



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0105717
(43) 공개일자 2016년09월07일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) H04L 9/00 (2006.01)
(52) CPC특허분류
H04L 63/1441 (2013.01)
H04L 63/0428 (2013.01)
(21) 출원번호 10-2016-0022540
(22) 출원일자 2016년02월25일
심사청구일자 2016년02월25일
(30) 우선권주장
1020150028004 2015년02월27일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
서민혜
서울특별시 양천구 목동중앙남로 21-15 101동 703호 (목동,에이스아파트)
이동훈
서울특별시 종로구 사직로8길 34, 1404호(내수동, 경희궁의아침3단지아파트)
(74) 대리인
김등용, 김홍석

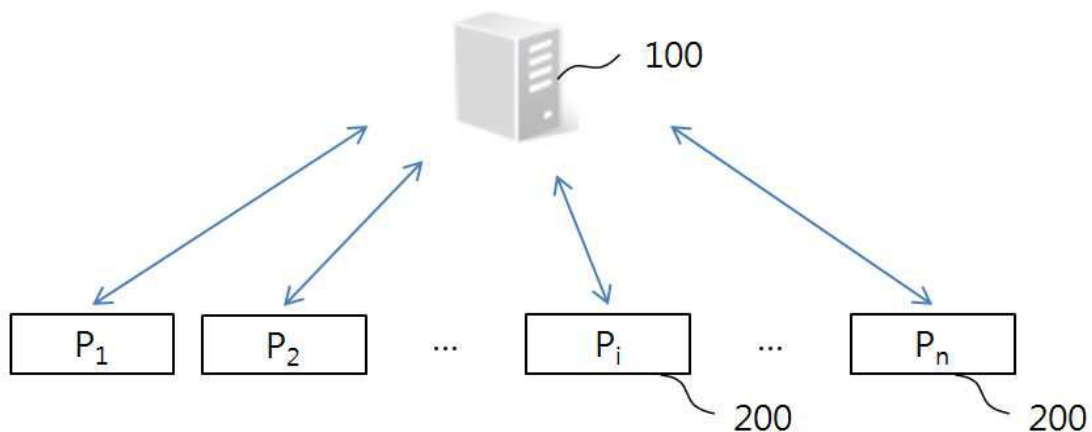
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 악의적인 사용자 탐지가 가능한 다자간 연산 방법 및 시스템

(57) 요약

악의적 사용자를 탐지하는 검증 서버 및 복수의 사용자 장치를 포함하는 다자간 연산 시스템을 이용하여, 제1 사용자 장치의 비밀값인 제1 비밀값(x_1)과 제2 사용자 장치의 비밀값인 제2 비밀값(x_2)에 대한 다자간 연산을 수행하는 방법이 개시된다. 다자간 연산 수행 방법은 각각의 사용자 장치(P_i)가 제1 사용자 장치로부터 수신한 제 (뒷면에 계속)

대표도 - 도1



10

1 비밀값(x_1)에 대한 제1 몫($x_{1,i}$) 및 제2 사용자로부터 수신한 제2 비밀값(x_2)에 대한 제2 몫($x_{2,i}$)을 이용하여, 수행하고자 하는 다자간 연산에 대한 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계, 각각의 사용자 장치(P_i)가 중간 연산 값(y_i)에 대한 제1 암호문, 커미트먼트, 및 비대화형 영지식 증명을 생성하고 암호화하여 제2 암호문을 생성하고, 상기 제2 암호문을 서버에 전송하는 단계, 서버가 제2 암호문을 복호화하고 검증하는 단계, 검증을 통과하면, 서버가 복수의 사용자 장치들로부터 수신한 제1 암호문들을 각각의 사용자 장치(P_i)에 전송하는 단계, 및 각각의 사용자 장치(P_i)가 제1 암호문들을 복호화하여 중간 연산값들을 획득하고, 이를 재건하여 연산 결과 값을 생성하는 단계를 포함한다.

(52) CPC특허분류

H04L 63/1416 (2013.01)

H04L 9/008 (2013.01)

H04L 2463/141 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2014049841

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 중견연구자지원_도약연구(전략)

연구과제명 속성 기반 공개키 시스템에 관한 연구

기 여 율 1/1

주관기관 고려대학교 산학협력단

연구기간 2014.09.01 ~ 2015.08.31

명세서

청구범위

청구항 1

악의적인 사용자를 탐지하는 서버 및 복수의 사용자 장치를 포함하는 다자간 연산 시스템을 이용하여, 상기 복수의 사용자 장치 중 제1 사용자 장치의 비밀값인 제1 비밀값(x_1)과 상기 복수의 사용자 장치 중 제2 사용자 장치의 비밀값인 제2 비밀값(x_2)에 대한 다자간 연산을 수행하는 방법에 있어서,

각각의 사용자 장치(P_i)가 상기 제1 사용자 장치로부터 수신한 상기 제1 비밀값(x_1)에 대한 제1 몫($x_{1,i}$) 및 상기 제2 사용자 장치로부터 수신한 상기 제2 비밀값(x_2)에 대한 제2 몫($x_{2,i}$)을 이용하여, 수행하고자 하는 상기 다자간 연산에 대한 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계;

상기 각각의 사용자 장치(P_i)가 상기 중간 연산 값(y_i)에 대한 제1 암호문, 상기 중간 연산 값(y_i)에 대한 커밋먼트, 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 생성하는 단계;

상기 각각의 사용자 장치(P_i)가 상기 제1 암호문, 상기 중간 연산 값(y_i)에 대한 커밋먼트, 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 암호화하여 제2 암호문을 생성하고, 상기 제2 암호문을 상기 서버에 전송하는 단계;

상기 서버가 상기 중간 연산 값(y_i)에 대한 커밋먼트를 생성하는 단계;

상기 서버가 상기 제2 암호문을 복호화하여 상기 제1 암호문, 상기 중간 연산 값(y_i)에 대한 커밋먼트, 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 생성하는 단계;

상기 서버가 생성한 상기 중간 연산 값(y_i)에 대한 커밋먼트를 이용하여, 상기 제2 암호문을 복호화하여 획득한 상기 중간 연산 값(y_i)에 대한 커밋먼트를 검증하는 단계;

상기 서버가 상기 제2 암호문을 복호화하여 획득한 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 검증하는 단계;

상기 중간 연산 값(y_i)에 대한 커밋먼트 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명에 대한 검증을 통과하면, 상기 서버가 상기 제2 암호문을 복호화하여 획득한 상기 제1 암호문들을 상기 각각의 사용자 장치(P_i)에 전송하는 단계; 및

상기 각각의 사용자 장치(P_i)가 상기 서버로부터 수신한 상기 제1 암호문들을 복호화하여 중간 연산값들을 생성하고, 상기 중간 연산값들을 재건하여 상기 제1 비밀값(x_1)과 상기 제2 비밀값(x_2)에 대한 연산 결과 값을 생성하는 단계;를 포함하는 다자간 연산 방법.

청구항 2

제1항에 있어서,

상기 다자간 연산을 수행하기 전에,

상기 각각의 사용자 장치(P_i)가 비밀값(x_i)에 대한 아래의 수학적 식 1을 만족하는 몫($x_{i,j}$)을 다른 사용자 장치들에 분배하는 단계를 더 포함하는 다자간 연산 방법.

