(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년12월23일
(11) 등록번호 10-1475747
(24) 등록일자 2014년12월17일

(51) 국제특허분류(Int. Cl.)
H04L 9/30 (2006.01)
(21) 출원번호 10-2014-0007696
(22) 출원일자 2014년01월22일
심사청구일자 2014년01월22일
(56) 선행기술조사문헌
KR1020110101486 A
JP2002232416 A
KR1020020026547 A
KR101123729 B1

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이동훈
서울 종로구 사직로8길 34, 1404호 (내수동, 경희궁의아침3단지)
엄지은
서울 서대문구 가재울로 45, 102동 102호 (남가좌동, 현대아파트)
김효승
서울특별시 성북구 안암로9가길 56(안암동 5가) 206호
(74) 대리인
이강녕, 김홍석, 김등용, 문경진

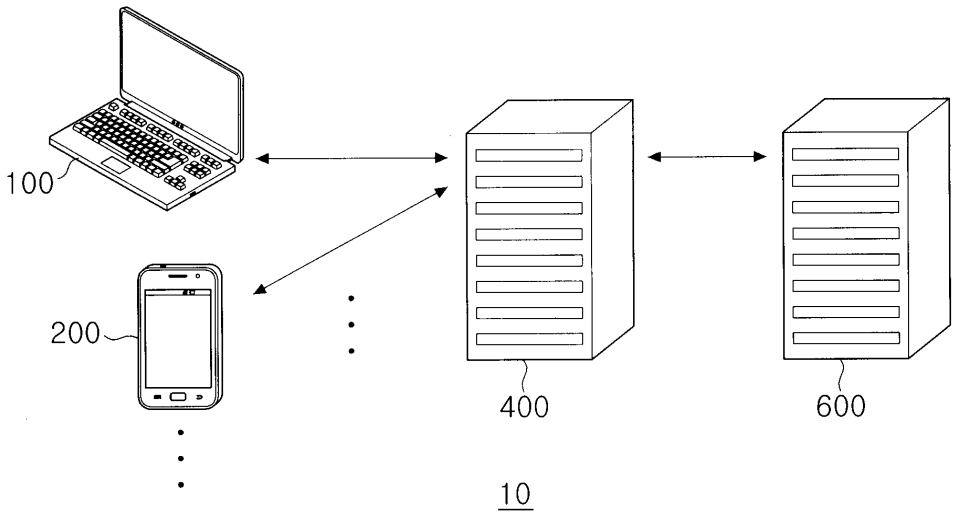
전체 청구항 수 : 총 10 항 심사관 : 홍기완

(54) 발명의 명칭 **동형 암호를 이용한 다자간 위탁 연산 방법**

(57) 요 약

위탁 연산 방법이 개시된다. 상기 위탁 연산 방법은 단말기가 공개키를 이용하여 상기 단말기의 메시지를 암호화하여 암호문을 생성하는 단계, 블라인딩 서버가 상기 암호문을 블라인딩하여 블라인딩된 암호문을 생성하는 단계, 연산 서버가 상기 블라인딩된 암호문에 대한 위탁 연산을 수행하여 암호화된 위탁 연산 결과를 생성하는 단계, 상기 블라인딩 서버가 상기 암호화된 위탁 연산 결과를 언블라인딩하여 언블라인딩된 연산 결과를 생성하는 단계, 및 상기 단말기가 상기 언블라인딩된 연산 결과를 복호화하는 단계를 포함한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 2013060756
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 중견연구자지원_도약연구(전략)
연구과제명 속성 기반 공개키 시스템에 관한 연구
기 여 율 1/1
주관기관 고려대학교 산학협력단
연구기간 2013.09.01 ~ 2014.08.31

특허청구의 범위

청구항 1

단말기가 공개키를 이용하여 상기 단말기의 메시지를 암호화하여 암호문을 생성하는 단계;

블라인딩 서버가 상기 암호문을 블라인딩하여 블라인딩된 암호문을 생성하는 단계;

연산 서버가 상기 블라인딩된 암호문에 대한 위탁 연산을 수행하여 암호화된 위탁 연산 결과를 생성하는 단계;

상기 블라인딩 서버가 상기 암호화된 위탁 연산 결과를 언블라인딩하여 언블라인딩된 연산 결과를 생성하는 단계; 및

상기 단말기가 상기 언블라인딩된 연산 결과를 복호화하는 단계를 포함하는 위탁 연산 방법.

청구항 2

제1항에 있어서, 상기 암호문을 생성하는 단계 이전에,

상기 연산 서버가 공개 파라미터와 마스터키를 생성하는 단계; 및

상기 단말기가 상기 공개 파라미터를 이용하여 상기 공개키와 개인키를 생성하는 단계를 더 포함하는 위탁 연산 방법.

청구항 3

제1항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는,

상기 공개키를 이용하여 상기 메시지와 블라인딩 값의 합에 대한 암호문을 생성함으로써 상기 블라인딩된 암호문을 생성하는 위탁 연산 방법.

청구항 4

제1항에 있어서,

상기 암호화된 위탁 연산 결과를 생성하는 단계는,

마스터키를 이용하여 상기 블라인딩된 암호문을 복호화하는 단계;

복호화된 암호문을 이용하여 위탁 연산을 수행하는 단계; 및

상기 공개키를 이용하여 상기 위탁 연산 결과를 암호화하여 상기 암호화된 위탁 연산 결과를 생성하는 단계를 포함하는 위탁 연산 방법.

청구항 5

제1항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는 상기 암호문의 덧셈에 대한 동형 성질을 이용하여 상기 블라인딩된 암호문을 생성하고,

상기 언블라인딩된 연산 결과를 생성하는 단계는 상기 암호화된 위탁 연산 결과의 덧셈에 대한 동형 성질을 이용하여 상기 언블라인딩된 연산 결과를 생성하는 위탁 연산 방법.

청구항 6

제1항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는,

상기 공개키를 이용하여 블라인딩 값을 암호화하는 단계; 및

암호화된 블라인딩 값을 상기 암호문에 더하여 상기 블라인딩된 암호문을 생성하는 단계를 포함하는 위탁 연산 방법.

