

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)(11) 공개번호 10-2023-0026774
(43) 공개일자 2023년02월27일

(51) 국제특허분류(Int. Cl.)

G16H 50/50 (2018.01) G06N 20/20 (2019.01)

G16H 50/20 (2018.01) G16H 50/70 (2018.01)

H04L 9/00 (2022.01)

(52) CPC특허분류

G16H 50/50 (2018.01)

G06N 20/20 (2021.08)

(21) 출원번호 10-2021-0108794

(22) 출원일자 2021년08월18일

심사청구일자 2021년08월18일

(71) 출원인

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

윤지원

서울특별시 용산구 한강대로43길 8, 102동 1401호

홍미연

경기도 성남시 분당구 내정로 152, 130동 901호

유준수

서울특별시 송파구 백제고분로36나길 28-1, 3층

(74) 대리인

김홍석

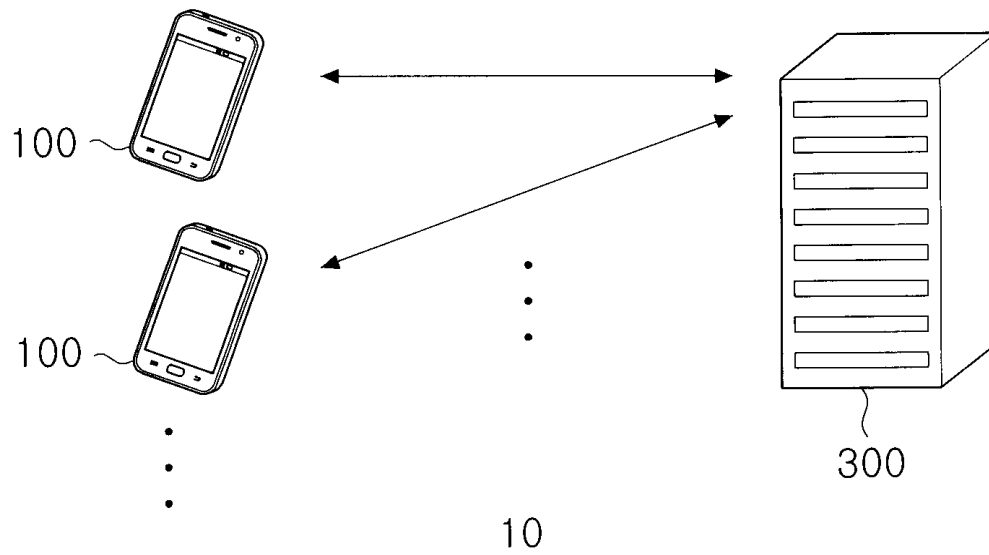
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 암호화된 데이터상에서 비트 단위 기반 완전동형암호 기법을 이용한 질병 진단 장치 및 방법

(57) 요약

질병 진단 방법이 개시된다. 상기 질병 진단 방법은 적어도 프로세서를 포함하는 질병 진단 장치에 의해 수행되고, 사용자 단말로부터 특정 질병의 측정 요소들 각각에 대한 암호화된 측정치들을 수신하는 단계, 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는 단계, 상기 암호화된 측정치들 각각의 비교 결과에 대한 AND 연산을 수행하는 단계, 및 AND 연산의 결과를 상기 사용자 단말로 송신하는 단계를 포함한다.

대표도 - 도4



(52) CPC특허분류

G16H 50/20 (2018.01)

G16H 50/70 (2018.01)

H04L 9/008 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134646
과제번호	2021-0-00558-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D, 정보화)
연구과제명	동형암호 기술을 활용한 국가통계 분석 시스템 개발
기 여 율	1/1
과제수행기관명	(주)마크애니
연구기간	2021.04.01 ~ 2021.12.31

명세서

청구범위

청구항 1

적어도 프로세서를 포함하는 질병 진단 장치에 의해 수행되는 질병 진단 방법에 있어서,
 사용자 단말로부터 특정 질병의 측정 요소들 각각에 대한 암호화된 측정치들을 수신하는 단계;
 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는 단계;
 상기 암호화된 측정치들 각각의 비교 결과에 대한 AND 연산을 수행하는 단계; 및
 AND 연산의 결과를 상기 사용자 단말로 송신하는 단계를 포함하는 질병 진단 방법.

청구항 2

제1항에 있어서,
 상기 암호화된 측정치들 각각은,
 측정치를 비트 단위로 표현하는 단계; 및
 각 비트에 대하여 완전동형암호화를 수행하는 단계를 수행함으로써 생성된,
 질병 진단 방법.

청구항 3

제1항에 있어서,
 상기 비교하는 단계는 하기의 제1 알고리즘(Algorithm 1)을 수행함으로써 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는,
 질병 진단 방법.
 [제1 알고리즘]

Algorithm 1 HE.CompareLE: Less than or equal algorithm in an encrypted state

```

1:  $ct.t_1[0] \leftarrow \text{Enc}(0)$ 
2: for  $i = 1$  to  $l - 1$  do
3:    $ct.t_2 \leftarrow \text{HE.XNOR}(ct.\hat{a}[i], ct.\hat{b}[i])$ 
4:    $ct.t_1 \leftarrow \text{HE.MUX}(ct.t_2, ct.t_1, ct.\hat{a}[i])$ 
5: end for
6:  $\hat{r} \leftarrow \text{HE.NOT}(ct.t_1)$ 
7: return  $\hat{r}$ 
    
```

청구항 4

제1항에 있어서,
 상기 AND 연산을 수행하는 단계는 완전동형암호의 AND 연산을 수행하는,
 질병 진단 방법.