[수학적 식 1]

$$x_i = x_{i,1} + x_{i,2} + \dots + x_{i,n}$$

청구항 3

제2항에 있어서,

상기 다자간 연산을 수행하기 전에,

상기 각각의 사용자 장치(P_i)가 상기 비밀값(x_i)에 대한 상기 몫($x_{i,j}$)의 커밋먼트를 생성하여 상기 서버에 전송하는 단계를 더 포함하고,

상기 서버가 상기 중간 연산 값(y_i)에 대한 커밋먼트를 생성하는 단계는,

상기 서버가 상기 다자간 연산을 수행하기 전에 상기 각각의 사용자 장치(P_i)로부터 수신한 상기 몫($x_{i,j}$)의 커밋먼트를 이용하여 상기 중간 연산 값(y_i)에 대한 커밋먼트를 생성하는, 다자간 연산 방법.

청구항 4

제1항에 있어서,

상기 다자간 연산에 대한 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계에서,

상기 다자간 연산이 상기 제1 비밀값(x_1) 및 상기 제2 비밀값(x_2)에 대한 덧셈 연산인 경우, 상기 사용자 장치(P_i)의 중간 연산값(y_i)은 상기 제1 몫($x_{1,i}$) 및 상기 제2 몫($x_{2,i}$)을 더한 값인 다자간 연산 방법.

청구항 5

제1항에 있어서,

상기 다자간 연산에 대한 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계에서,

상기 다자간 연산이 상기 제1 비밀값(x_1) 및 상기 제2 비밀값(x_2)에 대한 곱셈 연산인 경우, 아래의 수학적 식 2를 만족하는 곱셈 트리플(a_i, b_i, c_i)을 이용하여 곱셈 연산을 수행하는 다자간 연산 방법.

[수학적 식 2]

$$a = \sum_{i=1}^n a_i, \quad b = \sum_{i=1}^n b_i, \quad c = \sum_{i=1}^n c_i, \quad c = ab$$

청구항 6

제5항에 있어서,

상기 다자간 연산에 대한 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계는,

상기 각각의 사용자 장치(P_i)가 제1 중간값($\gamma_i = (x_{1,i} - a_i)$) 및 제2 중간값($\delta_i = (x_{2,i} - b_i)$)을 계산하여 공개하는 과정,

상기 복수의 사용자 장치들에 의해 공개된 상기 제1 중간값들을 이용하여 제1 중간 재건값

($\gamma = \sum_{i=1}^n \gamma_i = (x_1 - a)$) 을 재건하는 과정,

상기 복수의 사용자 장치들에 의해 공개된 상기 제2 중간값들을 이용하여 상기 제2 중간 재건값

($\delta = \sum_{i=1}^n \delta_i = (x_2 - b)$)을 재건하는 과정, 및

아래의 수학식 3을 이용하여 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 계산하는 과정을 포함하는 다자간 연산 방법.

[수학식 3]

$$y_i = c_i + (\delta \times a_i) + (\gamma \times b_i) + \gamma \delta$$

청구항 7

악의적인 사용자를 탐지하는 서버 및 복수의 사용자 장치를 포함하는 다자간 연산 시스템에 있어서,

상기 사용자 장치는,

다른 사용자 장치들로부터 분배받은 비밀값에 대한 몫을 이용하여 중간연산값을 생성하는 연산부,

상기 중간연산값에 대한 커미트먼트 및 상기 중간연산값에 대한 비대화형 영지식 증명을 생성하는 검증값 생성부,

상기 중간연산값을 암호화하여 제1 암호문을 생성하거나, 상기 제1 암호문, 상기 중간연산값에 대한 커미트먼트 및 상기 중간연산값에 대한 비대화형 영지식 증명을 암호화하여 제2 암호문을 생성하는 암호화부, 및

상기 서버로부터 수신한 다른 사용자 장치들의 중간연산값을 이용하여 연산결과값을 재건하는 재건부,를 포함하고,

상기 서버는,

상기 사용자 장치로부터 수신한 제2 암호문을 복호화하여 상기 제1 암호문, 상기 중간연산값에 대한 커미트먼트 및 상기 중간연산값에 대한 비대화형 영지식 증명을 획득하는 복호화부, 및

상기 중간연산값에 대한 커미트먼트 및 상기 중간연산값에 대한 비대화형 영지식 증명을 이용하여 악의적인 사용자인지 여부를 검증하는 검증부를 포함하는,

다자간 연산 시스템.

발명의 설명

기술 분야

[0001] 본 발명의 개념에 따른 실시 예는 다자간 연산(Multi-party computation)을 수행하는 과정에서 악의적인 사용자를 탐지할 수 있는 다자간 연산 방법 및 시스템에 관한 것으로, 특히 동형(Homomorphic) 성질을 가지는 커미트먼트(commitment)와 비대화형 영지식 증명(Non-interactive zero knowledge)을 응용하여 연산 값을 검증하고 악의적인 사용자를 탐지하는 다자간 연산 방법에 관한 것이다.

배경 기술

[0002] 다자간 연산을 수행하는 과정에서 악의적인 사용자가 존재하는 경우 연산의 결과로 잘못된 값이 출력될 수 있기 때문에 연산에 참여하는 사용자들의 비밀 값을 입력으로 하여 주어진 연산이 제대로 수행되었는지 검증하는 기술이 필요하다.

[0003] 또한, 연산에 참여하는 사용자들이 제대로 연산을 수행하였는지 검증하는 과정이 사용자들의 중간 연산 값을 공개한 후에 수행되면, 악의적인 사용자가 정당하지 않은 값을 공개함으로써 악의적인 사용자만이 정당한 결과 값을 얻게되는 문제점이 발생한다. 이 경우 검증 과정에서 악의적인 사용자가 탐지된다고 하더라도 이미 악의적인 사용자만 정당한 결과 값을 얻은 후이기 때문에, 연산의 결과 값이 실시간성을 가지거나 경제적 과급력이 큰 경우 이를 방지할 수 있는 기술이 필요하다.

발명의 내용

해결하려는 과제

[0004] 본 발명이 이루고자 하는 기술적인 과제는 동형(Homomorphic) 성질을 가지는 커미트먼트(commitment)와 비대화형 영지식 증명(Non-interactive zero knowledge)을 응용하여 연산 값을 검증하고 악의적인 사용자를 탐지하는 다자간 연산 방법 및 시스템을 제공하는 것이다.