청구항 7

블라인딩 서버에서 수행되는 위탁 연산 방법에 있어서,

단말기로부터 수신된 암호문을 블라인딩하여 블라인딩된 암호문을 생성하는 단계;

상기 블라인딩된 암호문을 연산 서버로 송신하는 단계;

상기 연산 서버로부터 암호화된 위탁 연산 결과를 수신하는 단계;

상기 암호화된 위탁 연산 결과를 언블라인딩하여 언블라인딩된 위탁 연산 결과를 생성하는 단계; 및

상기 언블라인딩된 연산 결과를 상기 단말기로 송신하는 단계를 포함하는 위탁 연산 방법.

청구항 8

제7항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는,

메시지와 블라인딩 값의 합에 대한 암호문을 생성함으로써 상기 블라인딩된 암호문을 생성하는 위탁 연산 방법.

청구항 9

제7항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는 상기 암호문의 덧셈에 대한 동형 성질을 이용하여 상기 블라인딩된 암호문을 생성하고,

상기 언블라인딩된 연산 결과를 생성하는 단계는 상기 암호화된 위탁 연산 결과의 덧셈에 대한 동형 성질을 이용하여 상기 언블라인딩된 연산 결과를 생성하는 위탁 연산 방법.

청구항 10

제7항에 있어서,

상기 블라인딩된 암호문을 생성하는 단계는,

상기 단말기로부터 수신된 공개키를 이용하여 블라인딩 값을 암호화하는 단계; 및

암호화된 블라인딩 값을 상기 암호문에 더하여 상기 블라인딩된 암호문을 생성하는 단계를 포함하는 위탁 연산 방법.

명 세 서

기 술 분 야

[0001] 본 발명의 개념에 따른 실시 예는 동형 암호를 이용한 다자간 위탁 연산 방법에 관한 것으로, 특히 메시지의 암호문을 블라인딩하고 블라인딩된 암호문을 이용하여 위탁 연산을 수행할 수 있는 다자간 위탁 연산 방법에 관한 것이다.

배 경 기 술

[0002] 다자간 연산(Multi-party Computation, MPC) 기술은 다수의 사용자가 각자의 비밀 값을 입력 값으로 하여 함수 값을 함께 계산하는 기술을 의미한다. 상기 다자간 연산 기술에 관한 연구의 대부분은 양자간(two-party) 모델 하에 구성되어 다자간(multi-party) 연산으로 확장이 어려운 문제점이 있다.

[0003] 즉, 종래의 다자간 연산 기술은 완벽히 신뢰할 수 있는 제3의 기관(trusted third party)을 두기 어려운 인터넷 환경에서 각각의 사용자들 사이의 상호 전송을 통해 연산을 수행하며 입력 값으로 제공한 개인 정보가 외부로 노출되지 않는 것을 목표로 하였다.

[0004] 그러나, 상기 종래의 다자간 연산 기술은 컴퓨팅 파워가 작은 사용자들의 단말기에 연산에 대한 부담을 주고, 사용자들의 단말기 사이의 전송량이 많아 효율적인 해법을 제시하고 있지 못하다. 이러한 이유로 상기 종래의 다자간 연산 기술은 실제 환경에서 활용하기에는 한계가 있다.

[0005] 최근에 사용자들의 단말기의 연산량을 최소화하고, 상기 단말기 간의 통신량을 줄이기 위해 컴퓨팅 파워가 우수한 클라우드 서버에 연산을 위탁하는 시스템에 대한 개념이 제시되었고, 이에 대한 연구가 진행되고 있다.

[0006] 일반적으로 클라우드 서버는 신뢰할 수 없기 때문에 모든 데이터는 암호화되어 전송되어야 하고 주어진 값에 대한 연산을 수행하는 과정에서 상기 클라우드 서버는 입력 값이나 연산의 결과 값에 대한 어떠한 정보도 얻을 수 없어야 한다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 대한민국 공개특허 제2001-0013155호

(특허문헌 0002) 대한민국 등록특허 제10-0772550호

발명의 내용

해결하려는 과제

[0008] 본 발명이 이루고자 하는 기술적인 과제는 암호화된 메시지를 블라인딩하고 블라인딩된 암호문에 대한 위탁 연산을 수행함으로써, 사용자의 메시지와 위탁 연산 결과에 대한 프라이버시를 보장할 수 있는 위탁 연산 방법을 제공하는 것이다.

과제의 해결 수단

[0009] 본 발명의 실시 예에 따른 위탁 연산 방법은 단말기가 공개키를 이용하여 상기 단말기의 메시지를 암호화하여 암호문을 생성하는 단계, 블라인딩 서버가 상기 암호문을 블라인딩하여 블라인딩된 암호문을 생성하는 단계, 연산 서버가 상기 블라인딩된 암호문에 대한 위탁 연산을 수행하여 암호화된 위탁 연산 결과를 생성하는 단계, 상기 블라인딩 서버가 상기 암호화된 위탁 연산 결과를 언블라인딩하여 언블라인딩된 연산 결과를 생성하는 단계, 및 상기 단말기가 상기 언블라인딩된 연산 결과를 복호화하는 단계를 포함한다.

[0010] 상기 위탁 연산 방법은 상기 암호문을 생성하는 단계 이전에, 상기 연산 서버가 공개 파라미터와 마스터키를 생성하는 단계, 및 상기 단말기가 상기 공개 파라미터를 이용하여 상기 공개키와 개인키를 생성하는 단계를 더 포함할 수 있다.

[0011] 상기 블라인딩된 암호문을 생성하는 단계는 상기 공개키를 이용하여 상기 메시지와 블라인딩 값의 합에 대한 암호문을 생성함으로써 상기 블라인딩된 암호문을 생성할 수 있다.

[0012] 상기 암호화된 위탁 연산 결과를 생성하는 단계는 마스터키를 이용하여 상기 블라인딩된 암호문을 복호화하는 단계, 복호화된 암호문을 이용하여 위탁 연산을 수행하는 단계, 및 상기 공개키를 이용하여 상기 위탁 연산 결과를 암호화하여 상기 암호화된 위탁 연산 결과를 생성하는 단계를 포함할 수 있다.

[0013] 또한, 상기 블라인딩된 암호문을 생성하는 단계는 상기 암호문의 덧셈에 대한 동형 성질을 이용하여 상기 블라인딩된 암호문을 생성하고, 상기 언블라인딩된 연산 결과를 생성하는 단계는 상기 암호화된 위탁 연산 결과의 덧셈에 대한 동형 성질을 이용하여 상기 언블라인딩된 연산 결과를 생성할 수 있다.