청구항 5

적어도 프로세서를 포함하는 질병 진단 장치에 의해 수행되는 질병 진단 방법에 있어서,
 사용자 단말로부터 특정 질병의 측정 요소들 각각에 대한 암호화된 측정치들을 수신하는 단계;

상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는 단계;
 상기 암호화된 측정치들 각각의 비교 결과에 대한 덧셈 연산을 수행하는 단계; 및
 덧셈 연산의 결과를 상기 사용자 단말로 송신하는 단계를 포함하는 질병 진단 방법.

청구항 6

제5항에 있어서,
 상기 암호화된 측정치들 각각은,
 측정치를 비트 단위로 표현하는 단계; 및
 각 비트에 대하여 완전동형암호화를 수행하는 단계를 수행함으로써 생성된,

청구항 7

제5항에 있어서,
 상기 비교하는 단계는 하기의 제1 알고리즘(Algorithm 1)을 수행함으로써 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는,
 질병 진단 방법.

[제1 알고리즘]

Algorithm 1 HE.CompareLE: Less than or equal algorithm in an encrypted state

```

1:  $ct.t_1[0] \leftarrow \text{Enc}(0)$ 
2: for  $i = 1$  to  $l - 1$  do
3:    $ct.t_2 \leftarrow \text{HE.XNOR}(ct.\hat{a}[i], ct.\hat{b}[i])$ 
4:    $ct.t_1 \leftarrow \text{HE.MUX}(ct.t_2, ct.t_1, ct.\hat{a}[i])$ 
5: end for
6:  $\hat{r} \leftarrow \text{HE.NOT}(ct.t_1)$ 
7: return  $\hat{r}$ 
    
```

청구항 8

제5항에 있어서,
 상기 덧셈 연산을 수행하는 단계는 하기의 제2 알고리즘(Algorithm 2)을 수행함으로써 상기 암호화된 측정치들 각각의 비교 결과에 대한 덧셈 연산을 수행하는,
 질병 진단 방법.

[제2 알고리즘]

Algorithm 2 HE.Addition: Addition algorithm in an encrypted state

```

1:  $ct.\hat{c}[0] \leftarrow \text{Enc}(0)$ 
2: for  $i = 1$  to  $l - 1$  do
3:    $ct.t_1 \leftarrow \text{HE.XOR}(\hat{a}[i], \hat{b}[i])$ 
4:    $ct.t_2 \leftarrow \text{HE.AND}(\hat{a}[i], \hat{b}[i])$ 
5:    $ct.\hat{r}[i] \leftarrow \text{HE.XOR}(ct.t_1, \hat{c}[i])$ 
6:    $ct.t_1 \leftarrow \text{HE.AND}(ct.t_1, \hat{c}[i])$ 
7:    $ct.\hat{c}[i + 1] \leftarrow \text{HE.OR}(ct.t_1, ct.t_2)$ 
8: end for
9:  $ct.t_1 \leftarrow \text{HE.XOR}(ct.\hat{a}[l - 1], ct.\hat{b}[l - 1])$ 
10:  $ct.\hat{r}[l - 1] \leftarrow \text{HE.XOR}(ct.t_1, ct.\hat{c}[l - 1])$ 
11: return  $\hat{r}$ 
    
```

발명의 설명

기술 분야

[0001] 본 발명은 사용자의 데이터 프라이버시를 보존하면서 사용자 데이터를 분석할 수 있는 방법에 관한 것으로, 특히 비트 단위 암호화를 기반으로 하는 완전동형암호를 이용하여 데이터 프라이버시를 보존하고 별도의 복호화 없이 질병 진단을 수행할 수 있는 장치 및 방법에 관한 것이다.

배경 기술

[0002] 클라우드가 사용자가 원하는 서비스를 대행해 주는 클라우드 기반 서비스 기술이 급증하고 있다. 문제는 사용자의 데이터를 클라우드와 함께 공유함으로써, 사용자의 데이터 프라이버시에 대한 문제가 발생한다는 것이다. 이에 대응하여, 동형암호 기술(homomorphic encryption)을 사용하게 되면 서버는 사용자의 데이터에 대해 아무런 정보를 얻지 않고, 다만 사용자가 요청한 쿼리(query)만을 수행하게 된다. 클라우드는 사용자에게 결과값을 돌려주고, 사용자는 개인정보를 노출하지 않고 결과값을 얻을 수 있다.