과제의 해결 수단

[0005] 본 발명의 실시 예에 따른 악의적인 사용자를 탐지하는 서버 및 복수의 사용자 장치를 포함하는 다자간 연산 시스템을 이용하여, 제1 사용자 장치의 비밀값인 제1 비밀값(x_1)과 제2 사용자 장치의 비밀값인 제2 비밀값(x_2)에 대한 다자간 연산을 수행하는 방법은, 각각의 사용자 장치(P_i)가 상기 제1 사용자 장치로부터 수신한 상기 제1 비밀값(x_1)에 대한 제1 몫($x_{1,i}$) 및 상기 제2 사용자로부터 수신한 상기 제2 비밀값(x_2)에 대한 제2 몫($x_{2,i}$)을 이용하여, 수행하고자 하는 상기 다자간 연산에 대한 상기 사용자 장치(P_i)의 중간 연산값(y_i)을 생성하는 단계, 상기 각각의 사용자 장치(P_i)가 상기 중간 연산 값(y_i)에 대한 제1 암호문, 상기 중간 연산 값(y_i)에 대한 커미트먼트, 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 생성하는 단계, 상기 각각의 사용자 장치(P_i)가 상기 제1 암호문, 상기 중간 연산 값(y_i)에 대한 커미트먼트, 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 암호화하여 제2 암호문을 생성하고, 상기 제2 암호문을 상기 서버에 전송하는 단계, 상기 서버가 상기 몫($x_{i,j}$)에 대한 커미트먼트를 이용하여 상기 중간 연산 값(y_i)에 대한 커미트먼트를 생성하는 단계, 상기 서버가 상기 제2 암호문을 복호화하는 단계, 상기 서버가 생성한 상기 중간 연산 값(y_i)에 대한 커미트먼트를 이용하여, 상기 제2 암호문을 복호화하여 획득한 상기 중간 연산 값(y_i)에 대한 커미트먼트를 검증하는 단계, 상기 서버가 상기 제2 암호문을 복호화하여 획득한 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명을 검증하는 단계, 상기 중간 연산 값(y_i)에 대한 커미트먼트 및 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명에 대한 검증을 통과하면, 상기 서버가 상기 복수의 사용자 장치들로부터 수신한 상기 제1 암호문들을 상기 각각의 사용자 장치(P_i)에 전송하는 단계, 및 상기 각각의 사용자 장치(P_i)가 상기 서버로부터 수신한 상기 제1 암호문들을 복호화하여 중간 연산값들을 생성하고, 상기 중간 연산값들을 재건하여 상기 제1 비밀값(x_1)과 상기 제2 비밀값(x_2)에 대한 연산 결과 값을 생성하는 단계를 포함한다.

발명의 효과

[0006] 본 발명의 실시 예에 따른 다자간 연산 방법에 의할 경우, 다자간 연산에 참여하는 사용자들의 중간 연산 값을 서버가 검증함으로써 사용자들 각각의 연산량이 증가하는 것을 방지할 수 있으며, 서버는 사용자들의 비밀 값

뿐만 아니라, 중간 연산 값 및 연산의 결과 값에 대한 정보를 얻을 수 없고 오직 검증에 대한 결과만 알 수 있게 된다.

[0007] 또한, 악의적인 사용자가 정당하지 않은 값을 서버로 전송하는 경우 검증 과정에서 이를 탐지할 수 있고, 탐지된 정보를 정당한 사용자에게 전송함으로써 정당한 사용자들끼리 안전한 다자간 연산을 수행할 수 있다.

[0008] 또한, 서버가 사용자들로부터 중간 연산 값에 대한 암호문을 전송받고, 모든 중간 연산 값에 대한 검증이 통과해야지만 사용자들에게 암호문을 전송하기 때문에 연산에 참여하는 모든 사용자가 정당한 결과 값을 얻을 수 있으며, 악의적인 사용자만이 정당한 결과 값을 얻는 것을 방지할 수 있다.

도면의 간단한 설명

[0009] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 본 발명의 일 실시예에 따른 악의적인 사용자 탐지가 가능한 다자간 연산 시스템의 개략도이다.

도 2는 도 1에 도시된 서버의 기능 블록도이다.

도 3은 도 1에 도시된 사용자 장치의 기능 블록도이다.

도 4는 도 1에 도시된 다자간 연산 시스템을 이용한 악의적인 사용자 탐지가 가능한 다자간 연산 방법을 설명하기 위한 흐름도이다.

도 5는 도 4의 단계 S200 및 S300을 보다 세부적으로 도시한 도면이고, 도 6은 도 4의 단계 S400을 보다 세부적으로 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0010] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시 예들에 대해서 특정한 구조적 또는 기능적 설명은 단지 본 발명의 개념에 따른 실시 예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시 예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시 예들에 한정되지 않는다.

[0011] 본 발명의 개념에 따른 실시 예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시 예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시 예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0012] 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.

[0013] 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 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.

[0014] 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 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.

[0015] 우선, 본 발명에 대한 설명에 앞서, 본 발명의 배경이 되는 기술에 대해 설명하면 다음과 같다.

[0016] 본 발명에서는 비밀 분배 기법을 이용한 안전한 다자간 연산(Secure Multi-party Computation) 기법을 사용한다. 안전한 다자간 연산은 다수의 사용자가 각자의 비밀 값을 입력으로 하여 주어진 연산을 함께 계산함으로써 원하는 결과 값을 얻을 수 있는 기술이다. 이때, 사용자들의 비밀 값에 대한 정보는 노출되지 않아야 한

다. 해당 연구는 30여년간 이론 및 응용분야에 걸쳐 널리 연구가 진행되었고, 혼돈 회로(Garbled circuit), 동형 암호(Homomorphic encryption), 비밀 분배 기법(Secret sharing scheme) 등 다양한 프리미티브를 이용하여 설계할 수 있다.

[0017] 본 발명에서는 비밀 분배 기법을 이용한 안전한 다자간 연산 기술을 사용한다. 각각의 사용자가 자신의 비밀 값을 분할하여 몫(share)들을 생성하여 다른 사용자들에게 전송하고, 사용자들은 전송받은 몫들을 이용하여 주어진 연산을 수행하고, 중간 연산 값을 생성하여 공개한다. 다음, 상기 공개된 중간 연산 값을 재건(reconstruction)하여 사용자들의 비밀 값을 입력으로 하는 연산의 결과 값을 획득하는 기술이다. 이하, 본 발명에서는 사용하는 덧셈 및 곱셈을 이용한 비밀 분배 기법을 사용에 대하여 상세하게 설명한다.

[0018] 먼저, 덧셈에 대한 연산을 수행하는 경우 사용자들 간에 통신이 요구되지 아니한다. 사용자 P_i 의 비밀 값 x_i 에 대하여 아래의 식을 만족하는 $x_{i,j}$ 들이 몫이 된다.

$$x_i = x_{i,1} + x_{i,2} + \dots + x_{i,n}$$

[0020] 사용자 P_i 는 사용자 P_i 의 비밀 값 x_i 에 대한 각각의 몫 $x_{i,j}$ 를 사용자 P_j 에게 전송한다. 모든 사용자가 전송을 완료하면, 사용자 P_i 는 몫 $(x_{1,i}, x_{2,i}, \dots, x_{n,i})$ 를 갖게 된다.

[0021] 사용자들이 덧셈에 대한 연산 $(x_1 + x_2)$ 을 수행하고 싶다면, 각 사용자 P_i 는 자신이 가지고 있는 몫을 이용하여 덧셈에 대한 중간 연산값 $y_i = (x_{1,i} + x_{2,i})$ 를 계산한 후 공개한다. 각각의 사용자 P_i 에 의하여 공개된 덧셈에 대한 중간 연산 값 y_i (연산의 결과 값 y 에 대한 사용자 P_i 의 몫)들을 이용하여 연산에 대한 결과 값(y)을 아래와 같이 구할 수 있다.

$$\begin{aligned} y &= y_1 + \dots + y_n \\ &= (x_{1,1} + x_{2,1}) + \dots + (x_{1,n} + x_{2,n}) \\ &= (x_{1,1} + \dots + x_{1,n}) + (x_{2,1} + \dots + x_{2,n}) \\ &= (x_1 + x_2) \end{aligned}$$

[0022] 다음, 곱셈에 대한 연산을 수행하는 경우에는 사용자들 간에 통신이 필요하다. 먼저, 연산에 참여하는 각 사용자 P_i 는 연산을 수행하기 전에 아래의 수확식을 이용하여 곱셈 트리플(Multiplicative triples, (a_i, b_i, c_i))을 연산한다.