[0014] 상기 블라인딩된 암호문을 생성하는 단계는 상기 공개키를 이용하여 블라인딩 값을 암호화하는 단계, 및 암호화된 블라인딩 값을 상기 암호문에 더하여 상기 블라인딩된 암호문을 생성하는 단계를 포함할 수 있다.

[0015] 본 발명의 다른 실시 예에 따른 위탁 연산 방법은 블라인딩 서버에서 수행되는 위탁 연산 방법으로서, 단말기로부터 수신된 암호문을 블라인딩하여 블라인딩된 암호문을 생성하는 단계, 상기 블라인딩된 암호문을 연산 서버로 송신하는 단계, 상기 연산 서버로부터 암호화된 위탁 연산 결과를 수신하는 단계, 상기 암호화된 위탁 연산 결과를 언블라인딩하여 언블라인딩된 위탁 연산 결과를 생성하는 단계, 및 상기 언블라인딩된 연산 결과를 상기

단말기로 송신하는 단계를 포함한다.

[0016] 상기 블라인딩된 암호문을 생성하는 단계는 메시지와 블라인딩 값의 합에 대한 암호문을 생성함으로써 상기 블라인딩된 암호문을 생성할 수 있다.

[0017] 또한, 상기 블라인딩된 암호문을 생성하는 단계는 상기 암호문의 덧셈에 대한 동형 성질을 이용하여 상기 블라인딩된 암호문을 생성하고, 상기 언블라인딩된 연산 결과를 생성하는 단계는 상기 암호화된 위탁 연산 결과의 덧셈에 대한 동형 성질을 이용하여 상기 언블라인딩된 연산 결과를 생성할 수 있다.

[0018] 상기 블라인딩된 암호문을 생성하는 단계는 상기 단말기로부터 수신된 공개키를 이용하여 블라인딩 값을 암호화하는 단계, 및 암호화된 블라인딩 값을 상기 암호문에 더하여 상기 블라인딩된 암호문을 생성하는 단계를 포함할 수 있다.

발명의 효과

[0019] 본 발명의 실시 예에 따른 다자간 위탁 연산 방법은 사용자들의 메시지를 암호화하여 서버로 전송하기 때문에 상기 사용자들의 단말기 사이의 통신량이 없고, 상기 서버에 연산을 위탁하기 때문에 상기 단말기의 연산량을 감소시킬 수 있는 효과가 있다.

[0020] 또한, 상기 다자간 위탁 연산 방법은 상기 사용자들 각각의 단말기로부터 수신된 암호문을 블라인딩하는 블라인딩 서버와 상기 블라인딩 서버로부터 수신된 블라인딩된 암호문에 대한 연산을 수행하는 연산 서버로 구현되어, 상기 사용자들의 입력 값과 연산의 결과 값에 대한 프라이버시를 보호할 수 있는 효과가 있다.

[0021] 또한, 상기 다자간 위탁 연산 방법은 상기 사용자들 각각의 단말기와 상기 블라인딩 서버 사이의 통신량과 상기 블라인딩 서버와 상기 연산 서버 사이의 통신량을 감소시켜 효율적인 다자간 연산을 제공할 수 있는 효과가 있다.

도면의 간단한 설명

[0022] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 본 발명의 일 실시예에 따른 위탁 연산 시스템의 개략도이다.

도 2는 도 1에 도시된 제1 단말기의 기능 블럭도이다.

도 3은 도 1에 도시된 위탁 연산 시스템을 이용한 위탁 연산 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 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.

[0024] 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.

[0025] 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.

[0026] 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.

[0027] 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.

[0028] 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.

[0029] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다.

[0030] 우선 본 발명의 배경이 되는 기술인 동형 암호(homomorphic encryption)는 아래와 같이 설명될 수 있다.

[0031] 상기 동형 암호는 암호문을 복호화하지 않고도 입력 값에 대한 함수 연산을 수행할 수 있는 기술로 암호문을 생성하여 전송하는 송신자(Sender), 상기 암호문에 대한 연산을 수행하는 계산자(Evaluator), 및 상기 계산자의 출력 값을 복호화하여 최종 연산 결과 값을 얻는 수신자(Receiver)로 구성된다. 경우에 따라, 상기 송신자와 상기 수신자는 동일할 수 있다. 상기 계산자는 비밀키에 대한 어떤 정보도 없이 오직 공개되어 있는 연산키를 이용하여 연산을 수행할 수 있다.

[0032] 현재까지 제안된 기법들은 주로 동일한 키로 암호화된 데이터에 대해서만 암호문 연산(Evaluate)이 가능하다. 덧셈에 대한 동형 성질을 제공하는 대표적인 암호 기법으로 1999년에 제안된 Paillier 암호 시스템이 존재하고, 곱셈에 대한 동형 성질을 제공하는 암호 기법으로 1984년에 제안된 Elgamal 암호 시스템이 있다.

[0033] 완전 동형 암호(Fully homomorphic encryption)는 모든 연산을 지원하는 암호를 의미하는데, 비트 단위의 AND(곱셈)와 XOR(덧셈) 연산이 가능할 경우 두 연산을 조합하여 모든 논리 연산을 설계할 수 있기 때문에 암호문 간의 두 연산에 대한 동형 성질을 제공하도록 구성하면 모든 연산을 지원할 수 있다. 그러나, 곱셈을 효율적으로 연산하는 기술을 설계하는 것은 매우 어렵기 때문에 현재까지도 곱셈과 덧셈에 대한 동형 성질을 제공하는 효율적인 기법에 대한 연구가 활발하게 진행되고 있다.

[0034] 본 발명에서 이용하는 동형 암호 기법은 2003년 Bresson, Catalano 및 Pointcheval이 제안한 기법(이하, BCP 암호 기법이라 함)으로 덧셈에 대한 동형 성질을 제공한다. 상기 BCP 암호 기법은 Paillier 암호 기법을 기반으로 설계된 것으로써, 사용자의 개인키를 가지고 복호화하거나 마스터키를 가지고 복호화하는 두 개의 복호화 알고리즘이 있는 점을 특징으로 한다.

[0035] 다만 본 발명이 상기 BCP 암호 기법에 제한되는 것은 아니며, 암호 기법의 이름과는 무관하게 상기 BCP 암호 기법과 동일하거나 유사한 알고리즘을 제공하는 암호 기법을 이용할 수도 있다.

[0036] 상기 BCP 암호 기법은 5개의 알고리즘(BCP.Setup, BCP.KeyGen, BCP.Enc, BCP.Dec, BCP.mDec)을 포함한다.

