

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2012-0028432 (43) 공개일자 2012년03월23일
(51) 국제특허분류(Int. Cl.) G06F 17/10 (2006.01) (21) 출원번호 10-2010-0090249 (22) 출원일자 2010년09월15일 심사청구일자 2010년09월15일	(71) 출원인 고려대학교 산학협력단 서울 성북구 안암동5가 1 (72) 발명자 홍석희 서울특별시 도봉구 마들로 859-19, 한신아파트 103동 107호 (도봉동) 서석충 서울특별시 중랑구 망우로72가길 22 (망우동) (뒷면에 계속) (74) 대리인 특허법인충현	

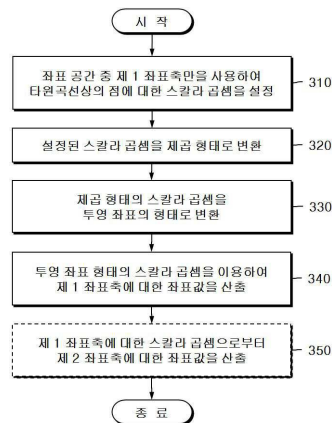
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 타원곡선암호를 위한 연산 장치 및 방법

(57) 요약

본 발명은 타원곡선을 이용한 암호를 위한 연산 장치 및 방법에 관한 것으로, 본 발명에 따른 타원곡선암호를 위한 연산 방법은 좌표 공간 중 하나의 좌표축만을 사용하여 타원곡선상의 점에 대한 스칼라 곱셈을 설정하고, 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 변환된 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하며, 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 하나의 좌표축에 대한 좌표값을 산출한다.

대표도 - 도3



(72) 발명자

조성민

서울특별시 성북구 화랑로41길 9, 2층 201호 (장위동)

권희택

서울특별시 성북구 개운사길 70, 반 2층 5호실 (안암동5가)

특허청구의 범위

청구항 1

타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈(scalar multiplication)을 설정하는 단계;

상기 설정된 스칼라 곱셈을 제곱 형태(square form)로 변환하는 단계;

상기 제곱 형태의 스칼라 곱셈을 투영 좌표(projective coordinate)의 형태로 변환하는 단계; 및

상기 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 단계를 포함하는 타원곡선암호를 위한 연산 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하는 단계를 더 포함하는 타원곡선암호를 위한 연산 방법.

청구항 3

제 2 항에 있어서,

상기 설정된 스칼라 곱셈은 상기 타원곡선상에 위치한 2개의 점들의 차이가 소정값으로 유지되는 것을 특징으로 하는 타원곡선암호를 위한 연산 방법.

청구항 4

제 1 항에 있어서,

상기 설정된 스칼라 곱셈은 타원곡선 애디션(addition) 연산과 타원곡선 더블링(doubling) 연산을 포함하고,

상기 타원곡선 애디션 연산과 상기 타원곡선 더블링 연산은 병렬적으로 수행되는 것을 특징으로 하는 타원곡선 암호를 위한 연산 방법.

청구항 5

비밀키를 입력받는 단계;

타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정하는 단계;

상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 상기 변환된 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하며, 상기 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 단계;

상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하는 단계; 및

상기 입력받은 비밀키를 이용하여 상기 제 1 좌표축 및 제 2 좌표축에 대한 좌표값으로부터 공개키를 생성하는 단계를 포함하는 타원곡선암호를 위한 연산 방법.

청구항 6

제 5항에 있어서,

상기 생성된 공개키를 이용하여 암호화한 암호문을 수신하는 단계; 및

상기 비밀키를 이용하여 상기 수신된 암호문으로부터 평문을 복호화하는 단계를 더 포함하는 타원곡선암호를 위

한 연산 방법.

청구항 7

소정 연산 방법을 이용하여 생성된 공개키를 수신하는 단계; 및

상기 수신된 공개키를 이용하여 평문으로부터 암호문을 생성하는 단계를 포함하고,

상기 소정 연산 방법은,

타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정하고,

상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 상기 변환된 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하고, 상기 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하고,

상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하며,

상기 입력받은 비밀키를 이용하여 상기 제 1 좌표축 및 제 2 좌표축에 대한 좌표값으로부터 공개키를 생성하는 것을 특징으로 하는 타원곡선암호를 위한 연산 방법.