[0003] 최근에는 클라우드 서버가 기계학습(machine learning) 서비스를 사용자에게 제공하는 MLaaS(Machine Learning as a Service) 형태가 증가하고 있다. 기계학습의 기초가 되는 논리 게이트(Logic gate)는 A와 B라는 입력값을 받아서 어떤 결과값 Y를 도출하게 해주는 장치이다. 본 발명에서는 동형암호를 통해 사용자 데이터 프라이버시를 보존하면서 논리 게이트를 바탕으로 한 알고리즘을 이용해 질병을 진단할 수 있는 장치 또는 방법론을 다룬다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 대한민국 공개특허 제2019-0117286호 (2019.10.16.)
(특허문헌 0002) 대한민국 등록특허 제1677114호 (2016.11.18.)
(특허문헌 0003) 대한민국 등록특허 제1923293호 (2018.07.23.)

발명의 내용

해결하려는 과제

[0005] 본 발명이 이루고자 하는 기술적인 과제는 암호화된 데이터상에서 비트 단위 기반 완전동형암호 기법을 이용한 질병 진단 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0006] 본 발명의 일 실시예에 따른 질병 진단 방법은 적어도 프로세서를 포함하는 질병 진단 장치에 의해 수행되고, 사용자 단말로부터 특정 질병의 측정 요소들 각각에 대한 암호화된 측정치들을 수신하는 단계, 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는 단계, 상기 암호화된 측정치들 각각의 비교 결과에 대한 AND 연산을 수행하는 단계, 및 AND 연산의 결과를 상기 사용자 단말로 송신하는 단계를 포함한다.

[0007] 본 발명의 다른 실시예에 따른 질병 진단 방법은 적어도 프로세서를 포함하는 질병 진단 장치에 의해 수행되고, 사용자 단말로부터 특정 질병의 측정 요소들 각각에 대한 암호화된 측정치들을 수신하는 단계, 상기 암호화된 측정치들 각각을 대응되는 기준치와 비교하는 단계, 상기 암호화된 측정치들 각각의 비교 결과에 대한 덧셈 연산을 수행하는 단계, 및 덧셈 연산의 결과를 상기 사용자 단말로 송신하는 단계를 포함한다.

발명의 효과

[0008] 본 발명의 실시예에 따른 질병 진단 장치 및 방법에 의할 경우, 프라이버시를 보존하면서 데이터를 분석할 수 있는 효과가 있다. 특히, 인간 유전자 게놈 데이터 또는 금융 데이터와 같이 높은 프라이버시가 요구되는 분야에서 데이터 처리를 사용하는 곳에 적용이 가능하다.

[0009] 또한, 본 발명이 실시예에 따른 질병 진단 장치 및 방법은 다른 동형암호 머신러닝 분야에 적용이 가능하다. 본 발명에서 제안하는 논리 게이트는 연산이 간단하면서도 정확한 결과값을 도출하기 때문에 스마트 헬스케어 산업에서의 다양한 머신러닝 모델에 적용할 수 있다.

[0010] 또한, 본 발명의 실시예에 따른 질병 진단 장치 및 방법은 스마트 헬스케어 산업에서 활용이 가능하다. 본 발명은 의료계에 종사하는 사람들이 질병의 치료보다 진단과 예방의 차원에서 활용이 가능하다. 직접 병원을 방문하지 하고, 측정 장비를 활용해 검사를 실시한다면, 암호화된 데이터는 클라우드에서 연산을 실시하고, 결과값을 의사와 같은 의료계 종사자들에게 공유할 수 있다.

도면의 간단한 설명

[0011] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 암호화된 실수 데이터의 표기법을 설명하기 위한 도면이다.

도 2는 동형암호 작거나 같음 연산 알고리즘을 도시한다.

도 3은 동형암호 덧셈 연산 알고리즘을 도시한다.

도 4는 본 발명의 일 실시예에 따른 진단 시스템을 도시한다.

도 5는 도 4에 도시된 서버의 기능블록도이다.

도 6은 2-Input AND 게이트의 입력값에 대한 출력값을 도시한다.

도 7은 AND 게이트 연산을 통해 결과값을 반환하는 다중 입력 논리 게이트를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0013] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0014] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0015] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0016] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0017] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0018] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위

가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

1. 기호 및 동형암호 함수 정리

(1) 기호 정의

본 명세서에서 사용되는 기호는 다음과 같이 표현된다.

배열(벡터)은 알파벳 기호 $\hat{a}$ 로 나타내며, l 은 배열의 길이(length)를 나타낸다. 각각의 인덱스에 해당하는 요소들은 $\hat{a}[i]$ 로 나타낸다. 여기서 i 는 정수로써, 0부터 l 사이의 값을 갖는다. 또한, Enc 는 암호화를 의미하고, Dec 는 복호화를 의미한다.

용어 *bitwise*는 특정 연산을 각 배열의 요소들 모두에게 각각 적용하는 것을 의미한다.

HE는 동형암호 게이트들과 일반 게이트들과의 차이를 나타내기 위해 표시한다. 일반적으로 완전동형암호 게이트는 연산 이후 잡음(noise)을 제거하기 위한 부트스트래핑(bootstrapping)이 필요하다.

예컨대, $\hat{c}[0] \leftarrow HE.AND(\hat{a}[0], \hat{b}[0])$ 는 두 입력 암호비트 $\hat{a}[0], \hat{b}[0]$ 에 대해서 AND 연산을 수행하고, 연산 도중 발생하는 잡음이 부트스트래핑에 의해 제거되어 반환되는 암호비트 값을 $\hat{c}$ 배열의 0번째 요소에 저장하는 것을 의미한다.