[0037] BCP.Setup 알고리즘(셋업 알고리즘)은 보안 상수(k)를 입력받아 공개 파라미터와 마스터키를 생성하는 알고리즘이다. 상기 셋업 알고리즘은 안전 소수(safe prime)인 p(p=2p´+1)와 q(q=2q´+1)를 생성한 뒤 상기 p와 상기 q를 곱한 N(N=pq)을 계산한다. 상기 p´과 상기 q´은 소수(prime number)이다.

[0038] 또한, 상기 셋업 알고리즘은 k($k \in [1, N-1]$)에 대해 $g^{p'q'} = 1 + kN \bmod N^2$ 를 만족하는 임의의 g($g \in Z_{N^2}^*$)를 선택한다. 이때 메시지 공간은 Z_N 이다. 즉, k는 1보다 크거나 같고 N보다 작은 정수 값을 갖는다. 상기 g는 1보다 크거나 같고 (N^2-1)보다 작거나 같은 정수들의 집합의 원소이다.

[0039] 상기 셋업 알고리즘은 최종적으로 공개 파라미터(PP=(N,k,g))와 마스터키(MK=(p´,q´))를 생성하여 출력할 수 있다.

[0040] BCP.KeyGen 알고리즘(키생성 알고리즘)은 상기 공개 파라미터(PP)를 이용하여 사용자의 공개키(pk)와 개인키

(sk)를 생성하는 알고리즘이다.

[0041] 상기 키생성 알고리즘은 임의의 a($a \in Z_{N^2}$)를 선택하여 h($h = g^a \bmod N^2$)를 계산함으로써, 상기 공개키(pk=h)와 상기 개인키(sk=a)를 생성하여 출력한다.

[0042] BCP.Enc 알고리즘(암호화 알고리즘)은 상기 사용자의 상기 공개키(pk)를 이용하여 메시지(m)을 암호화하는 알고리즘이다.

[0043] 상기 암호화 알고리즘은 상기 공개키(pk=h)와 상기 메시지($m \in Z_N$)에 대해 임의의 난수($r = Z_{N^2}$)을 선택하여 암호문(A,B)을 생성하고, 생성된 암호문(A,B)을 출력한다. 상기 암호문(A,B)는 아래 수식을 이용하여 생성될 수 있다.

[0044] $$A = g^r \bmod N^2$$

[0045] $$B = h^r (1 + mN) \bmod N^2$$

[0046] BCP.Dec 알고리즘(제1 복호화 알고리즘)은 상기 사용자의 상기 개인키(sk=a)를 이용하여 상기 암호문(A,B)을 복호화하는 알고리즘이다.

[0047] 상기 제1 복호화 알고리즘은 아래 수식을 이용하여 상기 암호문(A,B)을 복호화하고 메시지(m)를 출력한다.

[0048] $$m = \frac{B/(A^a) - 1 \bmod N^2}{N}$$

[0049] BCP.mDec 알고리즘(제2 복호화 알고리즘)은 상기 마스터키(MK)를 이용하여 상기 암호문(A,B)을 복호화하는 알고리즘이다.

[0050] 상기 제2 복호화 알고리즘은 상기 마스터키(MK=(p´,q´))와 상기 암호문(A,B)에 사용된 공개키(pk=h)에 대해 다음 두 식을 계산하여 상기 개인키(sk=a)와 상기 임의의 난수(r)에 대한 정보를 얻는다.

[0051] $$a \bmod N = \frac{(h^{p'q'} - 1) \bmod N^2}{N} \times k^{-1} \bmod N$$

[0052] $$r \bmod N = \frac{(A^{p'q'} - 1) \bmod N^2}{N} \times k^{-1} \bmod N$$

[0053] 상기 제2 복호화 알고리즘은 다음의 식을 이용하여 상기 암호문(A,B)을 복호화 하고 메시지(m)을 출력한다.

[0054] $$m = \frac{((B/(g^{ar \bmod N}))^{p'q'} - 1) \bmod N^2}{N} \times (p'q')^{-1} \bmod N$$

[0055] 상술한 BCP 암호 기법은 덧셈에 대한 동형 성질을 갖기 때문에 메시지의 합에 대한 암호문을 생성할 수 있고, 지수승 연산을 통해 메시지의 상수배에 대한 암호문을 생성할 수 있다.

[0056] 즉, 덧셈에 대한 동형성질을 이용하여 두 암호문 $(A, B) = (g^r \bmod N^2, h^r (1 + mN) \bmod N^2)$ 와 $(A', B') = (g^{r'} \bmod N^2, h^{r'} (1 + m'N) \bmod N^2)$ 에 대해 각 항목의 값을 곱하면 두 암호문의 메시지를 더한 값(m + m´)의 암호문

$$(AA', BB') = g^{r+r'} \bmod N^2, h^{r+r'}(1+(m+m')N) \bmod N^2$$

을 계산할 수 있다.

[0057] 또한, 암호문 $(A,B) = (g^r \bmod N^2, h^r(1+mN) \bmod N^2)$ 의 각 항목에 $\sigma \in Z_N$ 를 지수승하면 mσ에 대한 암호문 $(A^\sigma, B^\sigma) = (g^{\sigma r} \bmod N^2, h^{\sigma r}(1+\sigma mN) \bmod N^2)$ 을 생성할 수 있다.

[0058] 도 1은 본 발명의 일실시 예에 따른 위탁 연산 시스템을 도시한다.

[0059] 도 1을 참조하면, 위탁 연산 시스템(10)은 제1 단말기(100)와 제2 단말기(200)를 포함하는 다수의 단말기들, 블라인딩 서버(400), 및 연산 서버(600)를 포함한다.

[0060] 도 1에는 제1 단말기(100)와 제2 단말기(200)가 도시되어 있으나, 상기 다수의 단말기들은 2 이상의 단말기를 포함할 수 있다.

[0061] 상기 다수의 단말기들 각각은 유선 또는 무선 통신망을 통하여 블라인딩 서버(400)와 통신할 수 있다.