청구항 8

제 1 항 내지 제 7 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

청구항 9

타원곡선상의 점을 표현하는 좌표값을 입력받는 입력부;

소정 연산 방법을 이용하여 상기 입력받은 좌표값으로부터 상기 타원곡선에 대한 스칼라 곱셈 결과값을 산출하는 연산부; 및

상기 산출된 스칼라 곱셈 결과값을 출력하는 출력부를 포함하고,

상기 소정 연산 방법은,

상기 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정하고,

상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고,

상기 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하며,

상기 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 것을 특징으로 하는 타원곡선암호를 위한 연산 장치.

청구항 10

제 9 항에 있어서,

상기 연산부는 상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 더 산출하는 것을 특징으로 하는 타원곡선암호를 위한 연산 장치.

청구항 11

제 10 항에 있어서,

상기 설정된 스칼라 곱셈은 상기 타원곡선상에 위치한 2개의 점들의 차이가 소정값으로 유지되는 것을 특징으로 하는 타원곡선암호를 위한 연산 장치.

청구항 12

제 9 항에 있어서,

상기 설정된 스칼라 곱셈은 타원곡선 애디션 연산과 타원곡선 더블링 연산을 포함하고,

상기 타원곡선 애디션 연산과 상기 타원곡선 더블링 연산을 수행하기 위해 각각 적어도 2개의 저장부를 더 포함하는 것을 특징으로 하는 타원곡선암호를 위한 연산 장치.

명세서

기술분야

[0001] 본 발명은 타원곡선을 이용한 암호를 위한 연산 장치 및 방법에 관한 것으로, 특히 타원곡선암호를 위해 타원곡선상에서 표현되는 점들에 대한 기하학적인 연산인 스칼라 곱셈을 처리하는 연산 장치, 방법 및 이를 기록한 기록매체에 관한 것이다.

배경기술

[0002] 암호학은 정보를 보호하기 위한 언어학적 및 수학적 방법론을 다루는 학문으로 수학을 중심으로 컴퓨터 및 통신 등 여러 학문 분야에서 공동으로 연구, 개발되고 있다. 통상적인 암호 시스템에서 송신자가 메시지의 의미를 알 수 있는 평문을 암호화하여 송신하게 되면, 수신자는 수신된 암호문을 다시 복호화하여 평문으로 복원하게 된다. 이러한 암호화 및 복호화 과정에서 사용되는 일정한 규칙 내지 방법을 암호화 키(key)라고 한다.

[0003] 키는 암호화 과정에서 사용되는 암호키와 복호화 과정에서 사용되는 복호키가 동일한 대칭 암호키(symmetric cipher key)와 암호키와 복호키가 다른 비대칭 암호키(asymmetric cipher key)로 구분될 수 있다. 암호 시스템의 안전성은 키의 길이 및 키의 안전한 관리에 의존하는데, 대칭 암호키 시스템에서는 암호키와 복호키가 동일하기 때문에 암호문의 작성자와 그 수신자가 키의 관리에 보다 주의하여야 할 필요가 있다. 즉, 대칭 암호키 시스템은 암호문의 작성자(송신자)와 수신자 모두가 폐쇄적인 사용자 그룹에 속하여 있는 것이 바람직하다. 그러나, 이러한 대칭 암호키 시스템은 키의 보안 유지에 어려움이 있을 뿐만 아니라, 해당 시스템 그룹에 속하는 사용자가 증가할수록 관리하여야 할 키의 개수가 급격하게 증가하게 된다는 불편함이 존재한다.

[0004] 이러한 한계를 보완하기 위해 등장한 암호 시스템이 비대칭 암호 시스템이다. 비대칭 암호 시스템에서 각 사용자는 두 개의 키를 부여받는다. 두 개의 키 중 하나(공개키, public key)는 공개되고, 다른 하나(비밀키, private key)는 사용자에게 의해 비밀리에 관리되어야 한다. 따라서, 비대칭 암호 시스템은 공개키 암호 시스템이라고도 한다. 이러한 비대칭 암호 시스템은 암호키와 복호키가 서로 다르기 때문에 양자의 수학적 특성에 기반하여 암호화 및 복호화 과정에서 여러 단계의 산술 연산을 수행하게 된다. 그로 인해 비대칭 암호 시스템은 대칭 암호키 시스템에 비해 그 수행 속도가 상대적으로 느리다는 약점을 갖는다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 기술적 과제는 비대칭 암호 시스템에서 암호화 및 복호화 과정에 많은 연산이 필요함으로 인해 그 수행 속도가 느려지는 한계를 극복하고, 더불어 암호화 및 복호화를 위해 많은 시스템 자원이 소모됨으로써 암호화 및 복호화 시스템의 구현 환경이 제약받는 문제점을 해결하는데 있다.