(2) 동형암호 함수

아래 표는 동형암호 함수 테이블로써, 동형암호를 이용한 질병 진단 시스템에 이용될 수 있다.

번호	분류	(동형암호) 함수	기호
1	대소비교	작거나 같다($\leq$)	HE.CompareLE
3	기본연산	덧셈(+)	HE.Addition

2. 동형암호 실수 표현 및 기초 연산

(1) 암호화된 데이터의 실수 표현 기술

종래 기술에 의할 경우, 정수 기반 동형암호 스키마를 이용한 방법은 실수 데이터에 대해 특정 값을 곱하여 정수로 만든 후 암호 알고리즘을 진행한다. 이로 인하여 오차가 발생하게 되는 단점이 있다.

하지만, 본 발명에서는 비트 단위 기반 동형암호 스키마를 사용하여, 실수 유효 숫자 범위를 유동적으로 정할 수 있다. 구체적으로, 도 1과 같이 암호화된 실수 데이터를 표현할 수 있다.

구체적으로 설명하면, 먼저 암호문 배열의 인덱스를 반대로 설정한다. 즉, 가장 오른쪽이 최상위 비트가 되게 한다.

일반적으로, 입력한 비트의 길이를 l , 암호문의 배열을 $\hat{a}$, i 번째 배열의 성분을 $\hat{a}[i]$ 라 하면, 인덱스 0부터 $(\frac{l}{2}-1)$ 까지의 $\frac{l}{2}$ 만큼을 실수 데이터의 소수 부분으로 두고(할당하고), 인덱스 $\frac{l}{2}$ 부터 $l-2$ 까지 $(\frac{l}{2}-1)$ 만큼을 실수 데이터의 정수 부분으로 둔다(할당한다). 마지막, 최상위 비트인 l 번째 비트를 부호 비트로 두고(할당하고) 실수를 표현한다.

상술한 데이터의 비트 단위 표현은 후술할 사용자 단말에 의해 수행된다. 즉, 사용자 단말은 실수값인 소정의 측정치를 비트 단위로 표현(변환)할 수 있다. 추가적으로, 사용자 단말은 비트 단위로 표현된 측정치에 완전동

형암호화를 수행한 후 암호화된 데이터를 서버로 송신할 수 있다.

- [0039] (2) 암호화된 데이터 간에 복호화 없이 비교 연산을 수행하는 기술
- [0040] 본 발명에서는 암호문 간에 대소비교 알고리즘 중 "Less than or equal(작거나 같음)"을 구축하였다. 이는 두 값을 비교하여 먼저 입력한 값이 작거나 같음을 알리는 **HE.CompareLE** 함수로 표현한다.
- [0041] ○ 동형암호 대소비교(대)
- [0042] 도 2는 동형암호 작거나 같음 연산 알고리즘으로써, 암호문 $\hat{a}$ 와 $\hat{b}$ 를 동형비교하여 $\hat{a}$ 가 $\hat{b}$ 보다 작거나 같으면 $Enc(1)$ 을 반환하고, 그렇지 않으면(크면) $Enc(0)$ 을 반환하는 알고리즘이다. 즉, $Dec(\hat{a}) \leq Dec(\hat{b})$ 이면 $Enc(1)$, 반대의 경우는 $Enc(0)$ 이다.
- [0043] 여기서, **HE.MUX(a, b, c)** 게이트는 평문에서의 MUX 게이트와 같은 역할로 a의 값에 따라 b 또는 c를 출력한다.
- [0044] 상술한 비교 연산은, 후술할 서버에 포함되는 비교 연산부에 의해 수행될 수 있다.
- [0046] ○ 동형암호 덧셈 기술
- [0047] 평문 상에서는 전가산기(full-adder)를 통해 덧셈을 구성할 수 있다. 마찬가지로, 이를 활용하면 암호문 간의 덧셈도 전가산기의 게이트와 같은 원리로 설계할 수 있다. 이를 나타내는 알고리즘(동형암호 덧셈 연산 알고리즘)은 도 3에 도시되어 있다.
- [0048] 도 3의 알고리즘을 통해 동형암호 덧셈 결과값이 반환된다. HE.AND, HE.OR, HE.XOR의 게이트들은 게이트에 입력한 두 입력값에 대해서 연산 도중 발생한 노이즈(noise)를 제거하는 부트스트래핑 게이트들이다.
- [0049] 상술한 덧셈 연산은 후술할 서버의 덧셈 연산부에 의해 수행될 수 있다.
- [0051] 도 4는 본 발명의 일 실시예에 따른 진단 시스템을 도시한다.
- [0052] 도 1을 참조하면, 진단 시스템(10)은 적어도 하나의 사용자 단말(100)과 서버(300)를 포함한다. 진단 시스템(10)에서, 사용자 단말(100)은 측정치들을 암호화한 후 암호화된 데이터를 서버(300)로 송신하고, 서버(300)는 수신된 암호화된 데이터를 이용하여 질병의 유무 및/또는 질병위험 요소의 개수에 대한 정보를 생성하고, 생성된 정보를 사용자 단말(100)로 송신할 수 있다. 사용자 단말(100)과 서버(300)는 소정이 유무선 통신망을 통하여 통신할 수 있다.
- [0053] 사용자 단말(100)은 프로세서(processor) 및/또는 메모리를 포함하는 컴퓨팅 장치로써, 데이터에 대한 암호화 동작과 암호문에 대한 복호화 동작을 수행할 수 있다. 또한, 사용자 단말(100)은 암호화된 데이터를 서버(300)로 송신하거나 암호화된 데이터를 서버(300)로부터 수신할 수 있다.
- [0054] 구체적으로, 사용자 단말(100)은 사용자 데이터를 자신의 개인키로 암호화하고 암호화된 데이터를 서버(300)로 송신할 수 있다. 예컨대, 사용자 단말(100)은 측정 요소들(예컨대, 나이, 체중, 공복 혈당, HDL 수치, 혈압 등) 각각에 대한 측정치들을 비트 단위 암호화할 수 있다. 다시 말해, 사용자 단말(100)은 각 측정치를 비트 단위로 표현한 후 각 비트에 대해서 소정의 암호화 알고리즘(예컨대, 완전동형암호의 암호화 알고리즘)을 적용하여 암호화된 데이터(암호화된 측정치)를 생성할 수 있다. 생성된 암호화된 데이터는 서버(300)로 송신될 수 있다. 이를 통해, 서버는 비트 단위의 완전동형암호 기법을 이용할 수 있다.
- [0055] 또한, 사용자 단말(100)은 서버(300)로부터 암호화된 최종 결과값을 수신하고 이를 복호화할 수 있다. 복호화 동작을 통해 사용자는 특정 질병의 유무와 특정 질병에 대하여 해당하는 질병위험 요소의 개수를 파악할 수 있다.
- [0056] 사용자 단말(100)은 PC(personal computer), 태블릿 PC, 노트북(notebook), 넷-북(net-book), e-리더(e-