[0062] 제1 단말기(100)는 블라인딩 서버(400)로부터 수신된 공개 파라미터(PP=(N,k,g))를 이용하여 BCP.KeyGen 알고리즘을 수행한다. 즉, 제1 단말기(100)는 상기 공개 파라미터를 이용하여 공개키($pk = h = g^a \bmod N^2$)와 개인키($sk = a \in Z_{N^2}$)를 생성한다. 제1 단말기(100)은 상기 공개키(pk)를 블라인딩 서버(400)로 전하고, 블라인딩 서버(400)는 상기 공개키(pk)를 연산 서버(600)로 전송할 수 있다. 실시 예에 따라, 제1 단말기(100)는 연산 서버(600)로부터 상기 공개 파라미터를 수신할 수도 있고, 상기 공개키(pk)를 연산 서버(600)로 직접 전송할 수도 있다.

[0063] 또한, 제1 단말기(100)는 BCP.Enc 알고리즘을 수행하여 메시지($m \in Z_N$)을 암호화한다. 상기 공개키(pk)를 이용하여 암호화된 암호문(A,B)는 다음과 같다.

[0064] $$(A,B) = g^r \bmod N^2, h^r(1+mN) \bmod N^2$$

[0065] 여기서 r($r = Z_{N^2}$)은 임의의 난수이다. 제1 단말기(100)는 상기 암호문(A,B)을 블라인딩 서버(400)로 전송한다.

[0066] 또한, 제1 단말기(100)는 BCP.Dec 알고리즘을 수행하여 블라인딩 서버(400)로부터 수신된 암호문(언블라인딩된 연산 결과)을 복호화한다. 따라서, 제1 단말기(100)는 암호화되지 않은 연산 결과를 생성할 수 있다.

[0067] 구체적으로 제1 단말기(100)는 블라인딩 서버로부터 수신된 언블라인딩된 연산 결과($(A,B) = g^r \bmod N^2, h^r(1+m_1m_2N) \bmod N^2$)에 대해 BCP.Dec 알고리즘을 수행한다. 즉, 제1 단말기(100)는 개인키(sk=a)를 이용하여 상기 언블라인딩된 연산 결과를 복호화함으로써 연산결과($m_1 + m_2$ 또는 m_1m_2)를 계산할 수 있다. 상기 연산 결과는 다음 식을 이용하여 계산될 수 있다.

[0068] $$\frac{B/(A^a) - 1 \bmod N^2}{N}$$

[0069] 제1 단말기(100)를 제외한 다수의 단말기들 각각은 제1 단말기(100)의 동작과 동일한 동작을 수행하므로, 이에 대한 기재는 생략하기로 한다.

[0070] 블라인딩 서버(400)는 덧셈에 대한 동형 성질을 이용하여, 제1 단말기(100)로부터 수신된 암호문(A,B)을 블라인딩한다. 또한, 블라인딩 서버(400)는 블라인딩된 암호문(A´,B)을 연산 서버(600)로 전송한다. 이때, 블라인딩

서버(400)와 연산 서버(600)는 서로 공모하지 않은 서버로 가정한다. 이는 블라인딩 서버(400)와 연산 서버(600)로부터 사용자의 메시지(m)와 연산의 결과에 대한 정보를 보호하기 위함이다.

[0071] 구체적으로 블라인딩 서버(400)는 암호문(A,B)의 암호화에 사용된 공개키(pk_1=h)로 임의의 σ_1($\sigma_1 \in Z_N$)를 암호화한 암호문($(A', B') = (g^{r'} \bmod N^2, h^{r'}(1+\sigma_1 N) \bmod N^2)$)을 생성한다.

[0072] 블라인딩 서버(400)는 두 암호문((A,B),(A´,B´))에 대해 덧셈에 대한 동형 성질을 이용하여 ($m + \sigma_1$)에 대한 암호문($(AA', BB') = g^{r+r'} \bmod N^2, h^{r+r'}(1+(m+\sigma_1)N) \bmod N^2$)을 생성하여, 생성된 암호문(AA´,BB´)을 저장한다. 간단히 말해서 블라인딩 서버(400)는 임의의 σ_1에 대해 아래 식을 수행한다.

[0073] $$Enc_{pk}(m) \times Enc_{pk}(\sigma_1) = Enc_{pk}(m+\sigma_1)$$

[0074] 여기서 임의의 σ는 블라인딩 값으로 명명될 수 있다.

[0075] 또한, 블라인딩 서버(400)는 연산 서버(600)로부터 수신된 암호화된 연산 결과를 언블라인딩(상기 암호화된 연산 결과에서 블라인딩 값을 제거)하고, 언블라인딩된 연산 결과를 제1 단말기(100)로 전송한다.

[0076] 연산 서버(600)는 BCP.Setup 알고리즘을 수행하여 공개 파라미터(PP=(N,k,g))와 마스터키(MK=(p´,q´))을 생성한다. 따라서, 연산 서버(600)는 어떠한 암호문에 대해서도 복호화할 수 있는 능력을 갖는다.

[0077] 또한, 연산 서버(600)는 블라인딩 서버(400)로부터 수신된 블라인딩된 암호문(A´,B´)에 대해 BCP.mDec 알고리즘을 실행한다. 즉, 연산 서버(600)는 마스터키(MK)를 이용하여 상기 블라인딩된 암호문(A´,B´)을 복호화한다.

[0078] 연산 서버(600)는 복호화된 암호문을 대상으로 연산(덧셈 또는 곱셈)을 수행한다. 연산 서버(600)는 연산 결과를 이용자의 공개키(pk)를 이용하여 암호화하고 암호화된 연산 결과를 블라인딩 서버(400)로 전송한다.

[0079] 이하에서는 제1 단말기(100)의 메시지(m_1)와 제2 단말기(200)의 메시지(m_2)에 대한 위탁 연산 과정을 상세하게 설명하기로 한다.

[0080] 우선, 제1 단말기(100)에서 사용된 공개키는 pk_1이고, 제2 단말기(200)에서 사용된 공개키는 pk_2라 한다.

[0081] 덧셈 연산을 수행하기 위한 과정은 다음과 같다.

[0082] 블라인딩 서버(400)는 제1 단말기(100)로부터 수신된 암호문($Enc_{pk_1}(m_1)$)을 σ_1을 블라인딩 값으로 하여 블라인딩하여 블라인딩된 암호문($Enc_{pk_1}(m_1+\sigma_1)$)을 생성하고, 제2 단말기(100)로부터 수신된 암호문($Enc_{pk_2}(m_2)$)을 σ_2를 블라인딩 값으로 하여 블라인딩하여, 블라인딩된 암호문($Enc_{pk_2}(m_2+\sigma_2)$)을 생성한다. 두 개의 블라인딩된 암호문은 연산 서버(600)로 전송된다.