$$a = \sum_{i=1}^n a_i, \quad b = \sum_{i=1}^n b_i, \quad c = \sum_{i=1}^n c_i \quad (\text{이때, } c = ab)$$

[0025] 곱셈 트리플을 이용하면 사용자 간의 통신량을 최소화하면서 곱셈 연산을 수행할 수 있다. 사용자들이 곱셈에 대한 연산 $(x_1 \times x_2)$ 을 수행하고 싶다면, 각 사용자는 자신이 가지고 있는 몫을 이용하여 아래와 같이 중간 연산 값 y_i (연산의 결과 값 y 에 대한 사용자 P_i 의 몫)를 계산한 후 공개한다.

$$y_i = c_i + (\delta \times a_i) + (\gamma \times b_i) + \gamma \delta$$

[0027] 이때, γ (제1 중간 재건값) 및 δ (제2 중간 재건값)는 각 사용자 P_i 가 자신의 몫을 이용하여 계산하고 다른 사용자에게 공개한 제1 중간값($\gamma_i = (x_{1,i} - a_i)$) 및 제2 중간값($\delta_i = (x_{2,i} - b_i)$)을 이용하여 아래와 같이 재건한 값이다.

$$\gamma = \sum_{i=1}^n \gamma_i = (x_1 - a)$$

$$\delta = \sum_{i=1}^n \delta_i = (x_2 - b)$$

다음, 각각의 사용자에게 의하여 공개된 곱셈에 대한 중간 연산 값 y_i 들을 이용하여 곱셈 연산에 대한 결과 값 (y)을 아래와 같이 구할 수 있다.

$$\begin{aligned} y &= \sum_{i=1}^n y_i = c + \delta a + \gamma b + \gamma \delta \\ &= ab + (x_2 - b)a + (x_1 - a)b + (x_1 - a)(x_2 - b) \\ &= (x_1 \times x_2) \end{aligned}$$

다음, 본 발명에서 사용하는 동형 커미트먼트(Homomorphic Commitment)에 대하여 설명한다. 커미트먼트(commitment)는 한 사용자가 자신의 비밀 값에 대해서 다른 사용자에게 약속(commit)할 때 사용되는 기술이다. 나중에 커미트먼트를 통해서 실제로 그 비밀 값이 맞음을 검증할 수 있다. 커미트먼트 기술은 다음의 두 가지 성질을 만족해야 한다. 첫 번째 성질은 바인딩(Binding)으로, 커미트먼트가 한번 생성되고 나면 이후에 사용된 비밀 값을 바꿀 수 없어야 한다는 것이다. 두 번째 성질은 하이딩(Hiding)으로, 다른 사용자들이 커미트먼트를 보고 생성될 때 사용된 비밀 값을 알 수 없어야 한다는 것이다.

본 발명에서는 비밀 값 m 에 대한 커미트먼트 $Com(r, m) = g^r v^m$ 을 이용하며, 랜덤 값 r_1, r_2 및 비밀 값 m_1, m_2 에 대하여 아래의 등식을 만족하면 동형 커미트먼트라고 한다.

$$Com(r_1, m_1) \cdot Com(r_2, m_2) = Com(r_1 + r_2, m_1 + m_2)$$

다음, 본 발명에서 사용하는 비대화형 영지식 증명(Non-interactive Zero Knowledge)에 대하여 설명한다. 비대화형 영지식 증명은 증명자(Prover)와 검증자(Verifier) 간에 통신 없이 영지식 증명을 수행하는 기술이다. 1980년대 후반에 비대화형 영지식 증명에 대한 개념이 등장했으며, 암호학적 프리미티브로써 CCA2에 안전한 암호시스템, 전자 서명 등 다양한 암호 프로토콜을 설계하는데 사용되었다. NP 언어로 표현되는 시스템에 대한 비대화형 영지식 증명 기술을 설계하는 것은 난제로 여겨졌으나, 2006년 Groth 등이 이를 해결함으로써 모든 NP 언어에 대한 비대화형 영지식 증명의 구현이 가능해졌다. 본 발명에서는 비대화형 영지식 증명 기술을 이용하여 악의적인 사용자만 연산에 대한 정당한 결과 값을 얻는 것을 방지한다.

다음, 본 발명에서 사용하는 공개키 기반 브로드 캐스트 암호(Public-key Broadcast Encryption) 기법에 대하여 설명한다. 브로드캐스트 암호는 수신자 집합에 대한 암호문을 생성하여 해당 집합에 포함된 사용자만 자신의 비밀키를 이용하여 복호화할 수 있는 기술이다. 공개키 기반 브로드캐스트 암호 기법은 시스템의 전체 사용자에게 대한 공개키가 생성되며, 송신자(Sender)는 공개키와 수신자 집합을 이용하여 메시지를 암호화하는 기법이다. 해당 암호문은 수신자 집합에 포함된 사용자만이 자신의 비밀키를 이용하여 복호화할 수 있으며, 수신자 집합에 포함되지 않은 사용자는 암호문을 통해 메시지에 대한 어떠한 정보도 얻을 수 없다.

공개키 기반 브로드캐스트 암호 기법은 아래와 같이 구성된다.

① 셋업(Setup): 전체 사용자 수 n 을 입력받아 공개키 PK 와 각 사용자의 비밀키 $d_1, \dots, d_n$ 를 출력하는 알고리즘이다.

② 암호화(Encrypt): 수신자 집합 S 와 공개키 PK 를 입력받아 (Hdr, K) 를 출력하는 알고리즘이다. Hdr 는 헤더, K 는 메시지 암호화에 사용되는 키로, 송신자는 전송하고자 하는 메시지 M 을 K 를 이용해 대칭키 암호화하여 C_M 을 생성한다. 송신자는 (S, Hdr, C_M) 을 수신자 집합에 포함된 사용자에게 브로드캐스트한

다.