reader), PDA(personal digital assistant), PMP(portable multimedia player), MP3 플레이어, 또는 MP4 플레이어와 같은 데이터 처리 장치(또는 컴퓨팅 장치)로 구현될 수 있다.

- [0057] 질병 진단 장치, 질병 진단 서버 등으로 명명될 수도 있는 서버(300)는 적어도 프로세서 및/또는 메모리를 포함하는 컴퓨팅 장치, 예컨대 범용 서버로써, 사용자 단말(100)로부터 수신된 암호화된 데이터를 수신하고, 수신된 암호화된 데이터를 분석, 가공, 연산 등의 동작을 수행함으로써 소정의 암호화된 결과물을 도출하고, 도출된 암호화된 결과물은 다시 사용자 단말(100)로 송신할 수 있다.
- [0059] 도 5는 도 4에 도시된 서버의 기능 블록도이다.
- [0060] 도 5를 참조하면, 서버(300)는 데이터 수신부(310), 데이터 연산부(320), 및 데이터 송신부(330)를 포함한다. 실시예에 따라, 서버(300)는 저장부(340)를 더 포함할 수도 있다.
- [0061] 데이터 수신부(310)는 사용자 단말(100)로부터 암호화된 데이터(비트 단위로 암호화된 데이터)를 수신할 수 있다. 수신된 암호화된 데이터는 저장부(340)에 저장될 수 있다.
- [0062] 데이터 연산부(320)는 암호화된 데이터를 대상으로 미리 정해진 연산 동작을 수행함으로써 연산 결과를 도출할 수 있다. 물론, 데이터 연산부(320)는 암호화된 데이터를 복호화하지 않고 암호화된 상태에서 연산 동작을 수행할 수 있다. 보다 구체적으로, 데이터 연산부(320)는 비교 연산부, AND 연산부, 및 덧셈 연산부 중 적어도 하나를 포함할 수 있다. 즉, 서버(300)가 질병의 진단 동작만을 수행하는 경우 데이터 연산부(320)는 비교 연산부와 AND 연산부만을 포함하고, 서버(300)가 질병위험 요소의 개수 판단 동작을 수행하는 경우 데이터 연산부(320)는 비교 연산부와 덧셈 연산부만을 포함할 수 있다. 물론, 서버(300)가 질병의 진단 동작과 질병위험 요소의 개수 판단 동작을 모두 수행하는 경우, 데이터 연산부(320)는 비교 연산부, AND 연산부, 및 덧셈 연산부를 모두 포함할 수 있다.
- [0063] 비교 연산부는 비교 연산 알고리즘(HE.CompareLE)을 이용하여 입력되는 두 개의 데이터의 대소를 비교할 수 있다. AND 연산부는 AND 연산 알고리즘을 이용하여 입력되는 두 개 이상의 데이터에 대한 AND 연산을 수행할 수 있다. 덧셈 연산부는 덧셈 연산 알고리즘(HE.Addition)을 이용하여 입력되는 두 개 이상의 데이터에 대한 덧셈 연산을 수행할 수 있다.
- [0064] 데이터 송신부(330)는 데이터 연산부(320)의 연산 결과를 사용자 단말(100)로 송신할 수 있다. 이때, 송신되는 연산 결과는 암호화된 데이터이다.
- [0065] 저장부(340)에는 데이터 수신부(310)에 의해 수신된 데이터, 데이터 연산부(320)에 의한 연산 결과, 연산 과정에서 일시적으로 또는 비일시적으로 생성되는 데이터, 각 측정 요소들에 대한 기준치 등이 저장될 수 있다.
- [0067] 이하에서는, 비트 단위 기반으로 하는 완전동형암호 질병진단 시스템 시나리오를 설명한다.
- [0068] (1) 시나리오 1 - 질병 유무 판단
- [0069] AND 게이트 구조와 기호 및 Multi-Input AND 게이트 구성
- [0070] AND 게이트의 출력은 게이트에 대한 모든 입력이 참(논리 1)인 경우에만 참을 나타낸다. 예를 들어, A와 B의 두 입력값에 대한 출력값은 도 6과 같이 나타낼 수 있다.
- [0071] 본 발명에서는 질병의 진단을 위해 디바이스로 측정된 여러 측정 값들을 입력값으로 넣어야 하기 때문에 동일한 유형의 게이트를 결합하여 다중 입력 게이트를 만든다. 도 7은 l 개의 입력값들이 동시에 AND 게이트 연산을 통해 결괏값을 돌려주는 과정을 도식화한 것이다. 완전동형암호에서 AND 연산은 잘 알려져 있고 기본 라이브러리로 제공되기 때문에, 이에 대한 상세한 설명은 생략하기로 한다.
- [0072] 당뇨병 진단의 예를 들어보면, 먼저 사람 A의 측정 요소들(나이, 체중, 공복 혈당, HDL 수치, 혈압 등)은 각각 $C_A^1, C_A^2, C_A^3, \dots, C_A^l$ 로 비트 단위 암호화된다. 암호화 동작, 구체적으로 측정치의 비트 단위 표현(변환) 동작과 각 비트에 대한 완전동형 암호화 동작은 사용자 단말(100)에 의해 수행되고, 암호화된 결과물은 서버(300)로 송신된다. 서버(300)의 데이터 수신부(310)는 사용자 단말(100)로부터 암호화된 데이터를 수신하고 수신된 데이터를 저장부(340)에 저장할 수 있다.

