 비트의 비트값의 순서를 바꾸는 것으로 X_i를 암호화하는 것이다. 예를 들어 $X_i = (1,0,1)$이면 $Y_i = (0,1,1)$으로 암호화 될 수 있다. 이는 상기 언급한 치환 행렬의 특성으로부터 기인한다.

[0059] 한편 이렇게 암호화된 데이터 Y_i의 복호화는, 도 1의 경우와 마찬가지로, 암호화 연산의 역 연산(inverse operation)으로 이루어질 수 있다. 역 연산(복호화)의 경우에는 상기 암호화를 위한 임의 데이터 즉, 치환 행렬 Ck의 역행렬을 이용하여 이루어지는데, 치환 행렬의 역행렬은 위에서 언급한 바와 같이 치환 행렬 자신의 전치 행렬이므로 도 2의 경우에 따라 암호화된 데이터의 복호화는 Ck의 역행렬을 구하기 위한 복잡한 절차가 필요가 없다.

[0060] $$X_i = CP_{Ck}^{-1}(Y_i) = (Y_{i,1} \cdots Y_{i,n}) \begin{pmatrix} ck_{1,1} & \cdots & ck_{1,n} \\ \vdots & \ddots & \vdots \\ ck_{n,1} & \cdots & ck_{n,n} \end{pmatrix}^{-1}$$
$$= (Y_{i,1} \cdots Y_{i,n}) \begin{pmatrix} ck_{1,1} & \cdots & ck_{n,1} \\ \vdots & \ddots & \vdots \\ ck_{1,n} & \cdots & ck_{n,n} \end{pmatrix} = (X_{i,1} \cdots X_{i,n})$$.

[0061] 도 2의 경우에 따른 암호화 기법의 적용 실제를 예를 들어 살펴본다.

[0062] $X_i = (1,0,1)$이고 $Ck = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ 이라 하자.

[0063] $$Y_i = CP_{Ck}(X_i) = X_i \cdot Ck = (101) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = (110)$$.

[0064] 따라서 원본 데이터의 비트값 순서가 바뀌어 X_i에 대한 암호화가 이루어짐을 알 수 있다.

[0065] 복호화는

[0066] $$X_i = CP_{Ck}^{-1}(Y_i) = Y_i \cdot Ck^{-1} = (110) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{-1} = (110) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{T} = (110) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = (101)$$.

[0067] 따라서 완벽하게 원문 데이터 X_i가 복호화됨을 알 수 있다.

[0068] 도 2에 의한 암호화 기법을 사용하면, 도 1에 의한 암호화 기법과 마찬가지로, 원본 데이터들 사이의 비트 차이, 즉 해밍 거리를 보존할 수 있음을 알 수 있다. 도 2에 의한 암호화 기법을 적용한 결과는 원본 데이터들에서 열 간의 치환(permutation)으로 만들어지기 때문에 원본 데이터들의 비트 단위별로 해밍 거리는 그대로 보존된 채 치환되며, 결과적으로 원본 데이터들 사이의 해밍 거리는 그대로 보존된다. 그리고 데이터들이 모두 비

트 문자열이기 때문에 해밍 거리가 보존되는 것은 유클리디안 거리 역시 보존되는 것을 의미한다. 또한, 도 2에 의한 암호화 기법은 암호화/복호화 시에 행렬의 곱셈 및 전치 행렬 계산으로만 구현되기 있기 때문에 암호화/복호화 하드웨어 또는 소프트웨어의 구현에 매우 효율적이다.

[0069] 도 3a는 본 시스템 발명의 바람직한 제3 구성을 제시한 것이며, 도 3b는 본 방법 발명의 바람직한 제3 흐름을 제시한 것이다.

[0070] 도 3에 의한 암호화는 도 1과 도 2에 의한 암호화 기법을 융합시킨 것으로 도 3에 의한 암호화 및 복호화 결과도 도 1과 도 2에 의한 암호화 및 복호화 결과와 동일하게 되는 것을 확인할 수 있다. 도 3에 의한 암호화는 먼저 원본 데이터를 도 1에 의한 암호화 기법에 의해 암호화하고(이하 편의상 '제1 암호화' 라고 한다), 제1 암호화 결과를 도 2에 의한 암호화 기법에 의해 암호화(이하 편의상 '제2 암호화' 라고 한다)하여 원본 데이터의 최종 암호화 결과를 얻는다.