- [0040] ③ 복호화(Decrypt): 수신자 집합 S , 헤더 Hdr , 공개키 PK , 사용자의 비밀키 d_i 를 입력 받고, 사용자 P_i 가 수신자 집합 S 에 포함되어 있으면 메시지 암호화에 사용된 키 K 를 출력하는 알고리즘이다. 수신자는 출력된 키 K 를 이용하여 암호문 C_M 을 복호화하고 메시지 M 을 얻는다.
- [0041] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다.
- [0042] 먼저, 도 1 내지 도 3을 참조하여, 본 발명의 일 실시예에 따른 악의적인 사용자 탐지가 가능한 다자간 연산 시스템에 대해 상술한다.
- [0043] 도 1은 본 발명의 일 실시예에 따른 다자간 연산 시스템(10)을 도시한다. 도 1을 참조하면, 본 발명에 따른 다자간 연산 시스템(10)은 중간 연산 값의 검증을 위한 서버(100), 및 다자간 연산에 참여하는 복수 개의 사용자 장치(200)를 포함한다.
- [0044] 다자간 연산에 참여하는 사용자 장치(200)들은 각각의 중간 연산 값을 연산하고, 서버(100)는 사용자 장치(200)들이 생성한 중간 연산 값이 사용자 장치(200)의 비밀 값을 이용하여 계산된 정당한 값인지 여부를 검증한다.
- [0045] 서버(100)는 중간 연산 값 검증을 통하여, 정당하지 않은 값을 전송하는 악의적인 사용자(장치)를 탐지하고, 탐지된 악의적인 사용자(장치)에 대한 정보를 사용자 장치(200)들에게 전송한다.
- [0046] 도 2는 도 1에 도시된 검증 서버(100)의 기능 블록도이다.
- [0047] 도 1과 도 2를 참조하면, 상기 검증 서버(100)는 복호화부(110), 검증부(130), 통신부(170), 저장부(180), 및 제어부(190)를 포함한다.
- [0048] 본 명세서에서 사용되는 '~부'라 함은 본 발명의 기술적 사상을 수행하기 위한 하드웨어 및 상기 하드웨어를 구동하기 위한 소프트웨어의 기능적, 구조적 결합을 의미할 수 있다. 예컨대, 상기 '~부'는 소정의 코드와 상기 소정의 코드가 수행되기 위한 하드웨어 리소스의 논리적인 단위를 의미할 수 있으며, 반드시 물리적으로 연결된 코드를 의미하거나 한 종류의 하드웨어를 의미하는 것은 아니다.
- [0049] 서버(100)의 복호화부(110)는 제어부(190)의 제어 하에, 사용자 장치(P_i)로부터 수신한 암호문을 서버의 공개키를 이용하여 복호화하여, 사용자 장치(P_i)의 중간 연산 값(y_i)에 대한 암호문($BE.Enc_{PK}(y_i)$) 및 상기 중간 연산 값(y_i)을 검증하기 위한 검증값들을 획득할 수 있다. 구체적으로 상기 검증값은 상기 중간 연산 값(y_i)에 대한 커밋먼트($Com(y_i)$) 및 상기 중간 연산 값(y_i)에 대한 암호문($BE.Enc_{PK}(y_i)$)과 상기 중간 연산 값(y_i)에 대한 커밋먼트($Com(y_i)$)에 같은 값이 사용되었음을 증명하는 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명($NIZK(y_i)$)을 포함할 수 있다.
- [0050] 서버(100)의 검증부(130)는 제어부(190)의 제어 하에, 사용자 장치(200, P_i)들이 생성한 중간 연산 값(y_i)들이 사용자의 비밀값(x_i)을 이용하여 계산된 정당한 값인지 여부를 검증한다. 검증부(130)는 $Com(y_i)$ 검증부 및 $NIZK(y_i)$ 검증부를 포함할 수 있으며, $Com(y_i)$ 검증부는 서버(100)의 $Com(y_i)$ 생성부에서 생성한 $Com(y_i)$ 와 사용자 장치(200)로부터 수신한 $Com(y_i)$ 가 동일한지 여부를 검증한다. $NIZK(y_i)$ 검증부는 비대화형 영지식 증명을 검증할 수 있다.
- [0051] 서버(100)의 통신부(170)는 제어부(190)의 제어 하에, 사용자 장치(200)들과 유선 또는 무선 통신망을 통하여 통신을 수행할 수 있다.
- [0052] 서버(100)의 저장부(180)는 프로그램 저장부와 데이터 저장부를 포함할 수 있다. 상기 프로그램 저장부는 인증 서버(100)의 동작을 제어하기 위한 프로그램들이 저장될 수 있다. 상기 데이터 저장부에는 상기 프로그램들을 수행하는 과정 중에 발생하는 데이터들이 저장될 수 있다.

- [0053] 서버(100)의 제어부(190)는 서버(100)의 전반적인 동작을 제어한다. 즉, 복호화부(110), 검증부(130), 통신부(170) 및 저장부(180)의 동작을 제어할 수 있다.
- [0054] 도 3은 도 1에 도시된 사용자 장치(200)의 기능 블록도이다. 도 1과 도 3을 참조하면, 상기 사용자 장치(200)는 암호화부(210), 연산부(220), 검증값 생성부(230), 재건부(240), 통신부(270), 저장부(280), 및 제어부(290)를 포함한다.
- [0055] 암호화부(210)는 제어부(290)의 제어 하에, 사용자 장치(200, P_i)가 생성한 중간 연산 값(y_i)을 공개키(PK)로 암호화하여 제1 암호문($BE.Enc_{PK}(y_i)$)을 생성할 수 있다. 또한, 암호화부(210)는 상기 중간 연산 값(y_i)에 대한 상기 제1 암호문($BE.Enc_{PK}(y_i)$), 상기 중간 연산 값(y_i)에 대한 커미트먼트($Com(y_i)$), 및 중간 연산 값 y_i 에 대한 비대화형 영지식 증명($NIZK(y_i)$)을 서버의 공개키(pk_s)로 암호화하여 제2 암호문을 생성할 수 있다.
- [0056] 또한, 암호화부(210)는 검증이 완료되어 서버(100)로부터 수신한 다자간 연산에 참여하는 모든 사용자 장치에 대한 제1 암호문들을 복호화하여 다자간 연산에 참여하는 모든 사용자 장치에 대한 중간연산값을 획득할 수 있다.
- [0057] 사용자 장치(200)의 연산부(220)는 제어부(290)의 제어 하에, 다른 사용자 장치의 비밀값에 대하여 분배받은 몫을 이용하여, 수행하고자 하는 연산에 대한 사용자 장치(P_i)의 몫, 즉 중간연산값(y_i)를 계산한다.
- [0058] 사용자 장치(200)의 검증값 생성부(230)는 커미트먼트 생성부 및 비대화형 영지식 증명 생성부를 포함하며, 커미트먼트 생성부는 상기 중간 연산 값(y_i)에 대한 커미트먼트($Com(y_i)$)를 생성하고, 비대화형 영지식 증명 생성부는 상기 중간 연산 값(y_i)에 대한 비대화형 영지식 증명($NIZK(y_i)$)을 생성할 수 있다.
- [0059] 사용자 장치(200)의 재건부(240)는 제어부(290)의 제어 하에, 상기 암호화부(210)에서 복호화하여 획득한 모든 사용자 장치에 대한 중간연산값들을 이용하여, 다자간 연산의 결과 값을 획득할 수 있다.
- [0060] 사용자 장치(200)의 통신부(270)는 제어부(290)의 제어 하에, 서버(100) 및 사용자 장치(200)들과 유선 또는 무선 통신망을 통하여 통신을 수행할 수 있다.
- [0061] 사용자 장치(200)의 저장부(280)는 프로그램 저장부와 데이터 저장부를 포함할 수 있으며, 사용자 장치(200)의 제어부(190)는 서버(100)의 전반적인 동작을 제어한다.
- [0062] 이하, 도 4를 참조하여, 본 발명의 일 실시예에 따른 다중 요소 임계 인증 시스템을 이용하여 다중 요소 임계 인증을 수행하는 방법에 대해 상술한다.
- [0063] 도 4는 도 1에 도시된 다중 요소 임계 인증 시스템을 이용한 다중 요소 임계 인증 방법을 설명하기 위한 흐름도이다. 이하에서는 상술된 내용과 중복되는 내용에 관한 기재는 생략하기로 한다.
- [0064] 사용자 장치들은 다자간 연산을 수행하기 전에 곱셈 연산을 수행하기 위한 값(이하, '곱셈 트리플(Multiplicative triples)'라고 함)을 미리 생성하여 저장한다. 곱셈 트리플을 미리 생성함으로써, 다자간 연산 수행 중에 곱셈 연산을 더욱 효율적으로 수행할 수 있다. 또한, 공개키 기반 브로드캐스트 암호 기법의 셋업 단계를 미리 수행하여, 각각의 사용자 장치가 공개키와 각각의 사용자 장치에 대응하는 비밀키를 저장하고 있다. 본 발명에서는 동형 커미트먼트, 공개키 기반 브로드캐스트 암호 기법, 비대화형 영지식 증명을 이용한다.
- [0065] 먼저, 같이 복수의 사용자 장치(200)들은 아래와 같이 초기 단계(Init, S100)를 수행한다.
- [0066] 각각의 사용자 장치(P_i)가 자신의 비밀 값(x_i)을 분할하여 연산에 참여하는 다른 사용자 장치들에 대한 몫 $x_{i,1}, \dots, x_{i,n}$ 을 생성한다. 사용자 장치 P_i 의 비밀 값 x_i 에 대한 j번째 몫을 $x_{i,j}$ 라고 표현한다. 이때, 사용자 장치 P_i 의 비밀값 x_i 에 대한 몫($x_{i,j}$)들의 합은 비밀값 x_i 와 동일하다.
- [0067]
$$x_i = x_{i,1} + x_{i,2} + \dots + x_{i,n}$$