[0083] 연산 서버(600)는 블라인딩된 암호문을 블라인딩 서버(400)로부터 수신하고, 블라인딩된 암호문을 복호화한다. 상기 두개의 블라인딩된 암호문을 복호화하면 ($m_1+\sigma_1$)와 ($m_2+\sigma_2$)를 얻을 수 있다. 연산 서버(600)는 복호화된 값들에 대한 덧셈연산을 수행하여 연산 결과 $(m_1+m_2+\sigma_1+\sigma_1)$를 얻는다. 연산

서버(600)는 상기 연산 결과를 pk_1과 pk_2로 암호화하고 암호화된 연산 결과($Enc_{pk_1}(m_1+m_2+\sigma_1+\sigma_1)$, $Enc_{pk_2}(m_1+m_2+\sigma_1+\sigma_2)$)를 블라인딩 서버(400)로 전송한다.

[0084] 블라인딩 서버(400)는 $(-\sigma_1-\sigma_2)$를 pk_1과 pk_2로 암호화한 $Enc_{pk_1}(-\sigma_1-\sigma_2)$과 $Enc_{pk_2}(-\sigma_1-\sigma_2)$을 계산한다. 블라인딩 서버(400)는 $Enc_{pk_1}(-\sigma_1-\sigma_2)$를 $Enc_{pk_1}(m_1+m_2+\sigma_1+\sigma_1)$에 곱하여 $Enc_{pk_1}(m_1+m_2)$를 계산하고, $Enc_{pk_2}(-\sigma_1-\sigma_2)$를 $Enc_{pk_2}(m_1+m_2+\sigma_1+\sigma_2)$에 곱하여 $Enc_{pk_2}(m_1+m_2)$를 얻는다. 즉, 블라인딩 서버(400)는 상기 암호화된 연산 결과에서 블라인딩 값(σ_1, σ_2)을 제거(언블라인딩)하여 언블라인딩된 연산 결과를 생성한다.

[0085] 블라인딩 서버(400)는 언블라인딩된 연산 결과($Enc_{pk_1}(m_1+m_2)$)를 제1 단말기(100)로 전송하고, 언블라인딩된 연산 결과($Enc_{pk_2}(m_1+m_2)$)를 제2 단말기(200)로 전송한다.

[0086] 제1 단말기(100)는 언블라인딩된 연산 결과($Enc_{pk_1}(m_1+m_2)$)를 복호화함으로써 연산 결과(m_1+m_2)를 얻을 수 있고, 제2 단말기(200)는 언블라인딩된 연산 결과($Enc_{pk_2}(m_1+m_2)$)를 복호화함으로써 연산 결과(m_1+m_2)를 얻을 수 있다.

[0087] 곱셈 연산을 수행하기 위한 과정은 다음과 같다.

[0088] 블라인딩 서버(400)는 두 개의 블라인딩된 암호문($Enc_{pk_1}(m_1+\sigma_1)$, $Enc_{pk_2}(m_2+\sigma_2)$) 외에 $Enc_{pk_1}(m_1\sigma_2)$와 $Enc_{pk_2}(m_2\sigma_1)$를 생성하여, 네 개의 암호문을 연산 서버(600)로 전송한다. 블라인딩 서버(400)는 BCP 암호 기법의 특징을 이용하여, $Enc_{pk_1}(m_1)$에 σ_2를 지수승하여 $Enc_{pk_1}(m_1\sigma_2)$를 계산할 수 있고, $Enc_{pk_2}(m_2)$에 σ_1를 지수승하여 $Enc_{pk_2}(m_2\sigma_1)$를 계산할 수 있다.

[0089] 연산 서버(600)는 수신된 네 개의 암호문 각각에 BCP.mDec 알고리즘을 수행하여 복호화한다. 즉, 연산 서버(600)는 마스터키(MK)를 이용하여 상기 네 개의 암호문 각각을 복호화할 수 있다. 연산 서버(600)는 복호화 결과로 ($m_1+\sigma_1$), ($m_2+\sigma_2$), ($m_1\sigma_2$), 및 ($m_2\sigma_1$)를 얻을 수 있다. 따라서 연산 서버(600)는 ($m_1+\sigma_1$)과 ($m_2+\sigma_2$)에 대해 곱셈 연산을 수행한 값에서 ($m_1\sigma_2$)과 ($m_2\sigma_1$)를 빼서 연산 결과($m_1m_2+\sigma_1\sigma_2$)를 얻을 수

있다.

[0090] 연산 서버(600)는 상기 연산 결과를 pk_1과 pk_2로 암호화한 후 암호화된 연산 결과를 블라인딩 서버(400)로 전송한다.

[0091] 블라인딩 서버(400)는 $Enc_{pk_1}(-\sigma_1\sigma_2)$를 계산하고, 연산 서버(600)로부터 수신된 $Enc_{pk_1}(m_1m_2+\sigma_1\sigma_2)$에 $Enc_{pk_1}(-\sigma_1\sigma_2)$를 곱함으로써 $Enc_{pk_1}(m_1m_2)$을 얻을 수 있다. 또한, 블라인딩 서버(400)는 $Enc_{pk_2}(-\sigma_1\sigma_2)$를 계산하고, 연산 서버(600)로부터 수신된 $Enc_{pk_2}(m_1m_2+\sigma_1\sigma_2)$에 $Enc_{pk_2}(-\sigma_1\sigma_2)$를 곱함으로써 $Enc_{pk_2}(m_1m_2)$를 얻을 수 있다. 즉, 블라인딩 서버(400)는 연산 서버(600)로부터 수신된 암호화된 결과값에서 블라인딩 값(σ_1, σ_2)를 제거(언블라인딩)함으로써 언블라인딩된 연산 결과($Enc_{pk_1}(m_1m_2)$, $Enc_{pk_2}(m_1m_2)$)를 얻을 수 있다.

[0092] 블라인딩 서버(400)는 언블라인딩된 연산 결과($Enc_{pk_1}(m_1m_2)$)를 제1 단말기(100)로 전송하고, 언블라인딩된 연산 결과($Enc_{pk_2}(m_1m_2)$)를 제2 단말기(200)로 전송한다.

[0093] 제1 단말기(100)와 제2 단말기(200)는 각자의 개인키를 이용하여 상기 언블라인딩된 연산 결과를 복호화함으로써, 연산 결과를 얻을 수 있다.