- [0073] 데이터 수신부(310)에 의해 수신된 암호화된 데이터 또는 데이터 수신부(310)에 의해 수신되어 저장부(340)에 저장된 암호화된 데이터(측정치)는 먼저 HE.CompareLE 함수를 이용해 질병 진단을 위한 기준치와 비교된다. 구체적으로, 서버(300)의 데이터 연산부(320)에 포함된 비교 연산부는 측정 요소들 각각을 대응되는 기준치와 비교하고 비교 결과를 출력한다. 예컨대, 비교 연산부는 제1 측정 요소에 대한 사용자 A의 암호화된 측정값 C_A^1 과 제1 측정 요소에 대한 기준치인 제1 기준치(암호화된 기준치를 의미할 수 있음)에 대하여 비교 연산을 수행할 수 있다. 결과적으로, 비교 연산부는 l 개의 측정 요소들에 대한 사용자 A의 암호화된 측정값들을 대응되는 기준치와 비교하는 비교 연산을 수행할 수 있다. 비교 연산부의 결과값은 $Enc(1)$ 또는 $Enc(0)$ 이다. 즉, 해당 측정 요소에 대한 측정값이 대응되는 기준치를 충족하는 경우의 결과값은 $Enc(1)$ 이고, 그렇지 않은 경우 $Enc(0)$ 일 수 있다. 이를 위해, 측정 요소들에 따라 입력 데이터의 입력 순서가 상이할 수 있다. 예컨대, 나이의 경우 기준치와 측정치가 순차적으로 입력되어 측정치가 기준치보다 큰 경우에 $Enc(1)$ 을 출력하고 그렇지 않은 경우 $Enc(0)$ 을 출력할 수 있다.
- [0074] 다중 입력을 위한 결과값들이 $Enc(0)$ 또는 $Enc(1)$ 로 표현되면, 각각의 값들은 AND 게이트 연산을 통해 최종 결과값(R_A)을 반환한다. 구체적으로, 서버(300)의 데이터 연산부(320)에 포함된 AND 연산부는 복수의 입력들(비교 연산부의 출력들을 의미할 수 있음)에 대한 AND 연산을 수행하고 연산 결과를 출력한다. 최종 결과값(R_A)은 $Enc(1)$ 또는 $Enc(0)$ 일 수 있다. 이때, 최종 결과값이 $Enc(1)$ 이면 질병에 해당하는 것으로 결정되고, $Enc(0)$ 이면 질병이 아닌 것으로 판단될 수 있다.
- [0075] 최종 결과값은 서버(300)의 데이터 송신부(330)에 의해 사용자 단말(100)로 송신될 수 있다.
- [0077] (2) 시나리오 2 - 질병위험 요소의 수 판단
- [0078] 시나리오 1의 예와 마찬가지로, 사람 A의 측정 요소들(나이, 체중, 공복 혈당, HDL 수치, 혈압 등)을 각각 $C_A^1, C_A^2, C_A^3, \dots, C_A^l$ 로 비트 단위 암호화한다. 암호화 동작, 구체적으로 측정치의 비트 단위 표현(변환) 동작과 각 비트에 대한 완전동형 암호화 동작은 사용자 단말(100)에 의해 수행되고, 암호화된 결과물은 서버(300)로 송신된다. 서버(300)의 데이터 수신부(310)는 사용자 단말(100)로부터 암호화된 데이터를 수신하고 수신된 데이터를 저장부(340)에 저장할 수 있다.
- [0079] 데이터 수신부(310)에 의해 수신된 암호화된 데이터 또는 데이터 수신부(310)에 의해 수신되어 저장부(340)에 저장된 암호화된 데이터(측정치)는 먼저 HE.CompareLE 함수를 이용하여 질병 진단을 위한 기준치와 비교된다. 구체적으로, 서버(300)의 데이터 연산부(320)에 포함된 비교 연산부는 측정 요소들 각각을 대응되는 기준치와 비교하고 비교 결과를 출력한다. 예컨대, 비교 연산부는 제1 측정 요소에 대한 사용자 A의 암호화된 측정값 C_A^1 과 제1 측정 요소에 대한 기준치인 제1 기준치(암호화된 기준치를 의미할 수 있음)에 대하여 비교 연산을 수행할 수 있다. 결과적으로, 비교 연산부는 l 개의 측정 요소들에 대한 사용자 A의 암호화된 측정값들을 대응되는 기준치와 비교하는 비교 연산을 수행할 수 있다. 비교 연산부의 결과값은 $Enc(1)$ 또는 $Enc(0)$ 이다. 즉, 해당 측정 요소에 대한 측정값이 대응되는 기준치를 충족하는 경우의 결과값은 $Enc(1)$ 이고, 그렇지 않은 경우 $Enc(0)$ 일 수 있다. 이를 위해, 측정 요소들에 따라 입력 데이터의 입력 순서가 상이할 수 있다. 예컨대, 나이의 경우 기준치와 측정치가 순차적으로 입력되어 측정치가 기준치보다 큰 경우에 $Enc(1)$ 을 출력하고 그렇지 않은 경우 $Enc(0)$ 을 출력할 수 있다.
- [0080] 다중 입력을 위한 결과값들이 $Enc(0)$ 또는 $Enc(1)$ 로 표현되면, 각각의 값들은 덧셈 연산(HE.Addition 연산)을 통해 최종 결과값(R_A)을 반환한다. 구체적으로, 서버(300)의 데이터 연산부(320)에 포함된 덧셈 연산부는 복수의 입력들(비교 연산부의 출력들을 의미할 수 있음)에 대한 덧셈 연산을 수행하고 연산 결과를 출력할 수 있다. 최종 결과값은 0부터 입력 측정 요소 개수만큼 표현될 수 있으며, 결과값이 $Enc(0)$ 이면 위험 요소가 존재하지 않음을, 결과값이 $Enc(6)$ 이면, 위험 요소가 6개 존재함을 알 수 있다.