과제의 해결 수단

[0006] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 타원곡선암호를 위한 연산 방법은 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈(scalar multiplication)을 설정하는 단계; 상기 설정된 스칼라 곱셈을 제곱 형태(square form)로 변환하는 단계; 상기 제곱 형태의 스칼라 곱셈을 투영 좌표(projective coordinate)의 형태로 변환하는 단계; 및 상기 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 단계를 포함한다.

[0007] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 타원곡선암호를 위한 연산 방법은 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하는 단계를 더 포함한다.

[0008] 상기 다른 기술적 과제를 해결하기 위하여, 본 발명에 따른 타원곡선암호를 위한 연산 방법은 비밀키를 입력받는 단계; 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을

설정하는 단계; 상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 상기 변환된 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하며, 상기 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 단계; 상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하는 단계; 및 상기 입력받은 비밀키를 이용하여 상기 제 1 좌표축 및 제 2 좌표축에 대한 좌표값으로부터 공개키를 생성하는 단계를 포함한다. 또한, 상기 타원곡선암호를 위한 연산 방법은 생성된 공개키를 이용하여 암호화한 암호문을 수신하는 단계; 및 상기 비밀키를 이용하여 상기 수신된 암호문으로부터 평문을 복호화하는 단계를 더 포함할 수 있다.

[0009] 상기 또 다른 기술적 과제를 해결하기 위하여, 본 발명에 따른 타원곡선암호를 위한 연산 방법은 소정 연산 방법을 이용하여 생성된 공개키를 수신하는 단계; 및 상기 수신된 공개키를 이용하여 평문으로부터 암호문을 생성하는 단계를 포함하고, 상기 소정 연산 방법은, 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정하고, 상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 상기 변환된 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하고, 상기 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하고, 상기 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 상기 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출하며, 상기 입력받은 비밀키를 이용하여 상기 제 1 좌표축 및 제 2 좌표축에 대한 좌표값으로부터 공개키를 생성하는 것이 바람직하다.

[0010] 또한, 이하에서는 상기 기재된 타원곡선암호를 위한 연산 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

[0011] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 타원곡선암호를 위한 연산 장치는 타원곡선상의 점을 표현하는 좌표값을 입력받는 입력부; 소정 연산 방법을 이용하여 상기 입력받은 좌표값으로부터 상기 타원곡선에 대한 스칼라 곱셈 결과값을 산출하는 연산부; 및 상기 산출된 스칼라 곱셈 결과값을 출력하는 출력부를 포함하고, 상기 소정 연산 방법은, 상기 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정하고, 상기 설정된 스칼라 곱셈을 제곱 형태로 변환하고, 상기 제곱 형태의 스칼라 곱셈을 투영 좌표의 형태로 변환하며, 상기 투영 좌표 형태의 스칼라 곱셈을 이용하여 상기 제 1 좌표축에 대한 좌표값을 산출하는 것이 바람직하다.

발명의 효과

[0012] 본 발명은 변환된 스칼라 곱셈을 이용하여 타원곡선상의 점에 대한 좌표값을 산출함으로써 비대칭 암호 시스템의 암호화 및 복호화 과정에 상대적으로 적은 수의 연산이 필요하게 되고, 그로 인해 그 수행 속도가 향상되며, 더불어 암호화 및 복호화를 위해 요구되는 시스템 자원을 감소시킴으로써, 스마트 카드(smart cards)와 같이 보다 열악한 컴퓨팅 파워를 갖는 환경에서도 효과적이고 신뢰도 높은 암호화 및 복호화 시스템을 구현할 수 있다.

도면의 간단한 설명

[0013] 도 1은 공개키 암호 시스템에서 평문을 암호화하고 이를 다시 복호화하는 과정을 설명하기 위한 도면이다.