[0094] 상기 덧셈 연산과 곱셈 연산을 수행하는 과정에서 제1 단말기(100)의 메시지(m_1)와 제2 단말기(200)의 메시지(m_2)만을 대상으로 하는 대한 연산이 예시하였으나, 본 발명이 이에 한정되는 것은 아니고, 단말기의 갯수와 메시지의 갯수에 무관하게 적용될 수 있음은 자명하다.

[0095] 도 2는 도 1에 도시된 제1 단말기의 기능 블럭도이다.

[0096] 도 1과 도 2를 참조하면, 제1 단말기는 키생성 모듈(110), 암호화 모듈(130), 복호화 모듈(150), 통신 모듈(170), 메모리(180), 및 제어 모듈(190)을 포함한다.

[0097] 본 명세서에서 사용되는 모듈이라 함은 본 발명의 기술적 사상을 수행하기 위한 하드웨어 및 상기 하드웨어를 구동하기 위한 소프트웨어의 기능적, 구조적 결합을 의미할 수 있다. 예컨대, 상기 모듈은 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.

[0098] 키생성 모듈(110)은 제어 모듈(190)의 제어 하에, 수신된 공개 파라미터(PP=(N,k,g))를 이용하여 BCP.KeyGen 알고리즘을 수행한다. 즉, 키생성 모듈(110)은 상기 공개 파라미터를 이용하여 공개키($pk = h = g^a \bmod N^2$)와 개인키($sk = a \in Z_{N^2}$)를 생성한다.

[0099] 암호화 모듈(130)은 제어 모듈(190)의 제어 하에, BCP.Enc 알고리즘을 수행하여 메시지($m \in Z_N$)을 암호화한다. 상기 공개키(pk)를 이용하여 암호화된 암호문(A,B)는 다음과 같다.

[0100] $$(A,B) = g^r \bmod N^2, h^r(1+mN) \bmod N^2$$

[0101] 여기서 r($r = Z_{N^2}$)은 임의의 난수이다.

[0102] 복호화 모듈(150)은 제어 모듈(190)의 제어 하에, BCP.Dec 알고리즘을 수행하여 블라인딩 서버(400)로부터 수신된 암호문(언블라인딩된 연산 결과)을 복호화한다. 따라서, 복호화 모듈(150)은 암호화되지 않은 연산 결과를 생성할 수 있다.

[0103] 구체적으로 복호화 모듈(150)은 블라인딩 서버로부터 수신된 언블라인딩된 연산 결과 ($(A,B) = g^r \bmod N^2, h^r(1+m_1 m_2 N) \bmod N^2$)에 대해 BCP.Dec 알고리즘을 수행한다. 즉, 복호화 모듈(150)은 개인키(sk=a)를 이용하여 상기 언블라인딩된 연산 결과를 복호화함으로써 연산결과($m_1 + m_2$ 또는 $m_1 m_2$)를 계산할 수 있다. 상기 연산 결과는 다음 식을 이용하여 계산될 수 있다.

[0104] $$\frac{B/(A^a) - 1 \bmod N^2}{N}$$

[0105] 통신 모듈(170)은 제어 모듈(190)의 제어 하에, 블라인딩 서버(400)와 유선 또는 무선 통신망을 통하여 통신을 수행할 수 있다. 실시 예에 따라 통신 모듈(170)은 연산 서버(600)와 통신을 수행할 수 있다.

[0106] 통신 모듈(170)은 블라인딩 서버(400) 또는 연산 서버(600)로부터 공개 파라미터를 수신하고, 블라인딩 서버(400)로부터 언블라인딩된 연산 결과를 수신할 수 있다.

[0107] 또한, 통신 모듈(170)은 블라인딩 서버(400) 또는 연산 서버(600)로 공개키(pk)를 전송하고, 블라인딩 서버(400)로 메시지(m)에 대한 암호문(A,B)을 전송할 수 있다.

[0108] 메모리(180)는 프로그램 메모리와 데이터 메모리를 포함할 수 있다. 상기 프로그램 메모리에는 제1 단말기(100)의 동작을 제어하기 위한 프로그램들이 저장될 수 있다. 상기 데이터 메모리에는 상기 프로그램들을 수행하는 과정 중에 발생되는 데이터들이 저장될 수 있다.

[0109] 제어 모듈(190)은 제1 단말기(100)의 전반적인 동작을 제어한다. 즉, 키생성 모듈(110), 암호화 모듈(130), 복호화 모듈(150), 통신 모듈(170), 및 메모리(180)의 동작을 제어할 수 있다.

[0110] 도 3은 도 1에 도시된 위탁 연산 시스템을 이용한 위탁 연산 방법을 설명하기 위한 흐름도이다.

[0111] 이하에서는 상술된 내용과 중복되는 내용에 관한 기재는 생략하기로 한다.

[0112] 도 1 내지 도 3을 참조하면, 연산 서버(600)는 BCP.Setup 알고리즘을 수행하여 공개 파라미터(PP=(N,k,g))와 마스터키(MK=(p´,q´))을 생성한다(S100). 따라서, 연산 서버(600)는 어떠한 암호문에 대해서도 복호화할 수 있는 능력을 갖는다.

[0113] 상기 공개 파라미터(PP)는 블라인딩 서버(400)로 전송되고, 블라인딩 서버(400)는 수신된 공개 파라미터(PP)를 제1 단말기(100)로 전송한다. 실시 예에 따라, 연산 서버(600)는 상기 공개 파라미터(PP)를 제1 단말기(100)로 직접 전송할 수도 있다.

[0114] 제1 단말기(100)는 블라인딩 서버(400) 또는 연산 서버(600)로부터 수신된 공개 파라미터(PP=(N,k,g))를 이용하여 BCP.KeyGen 알고리즘을 수행한다(S200). 즉, 제1 단말기(100)는 상기 공개 파라미터(PP)를 이용하여 공개키 ($pk = h = g^a \bmod N^2$)와 개인키($sk = a \in Z_{N^2}$)를 생성한다.

[0115] 제1 단말기(100)는 상기 공개키(pk)를 블라인딩 서버(400)로 전하고, 블라인딩 서버(400)는 상기 공개키(pk)를 연산 서버(600)로 전송할 수 있다. 실시 예에 따라, 제1 단말기(100)는 상기 공개키(pk)를 블라인딩 서버(400)와 연산 서버(600)로 전송할 수도 있다.