- [0081] 결괏값은 서버(300)의 데이터 송신부(330)에 의해 사용자 단말(100)로 송신될 수 있다.
- [0083] 사술한 본 발명은 기존의 동형암호 질병 진단 프로세스와 차이점을 갖는다.
- [0084] 기존 동형암호들은 비교 연산(Comparison)이 없기 때문에, 어느 두 요소를 입력값으로 넣었을 때 크거나 작음의 비교가 어렵다. 또한, GSW scheme을 이용하여 비교 연산을 생성할 수는 있지만, Enc(0) 또는 Enc(1) 처럼 비교 결과 크거나 작음으로 표현되는 것이 아니라, 입력값 그대로의 결괏값이 도출된다.
- [0085] 또한, 기존의 동형암호들은 게이트 연산을 진행하지 않는다. 본 발명에서 사용되는 동형암호 방법은 비트 단위 암호문들 사이에 AND 게이트 연산을 비롯한 다양한 연산이 가능하다. 게이트 연산은 연산 속도가 매우 빠르기 때문에, 속도 측면에서 다른 동형암호 연산보다 더 유리하며, 따라서 빠르고 정확한 결괏값을 도출할 수 있다.
- [0087] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 컨트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.
- [0088] 소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.
- [0089] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0090] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

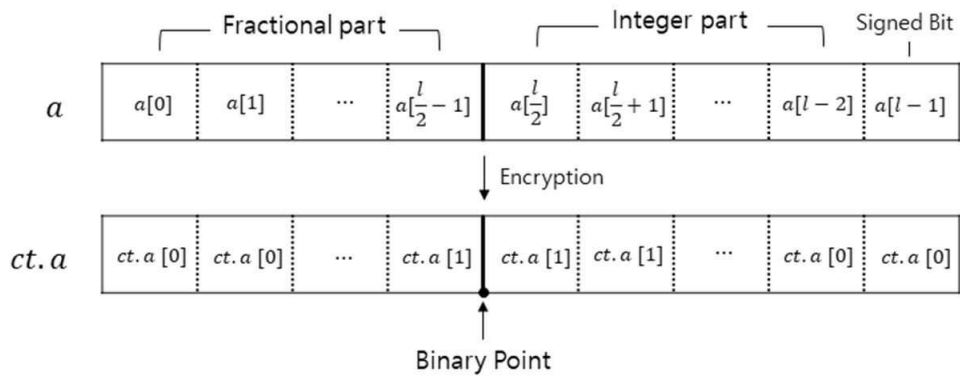
부호의 설명

[0091]