[0068] 다음, 각각의 묶 $x_{i,j}$ 에 대한 커미트먼트를 아래와 같이 생성한다.

[0069]
$$Com(x_{i,j}) = (g^{r_{i,j}}, g^{r_{i,j}} v^{x_{i,j}})$$

[0070] 사용자 장치(200)들은 비밀 값(x_i)에 대해 생성한 모든 묶들에 대한 커미트먼트들을 생성한 뒤 서버(100)에 전송한다.

[0071] 또한, 각각의 사용자 장치(P_i)에 대한 곱셈 트리플 $(a_{i,k}, b_{i,k}, c_{i,k})$ 을 생성하고, 아래와 같이 사용자 장치 P_i 의 곱셈 트리플에 대한 커미트먼트를 생성하여 서버(100)에 전송한다.

[0072]
$$\begin{aligned} Com(a_{i,k}) &= (g^{t_{i,k,1}}, g^{t_{i,k,1}} v^{a_{i,k}}), \\ Com(b_{i,k}) &= (g^{t_{i,k,2}}, g^{t_{i,k,2}} v^{b_{i,k}}), \\ Com(c_{i,k}) &= (g^{t_{i,k,3}}, g^{t_{i,k,3}} v^{c_{i,k}}) \end{aligned}$$

[0073] 또한, 각각의 사용자 장치(P_i)는 사용자 장치(P_i)가 생성한 묶 $x_{i,j}$ 를 다른 사용자 장치(P_j)에 전송한다. 즉, 사용자 장치 P_i 는 자신의 비밀값 x_i 에 대한 사용자 장치 P_j 의 묶인 $x_{i,j}$ 를 사용자 장치 P_j 에 전송한다. 또한, 사용자 장치 P_i 는 사용자 장치 P_j 로부터 사용자 장치 P_j 의 비밀값 x_j 에 대한 사용자 장치 P_i 의 묶인 $x_{j,i}$ 를 수신한다.

[0074] 비밀값에 대한 묶의 분배가 완료되면, 사용자 장치 P_i 는 묶 $(x_{1,i}, x_{2,i}, \dots, x_{n,i})$ 을 가지게 된다.

[0075] 다음, 사용자 장치(200)는 아래와 같이 연산 단계(Compute, S200)를 수행한다. 도 5는 도 4의 단계 S200 및 S300을 보다 세부적으로 도시한 도면이다. 도 4 내지 도 5를 참조하면, 각 사용자 장치(P_i)는 초기 단계(S100)에서 전송 받은 묶 $x_{j,i}$ 을 이용하여 덧셈 또는 곱셈에 대한 연산을 수행한다.

[0076] 구체적으로, 제1 사용자 장치의 비밀값(x_1)과 제2 사용자 장치의 비밀값(x_2)의 덧셈연산 $(x_1 + x_2)$ 를 수행하기 위해서 각 사용자 장치(P_i)는 중간 연산값 $(x_{1,i} + x_{2,i})$ 를 계산한다. $(x_{1,i} + x_{2,i})$ 는 덧셈 연산 $(x_1 + x_2)$ 에 대한 사용자 장치(P_i)의 묶(y_i)이 된다.

[0077] 또한, 제1 사용자 장치의 비밀값(x_1)과 제2 사용자 장치의 비밀값(x_2)의 곱셈연산 $(x_1 \times x_2)$ 를 수행하기 위해서 각 사용자 장치(P_i)는 곱셈 트리플을 이용하여 아래와 같이 계산한다.

[0078] 1) $\gamma_i = (x_{1,i} - a_i)$, $\delta_i = (x_{2,i} - b_i)$ 를 계산한다.

[0079] 2) 모든 사용자 장치들이 γ_i , δ_i 를 공개하고, 아래의 식을 이용하여 γ , δ 를 재건한다.

[0080]
$$\gamma = \sum_{i=1}^n \gamma_i = (x_1 - a) \quad \delta = \sum_{i=1}^n \delta_i = (x_2 - b)$$

[0081] 3) $(x_1 \times x_2)$ 에 대한 사용자 장치(P_i)의 묶인 τ_i (또는 y_i)를 아래와 같이 계산한다.

[0082]
$$\tau_i = c_{i,k} + (\delta \times a_{i,k}) + (\gamma \times b_{i,k}) + \gamma \delta$$

[0083] τ_i (또는 y_i)는 곱셈 연산 $(x_1 \times x_2)$ 에 대한 사용자 장치(P_i)의 몫(또는 '중간 연산값'이라고도 함)이 된다.

[0084] 다음, 아래와 같이 전송 단계(Send, S300)를 수행한다. 도 6은 도 4의 단계 S400을 보다 세부적으로 도시한 도면이다. 도 4 및 도 6을 참조하면, 각 사용자 장치(200, P_i)는 검증에 필요한 값을 생성한 후 서버(100)로 전송한다.

[0085] 구체적으로, 각각의 사용자 장치(P_i)는 공개키 기반 브로드캐스트 암호 기법을 이용하여 연산 단계에서 생성된 중간 연산 값 y_i 를 공개키(PK)로 암호화한 제1 암호문($BE.Enc_{PK}(y_i)$), 중간 연산 값 y_i 에 대한 커미트먼트($Com(y_i)$), 및 제1 암호문($BE.Enc_{PK}(y_i)$)과 커미트먼트($Com(y_i)$)에 같은 값이 사용되었음을 증명하는 중간 연산 값 y_i 에 대한 비대화형 영지식 증명($NIZK(y_i)$)을 생성한다.

[0086] 다음, 각각의 사용자 장치(P_i)는 서버(100)의 공개키(pk_s)로 제1 암호문($BE.Enc_{PK}(y_i)$), 커미트먼트($Com(y_i)$), 및 비대화형 영지식 증명 값 ($NIZK(y_i)$)을 아래와 같이 암호화하여 제2 암호문을 생성한 후 서버(100)로 전송한다.

[0087] $Enc_{pk_s}(Com(y_i), BE.Enc(y_i), NIZK(y_i))$

[0088] 다음, 서버(100)는 검증 단계(Verify, S400)를 수행한다. 즉, 서버(100)는 전송 단계(S300)에서 각 사용자 장치(P_i , 200)에게 전송 받은 제2 암호문을 복호화하고, 연산 단계(S200)에서 연산을 수행한 것과 동일한 방법으로 검증을 수행한다. 검증 수행 시, 초기 단계(S100)에서 사용자 장치(200)들로부터 전송받은 커미트먼트들을 이용한다.