[0116] 제1 단말기(100)는 BCP.Enc 알고리즘을 수행하여 메시지($m \in Z_N$)을 암호화한다(S300). 상기 공개키(pk)를 이용하여 암호화된 암호문(A,B)는 다음과 같다.

[0117] $$(A,B)=g^{r}\bmod N^{2},h^{r}(1+mN)\bmod N^{2}$$

[0118] 제1 단말기(100)는 상기 암호문(A,B)을 블라인딩 서버(400)로 전송한다.

[0119] 블라인딩 서버(400)는 덧셈에 대한 동형 성질을 이용하여, 제1 단말기(100)로부터 수신된 암호문(A,B)을 블라인딩한다(S400). 구체적인 블라인딩 과정에 관해서는 생략하기로 한다.

[0120] 블라인딩 서버(400)는 블라인딩된 암호문(A´,B´)을 연산 서버(600)로 전송한다.

[0121] 연산 서버(600)는 블라인딩 서버(400)로부터 수신된 블라인딩된 암호문(A´,B´)에 대해 BCP.mDec 알고리즘을 실행한다(S500). 즉, 연산 서버(600)는 마스터키(MK)를 이용하여 상기 블라인딩된 암호문(A´,B´)을 복호화한다.

[0122] 연산 서버(600)는 복호화된 암호문을 대상으로 연산(덧셈 또는 곱셈)을 수행한다. 연산 서버(600)는 연산 결과를 이용자의 공개키(pk)를 이용하여 암호화하고 암호화된 연산 결과를 블라인딩 서버(400)로 전송한다.

[0123] 블라인딩 서버(400)는 연산 서버(600)로부터 수신된 암호화된 연산 결과를 언블라인딩(상기 암호화된 연산 결과에서 블라인딩 값을 제거)하고, 언블라인딩된 연산 결과를 제1 단말기(100)로 전송한다(S600).

[0124] 제1 단말기(100)는 BCP.Dec 알고리즘을 수행하여 블라인딩 서버(400)로부터 수신된 암호문(언블라인딩된 연산 결과)을 복호화한다(S700). 따라서, 제1 단말기(100)는 암호화되지 않은 연산 결과를 생성할 수 있다.

[0125] 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.

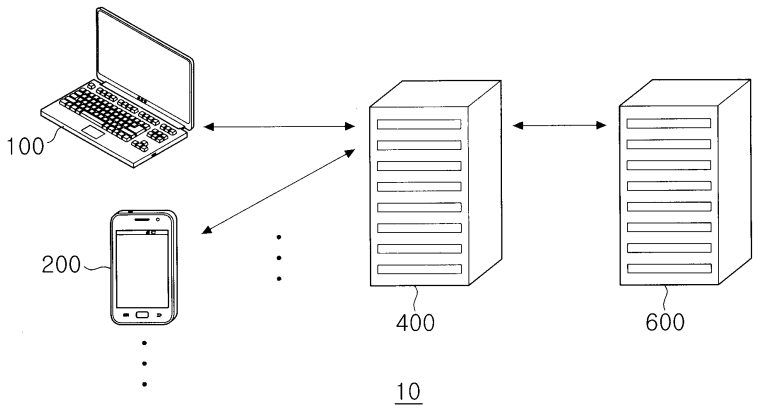
부호의 설명

[0126]

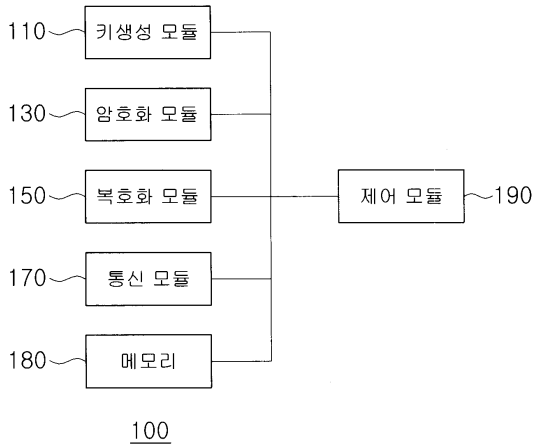
10 : 위탁 연산 시스템	100 : 제1 단말기
110 : 키생성 모듈	130 : 암호화 모듈
150 : 복호화 모듈	170 : 통신 모듈
180 : 메모리	190 : 제어 모듈
200 : 제2 단말기	400 : 블라인딩 서버
600 : 연산 서버	

도면

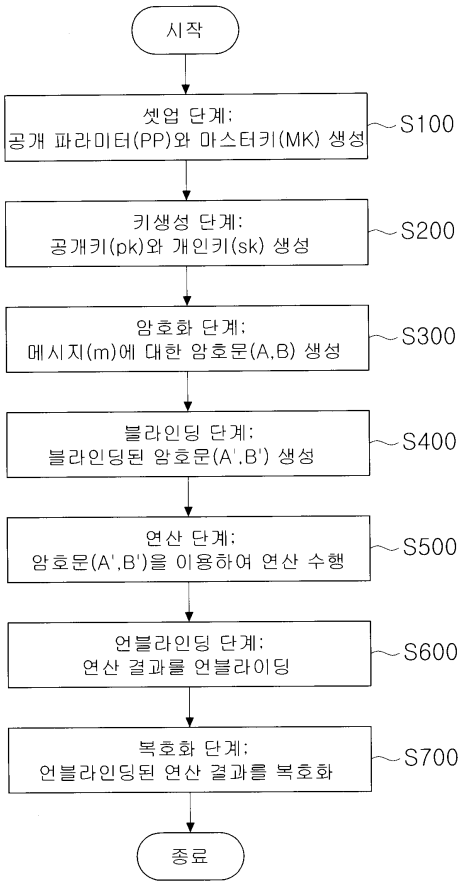
도면1



도면2


110
키생성 모듈
130
암호화 모듈
150
복호화 모듈
제어 모듈
190
170
통신 모듈
180
메모리
100

도면3


시작
셋업 단계:
공개 파라미터(PP)와 마스터키(MK) 생성
S100
키생성 단계:
공개키(pk)와 개인키(sk) 생성
S200
암호화 단계:
메시지(m)에 대한 암호문(A,B) 생성
S300
블라인딩 단계:
블라인딩된 암호문(A',B') 생성
S400
연산 단계:
암호문(A',B')을 이용하여 연산 수행
S500
언블라인딩 단계:
연산 결과를 언블라이딩
S600
복호화 단계:
언블라인딩된 연산 결과를 복호화
S700
종료