- 10 : 진단 시스템
- 100 : 사용자 단말
- 300 : 서버
- 310 : 데이터 수신부
- 320 : 데이터 연산부
- 330 : 데이터 송신부
- 340 : 저장부

도면

도면1



도면2

Algorithm 1 HE.CompareLE: Less than or equal algorithm in an encrypted state

```

1:  $ct.t_1[0] \leftarrow Enc(0)$ 
2: for  $i = 1$  to  $l - 1$  do
3:    $ct.t_2 \leftarrow HE.XNOR(ct.\hat{a}[i], ct.\hat{b}[i])$ 
4:    $ct.t_1 \leftarrow HE.MUX(ct.t_2, ct.t_1, ct.\hat{a}[i])$ 
5: end for
6:  $\hat{r} \leftarrow HE.NOT(ct.t_1)$ 
7: return  $\hat{r}$ 

```

도면3

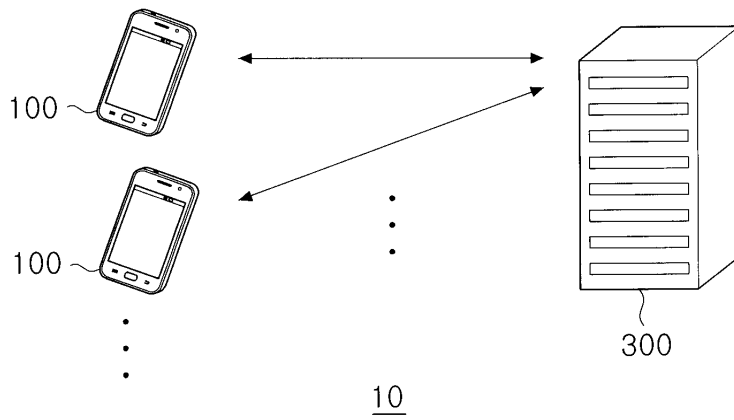
Algorithm 2 HE.Addition: Addition algorithm in an encrypted state

```

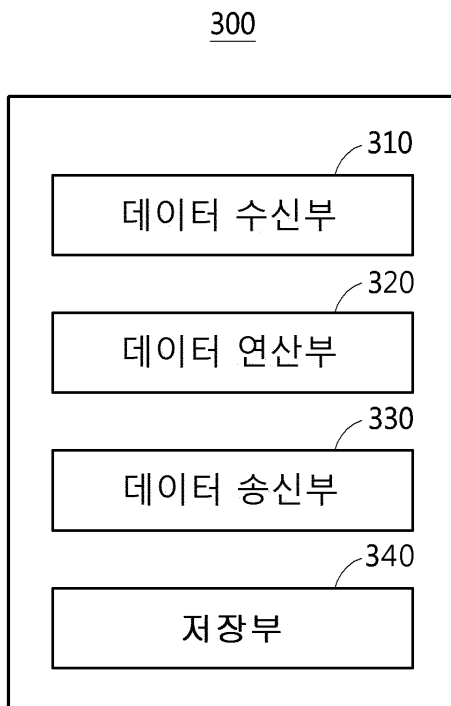
1:  $ct.\hat{c}[0] \leftarrow Enc(0)$ 
2: for  $i = 1$  to  $l - 1$  do
3:    $ct.t_1 \leftarrow HE.XOR(\hat{a}[i], \hat{b}[i])$ 
4:    $ct.t_2 \leftarrow HE.AND(\hat{a}[i], \hat{b}[i])$ 
5:    $ct.\hat{r}[i] \leftarrow HE.XOR(ct.t_1, \hat{c}[i])$ 
6:    $ct.t_1 \leftarrow HE.AND(ct.t_1, \hat{c}[i])$ 
7:    $ct.\hat{c}[i + 1] \leftarrow HE.OR(ct.t_1, ct.t_2)$ 
8: end for
9:  $ct.t_1 \leftarrow HE.XOR(ct.\hat{a}[l - 1], ct.\hat{b}[l - 1])$ 
10:  $ct.\hat{r}[l - 1] \leftarrow HE.XOR(ct.t_1, ct.\hat{c}[l - 1])$ 
11: return  $\hat{r}$ 

```

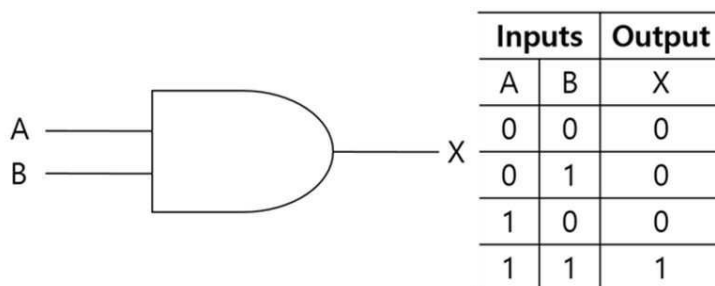
도면4



도면5



도면6



도면7