[0071] 제1 암호화 및 제2 암호화에 대한 각각의 상세 사항은 위에서 언급한 바와 동일하므로 설명을 생략하며 다만, 적용 실제를 예를 들어 살펴본다.

[0072] 제1 임의 데이터 생성부(31)는 암호화 대상 n-비트 데이터의 암호화를 위한 제1 임의 데이터 Bk를 생성한다(s31).

[0073] X_i = (1,0,1)이고 생성된 Bk = (1,1,0)이라 하자.

[0074] 제2 임의 데이터 생성부(32)는 상기 n-비트 데이터의 암호화를 위한 제2 임의 데이터 Ck를 다차원 벡터로 생성하는데(s32), 이는 상기한 치환 행렬을 생성하는 것이다.

[0075] $$Ck = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$.

[0076] 연산부(33)는 X_i와 Bk를 비트 대 비트 연산(bit by bit operation)한다(s33). 이는 제1 암호화를 수행하는 것이다.

[0077] $$BF_{Bk}(X_i) = X_i \oplus Bk = (101) \oplus (110) = (011)$$.

[0078] 암호화 완료부(34)는 상기 비트 대 비트 연산의 결과 $BF_{Bk}(X_i)$와 상기 다차원 벡터 Ck를 곱하여 상기 n-비트(n-bit) 데이터의 암호화 데이터 Y_i를 생성하여 암호화를 완료한다(s34). 이는 제2 암호화이다.

[0079] $$Y_i = CP_{Ck}(BF_{Bk}(X_i)) = BF_{Bk}(X_i) \cdot Ck = (011)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = (011)$$.

[0080] 즉, 제1 암호화 및 제2 암호화를 통해 원본 데이터 X_i = (1,0,1)이 Y_i = (0,1,1)로 암호화되었다.

[0081] 복호화는

[0082] $$X_i = CP_{Ck}^{-1}(Y_i) = [Y_i \cdot Ck^{-1}] \oplus Bk$$
$$= [(011)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{-1}] \oplus (110) = [(011)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{T}] \oplus (110) = [(011)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}] \oplus (110) = (101)$$.

[0083] 따라서, 도 3에 의한 암호화 및 복호화의 경우도 도 1 및 도 2의 결과와 같은 결과를 얻을 수 있음을 확인할 수 있다.

[0084] 도 3에 의한 암호화 기법에 의해서도 원본 데이터들 사이의 비트 차이, 즉 해밍 거리는 그대로 보존되는 것을

알 수 있다. 다시 말하면, 도 1 및 도 2에 의한 암호화 기법을 차례대로 결합하여 만든 암호화 기법도 역시 해밍 거리를 보존할 수 있음을 알 수 있다. 그리고 데이터들이 모두 비트 문자열이기 때문에 해밍 거리가 보존되는 것은 유클리디안 거리 역시 보존되는 것을 의미한다. 또한, 도 3에 의한 암호화 기법도 도 1 및 도 2의 경우와 마찬가지로 암호화/복호화 시에 XOR 연산, 행렬의 곱셈 및 전치 행렬 계산으로만 구현되기 있기 때문에 암호화/복호화 하드웨어 또는 소프트웨어의 구현에 매우 효율적이다.

[0085] 기존의 거리 보존 암호화 기법과 비교하여 본 발명에 의한 거리 보존 암호화 기법의 효율성에 대해 언급하면 다음과 같다.

[0086] 기존의 거리 보존 암호화 기법은 기본적으로 직교 행렬(Orthogonal matrix)을 곱하고 상수를 더하는 형태로 표현된다. 즉, 기존의 거리 보존 암호화 기법을 일반화한 식은 다음과 같다.

[0087] $y = Mx + v$.

[0088] 여기서 y는 열 벡터로 표현된 암호문이고, M은 n×n 직교행렬이며, x는 열 벡터로 표현된 평문(plain text), v는 열 벡터로 표현된 상수이다. 이 식에 의하면 n-비트 데이터를 암호화할 경우 n번의 덧셈 연산과 번의 행렬 곱 계산이 요구되며, 이때의 행렬 곱 계산은 n×n 행렬과 n×1 행렬의 곱이기 때문에 n^2번의 곱셈과 n(n-1)번의 덧셈으로 구성된다. 그리고 이것을 복호화할 경우 전치 행렬을 구해야 하는 것이 추가적으로 요구된다. 여기서 직교 행렬의 역행렬은 전치 행렬과 동일하므로, 이것을 계산하는 것은 정확히 n(n-1)/2번의 치환만으로 계산이 가능하다.