[0089] 구체적으로, 덧셈연산에 있어서 사용자 P_i 의 중간 연산값에 대한 검증 방법은 아래와 같다. 제1 사용자 장치 P_1 의 비밀값 x_1 (이하 '제1 비밀값'이라고 함)과 제2 사용자 장치 P_2 의 비밀값 x_2 (이하 '제2 비밀값'이라고 함)의 덧셈연산에 대한 사용자 장치 P_i 의 몫, $(x_{1,i} + x_{2,i})$ 에 대한 검증을 수행하기 위해 서버(100)는 초기 단계(S100)에서 제1 사용자 장치 P_1 로부터 전송받은 제1 커미트먼트($Com(x_{1,i}) = (C_{1,i}^1, C_{1,i}^2)$) 및 제2 사용자 장치 P_2 로부터 전송받은 제2 커미트먼트($Com(x_{2,i}) = (C_{2,i}^1, C_{2,i}^2)$)를 이용한다. 즉, 서버(100)는 아래의 식을 계산하여 덧셈연산에 대한 사용자 P_i 의 중간 연산값 $(x_{1,i} + x_{2,i})$ 에 대한 커미트먼트 $Com(y_i)$ 를 생성한다.

[0090]
$$(C_{1,i}^2 \cdot C_{2,i}^2) \cdot (C_{1,i}^1 \cdot C_{2,i}^1)^{-1} = v^{(x_{1,i} + x_{2,i})}$$

[0091] 다음, 서버(100)는 덧셈 연산에 대한 커미트먼트를 이용하여, 전송 단계(S300)에서 각 사용자에게 전송받은 중간 연산 값 y_i 에 대한 커미트먼트($Com(y_i)$)의 검증을 수행한다.

[0092] 이를 통과하면 비대화형 영지식 증명을 수행한다. 즉, $NIZK(y_i)$ 의 검증을 수행한다.

[0093] 또한, 곱셈연산에 있어서 사용자 P_i 의 중간 연산값 $(x_{1,i} \times x_{2,i})$ 에 대한 검증 방법은 아래와 같다. 제1 비밀값(x_1)과 제2 비밀값(x_2)의 곱셈연산에 대한 사용자 장치 P_i 의 몫, $(x_{1,i} \times x_{2,i})$ 에 대한 검증을 수행하기 위해 서버(100)는 초기 단계(S100)에서 제1 사용자 장치 P_1 로부터 전송받은 제1 커미트먼트, 제2 사용자 장치 P_2 로부터 전송받은 제2 커미트먼트, 및 곱셈트리플에 대한 커미트먼트를 이용한다.

- [0094] 1) 서버(100)는 $Com(x_{1,i}) = (C_{1,i}^1, C_{1,i}^2)$, $Com(a_{i,k}) = (A_{i,k}^1, A_{i,k}^2)$ 를 이용하여 연산 과정에서 공개한 $\gamma_i = (x_{1,i} - a_{i,k})$ 가 정당한지 검증한다. 즉, 아래의 식이 성립하는지 확인함으로써 γ_i 를 검증한다.
- $$C_{1,i}^2 \cdot (A_{i,k}^2)^{-1} \cdot (C_{1,i}^1)^{-1} \cdot A_{i,k}^1 = v^{\gamma_i}$$
- [0095]
- [0096] 2) 서버(100)는 $Com(x_{2,i}) = (C_{2,i}^1, C_{2,i}^2)$, $Com(b_{i,k}) = (B_{i,k}^1, B_{i,k}^2)$ 를 이용하여 연산 과정에서 공개한 $\delta_i = (x_{2,i} - b_{i,k})$ 가 정당한지 검증한다. 즉, 아래의 식이 성립하는지 확인함으로써 δ_i 를 검증한다.
- $$C_{2,i}^2 \cdot (B_{i,k}^2)^{-1} \cdot (C_{2,i}^1)^{-1} \cdot B_{i,k}^1 = v^{\delta_i}$$
- [0097]
- [0098] 3) 서버(100)는 초기 단계(S100)에서 수신한 곱셈트리플에 대한 커미트먼트 $Com(a_{i,k}) = (A_{i,k}^1, A_{i,k}^2)$, $Com(b_{i,k}) = (B_{i,k}^1, B_{i,k}^2)$, 및 $Com(c_{i,k}) = (C_{i,k}^1, C_{i,k}^2)$ 를 이용하여, 사용자 장치 P_i 의 중간연산값에 대한 커미트먼트($Com(y_i)$)를 생성한다. 아래의 식을 계산하여 $(x_{1,i} \times x_{2,i})$ 에 대한 커미트먼트를 생성할 수 있다.
- $$(A_{i,k}^2)^\delta \cdot (B_{i,k}^2)^\gamma \cdot v^{\delta\gamma} \cdot C_{i,k}^2 \cdot (A_{i,k}^1)^{-\delta} \cdot (B_{i,k}^1)^{-\gamma} \cdot (C_{i,k}^1)^{-1} = v^{(x_{1,i} \times x_{2,i})}$$
- [0099]
- [0100] 다음, 서버(100)는 전송 단계(S300)에서 각각의 사용자 장치(200)로부터 전송 받은 $Com(y_i)$ 를 검증하고, 이를 통과하면 $NIZK(y_i)$ 의 검증을 수행한다.
- [0101] 덧셈 연산 또는 곱셈 연산의 중간 연산 값에 대한 검증 과정에서 정당하지 않은 값이 탐지되면, 서버(100)는 정당한 사용자 장치들에게 악의적인 사용자(장치)에 대한 정보를 공지한다.
- [0102] 모든 값이 검증을 통과하면, 서버는 전송 단계(S300)에서 전송받은 모든 암호문($BE.Enc_{PK}(y_i)$)을 각각의 사용자 장치(200)에 전송한다.
- [0103] 다음, 재건 단계(Reconstruction, S500)를 수행한다. 정당한 사용자는 자신의 비밀키를 이용하여 검증 단계(S400)에서 서버(100)로부터 전송받은 모든 암호문($BE.Enc_{PK}(y_i)$)을 복호화한다. 복호화하여 얻은 중간 연산 값(y_i)들을 이용하여 연산의 결과 값(y)을 얻는다.
- [0104] 본 발명에서는 서버(100)가 각각의 사용자 장치(200)로부터 전송받은 커미트먼트($Com(y_i)$)를 검증하기 때문에, 악의적인 사용자가 중간 연산 값을 잘못 계산하는 경우 이를 탐지할 수 있다. 또한, 서버(100)가 $NIZK(y_i)$ 을 검증하기 때문에, 악의적인 사용자가 $Com(y_i)$ 에는 정당한 중간 연산 값을 사용하고 랜덤 값(R)을 이용하여 암호문($BE.Enc_{PK}(R)$)을 생성하더라도 이를 탐지할 수 있다. 따라서 악의적인 사용자가 정당하지 않은 값을 다른 사용자들에게 공개하려는 경우 서버(100)가 이를 탐지하여 악의적인 사용자만 연산에 대한 정당한 결과 값을 얻을 수 없도록 한다.
- [0105] 상술한 다자간 연산 방법은 컴퓨터에서 실행될 수 있는 프로그램으로 작성가능하고, 컴퓨터로 읽을 수 있는 기록매체를 이용하여 상기 프로그램을 동작시키는 범용 디지털 컴퓨터에서 구현될 수 있다.
- [0106] 상기 다자간 연산 프로그램은 상기 기록매체에 저장되며, 상기 기록매체는 마그네틱 저장매체(예를 들면, 롬, 플로피 디스크, 하드디스크 등), 광학적 판독 매체(예를 들면, 시디롬, 디브이디 등)와 같은 저장매체를 포함한다. 또한, 상기 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 명령어 세트가 저장되고 실행될 수 있다.
- [0107] 본 발명에서 개시된 블록도들은 본 발명의 원리들을 구현하기 위한 회로를 개념적으로 표현한 형태라고 당업자에게 해석될 수 있을 것이다. 유사하게, 임의의 흐름 차트, 흐름도, 상태 전이도, 의사코드 등은 컴퓨터 판독가능 매체에서 실질적으로 표현되어, 컴퓨터 또는 프로세서가 명시적으로 도시되든지 아니든지 간에 이러한 컴퓨