[0089] 본 발명에 의한 거리 보존 암호화 기법 역시 위의 식과 같은 형태로 변형할 수 있다. 즉, 다음과 같은 식으로 표현이 가능하다.

[0090] $Y = M`X + V`$.

[0091] 여기서 Y는 열벡터로 표현된 암호문이고, M`은 Bk의 j번째 원소 $Bk_j$가 1인 모든 j에 대해 Ck의 j번째 열의 모든 원소들에 -1을 곱해준 n×n 행렬이며, X는 열벡터로 표현된 평문, V`은 $Ck \cdot Bk^T$이다. 여기서 Bk^T는 Bk의 전치(transpose)를 의미한다. 이것은 n-비트 데이터를 암호화할 경우 n번의 XOR 연산과 1번의 행렬 곱 계산만이 요구된다. 그중에서도 특히 행렬 곱 계산은 n차 치환 행렬을 사용하기 때문에 최대 n번의 치환만으로 계산이 가능하다.

[0092] 그리고 복호화할 경우 요구되는 연산은 암호화 시에 요구되는 연산에서 치환 행렬의 역행렬을 계산하는 것이 추가될 뿐인데, 치환 행렬의 역행렬은 전치행렬과 동일하므로, 이것을 계산하는 것은 정확히 n(n-1)/2번의 치환만으로 계산이 가능하다. 따라서 본 발명에 의한 거리 보존 암호화/복호화 기법에 소요되는 연산량은 이진 데이터를 사용하는 경우에 기존의 기법들보다 매우 효율적이라고 할 수 있다.

[0093] 그리고 본 발명에 의한 거리 보존 암호화 기법은 기존의 거리 보존 암호화 기법들과는 달리 이진 데이터를 사용하는 경우에 암호문의 길이 확장(length expansion)이 발생하지 아니한다. 기존의 거리 보존 암호화 기법들을 사용한 암호문은 평문의 입력이 무엇이든 상관없이 직교 행렬과의 연산을 통해 실수로 표현되고, 컴퓨터에서 부동 소수점을 표현하는 규약을 정의해 놓은 IEEE 754 표준에 따르면 최소한 32비트 단정도(single-precision)로 구현되어야 한다.

[0094] 다시 말하면, 기존의 거리 보존 암호화 기법은 평문의 입력이 비트 원소로 구성되어있다고 하더라도 암호문의 원소들은 최소한 32비트로 구현되어야 한다는 것을 의미하며, 결과적으로 암호문을 표현하기 위해서는 평문의 길이보다 최소 32배 길어진다는 것을 의미한다. 하지만, 본 발명에 의한 암호화 기법은 평문의 입력이 n-비트 원소로 구성된다면 암호문의 원소들 역시 n-비트 원소로 구성되어 상기한 길이 확장(length expansion)이 발생하지 아니한다. 아래 표는 기존의 거리 보존 암호화 기법과 제안하는 기법의 효율성을 비교한 것이다.

		기존의 기법		본 발명에 의한 기법	
계산량 (연산량)	암호화	덧셈	n^2 번	XOR	n 번
		곱셈	n^2 번	치환	n 번
	복호화	덧셈	n^2 번	XOR	n 번
		곱셈	n^2 번		
		치환	n(n-1)/2 번	치환	n(n-1)/2 번
암호문의 길이 증가 (Length expansion)		○		×	

[0095]

[0096] 본 방법발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

[0097] 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 유무선 네트워크를 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

[0098] 이제까지 본 발명에 대하여 그 바람직한 실시예를 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다.

[0099] 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 균등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

도면의 간단한 설명

[0100] 도 1a는 본 시스템 발명의 바람직한 제1 구성을 제시한 도면이다.

[0101] 도 1b는 본 방법 발명의 바람직한 제1 흐름을 제시한 도면이다.

[0102] 도 2a는 본 시스템 발명의 바람직한 제2 구성을 제시한 도면이다.

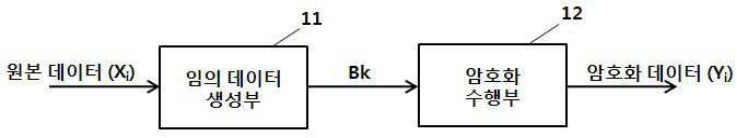
[0103] 도 2b는 본 방법 발명의 바람직한 제2 흐름을 제시한 도면이다.

[0104] 도 3a는 본 시스템 발명의 바람직한 제2 구성을 제시한 도면이다.

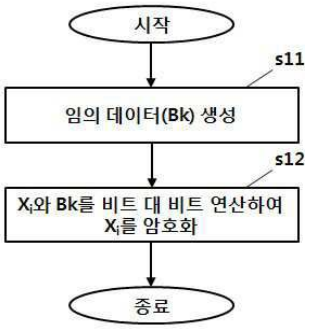
[0105] 도 3b는 본 방법 발명의 바람직한 제2 흐름을 제시한 도면이다.

도면

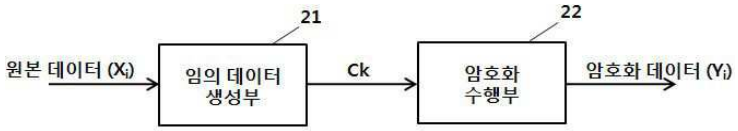
도면1a



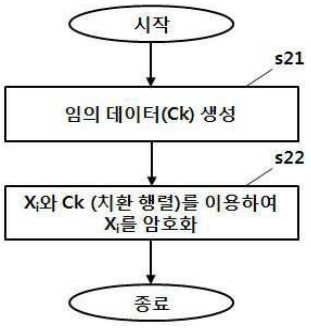
도면1b



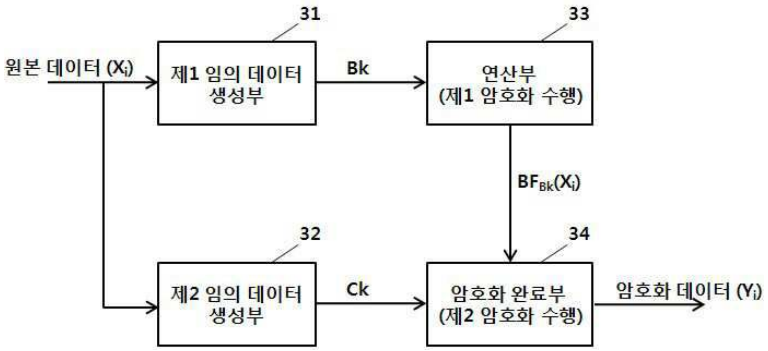
도면2a



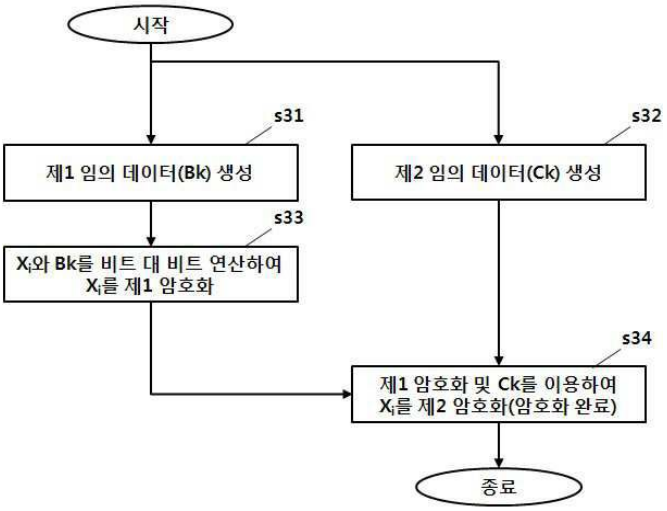
도면2b



도면3a



도면3b


시작
s31
제1 임의 데이터(Bk) 생성
s32
제2 임의 데이터(Ck) 생성
s33
Xi와 Bk를 비트 대 비트 연산하여
Xi를 제1 암호화
s34
제1 암호화 및 Ck를 이용하여
Xi를 제2 암호화(암호화 완료)
종료