터 또는 프로세서에 의해 실행될 수 있는 다양한 프로세스를 나타낸다는 것이 당업자에게 인식될 것이다.

[0108]

본 발명은 도면에 도시된 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

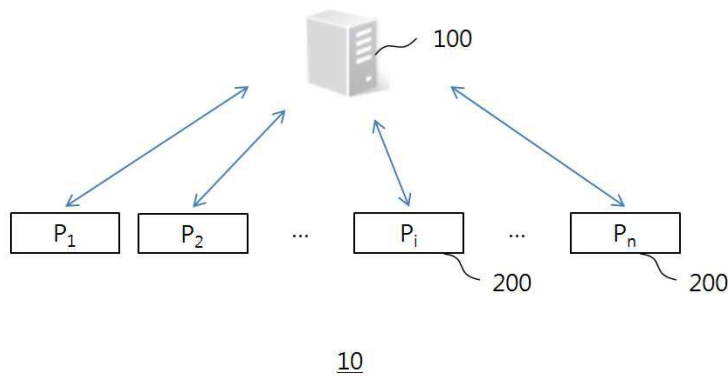
[0109]

10 : 다자간 연산 시스템

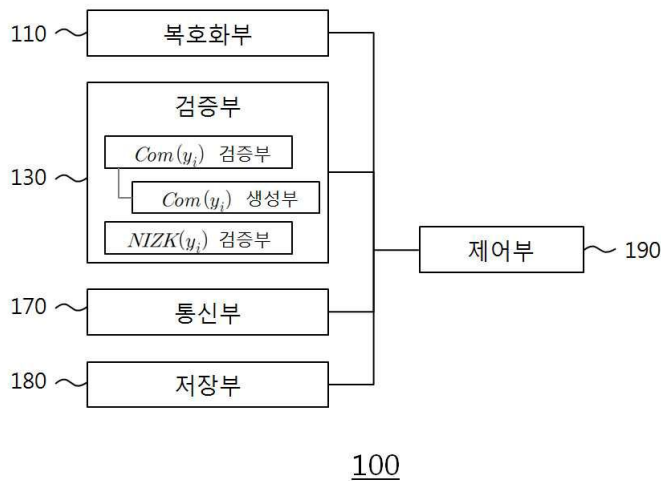
100 : 사용자 장치 200 : 검증 서버

도면

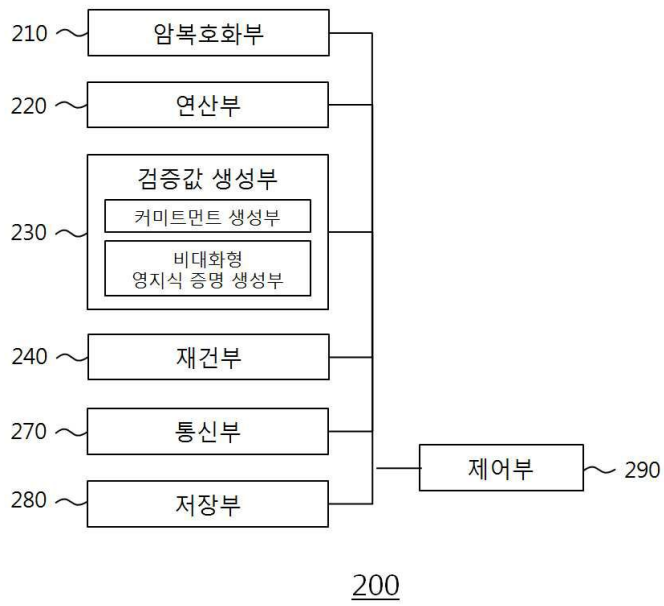
도면1



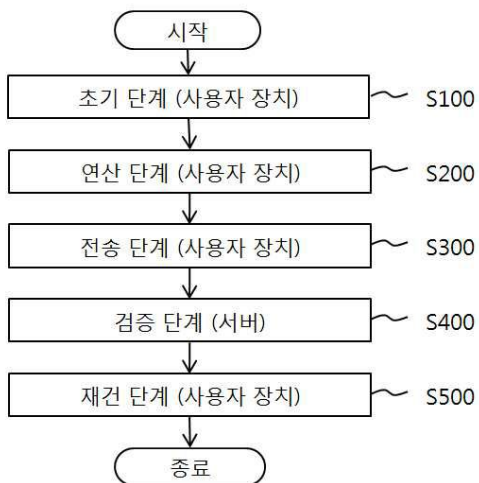
도면2



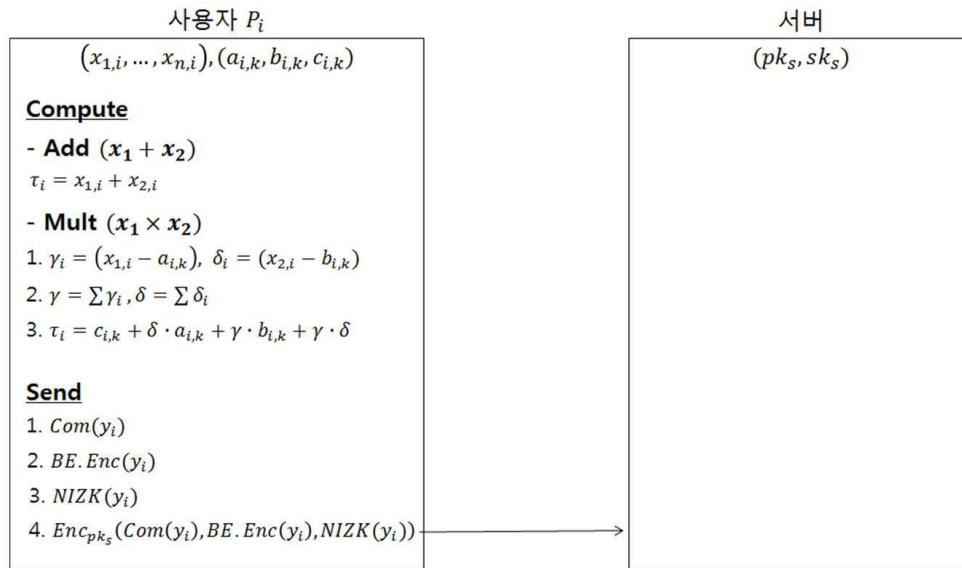
도면3



도면4



도면5



도면6