도 2는 타원곡선암호의 알고리즘을 설명하기 위해 예시한 몽고메리 래더 알고리즘(Montgomery ladder algorithm)의 의사 코드(pseudo code)이다.

도 3은 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 방법을 도시한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 방법에서 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 제 2 좌표축에 대한 좌표값을 복원하는 방법을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 방법을 설명하기 위한 알고리즘을 예시한 의사 코드이다.

도 6은 본 발명의 다른 실시예에 따른 타원곡선암호를 위한 연산 방법에 대해 각각 암호문 송신자 및 암호문 수신자 측의 처리 과정을 설명하기 위한 도면이다.

도 7은 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 장치를 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 발명의 실시예들을 구체적으로 설명하기에 앞서, 본 발명의 실시예들이 구현될 수 있는 대표적인 환경인 공

개키 암호 시스템에 대해 도 1을 참조하여 개괄적으로 소개하고자 한다. 도 1은 공개키 암호 시스템에서 평문을 암호화하고 이를 다시 복호화하는 과정을 설명하기 위한 도면으로서, 공개키 암호 시스템은 암호화 측을 나타내는 송신자(10), 복호화 측을 나타내는 수신자(20) 및 공개키 관리 시스템(30)을 포함한다.

- [0015] 앞서 간단히 소개하였듯이, 공개키 암호 시스템에서 각 사용자는 비밀키(41) 및 공개키(42)의 두 개의 키를 부여받는다. 공개키 암호 시스템에서 각 사용자는 자신의 비밀키만을 관리하면 되므로 키 관리의 어려움을 감소시킬 수 있다. 여기서, 키 교환시에는 공개키를 암호화키로 사용하여 일반에게 공개하게 되고, 비밀키를 이용하여 복호화한다고 가정하자. 통상적으로 공개키 암호 시스템은 상기된 바와 같은 구조를 갖는 것이 일반적이나, 전자서명시에는 반대로 이용될 수도 있다. 이하에서 도 1의 실시예에서는 수신자(20)를 기준으로 공개키와 비밀키를 설명하도록 하겠다.
- [0016] 공개키 관리 시스템(30)은 공개키 디렉터리라고도 불리며, 공개키 암호 시스템을 이용하는 사용자는 이러한 관리 시스템(30)에 자유롭게 접근하여 다른 사용자의 공개키를 열람할 수 있다. 한편, 공개키 관리 시스템(30)의 물리적인 위치는 구현의 편의에 따라 송신자(10)나 수신자(20) 측에 위치할 수도 있고, 도 1에 도시된 바와 같이 별도의 공간에 위치할 수도 있다. 도 1에서 공개키 관리 시스템(30)에는 다수의 공개키들이 저장될 수 있으며, 수신자(20)의 비밀키(41)에 대응하는 공개키(42) 역시 저장되어 있음을 볼 수 있다.
- [0017] 우선, 송신자(10)는 공개키 관리 시스템(30)에 접근하여 수신자(20)의 공개키(42)를 이용하여 평문을 암호화함으로써 암호문을 생성한다. 이렇게 생성된 암호문은 수신자(20)에게 전달된다. 다음으로 수신자(20)는 수신한 암호문을 자신의 비밀키(41)를 이용하여 복호화한다. 즉, 수신자(20)는 암호문을 해독하여 평문을 복원한다.
- [0018] 이상의 암호 시스템의 구조에서 비록 공개키 관리 시스템(30)에는 다양한 사용자의 공개키들이 저장되어 있고, 이를 이용하여 누구나 자유롭게 암호문을 생성할 수 있지만, 생성된 암호문을 평문으로 복호화할 수 있는 것은 해당 암호화 과정에 사용된 공개키에 대응하는 비밀키를 소유한 사용자만이 가능하다. 예를 들어, 도 1의 예에서 수신자(20) 이외의 제 3자가 송신자(10)가 생성한 암호문을 중간에서 가로채더라도 수신자(20) 고유의 비밀키가 없는 이상 암호문을 복호화할 수 없다.
- [0019] 이상과 같이 공개키 암호 시스템에서 각 사용자에게 부여되는 공개키와 비밀키에는 수학적 연관이 있기 때문에 암호화 및 복호화가 가능하다. 즉, 공개키 암호 방식은 그 설계자만이 알고 있는 특정한 종류의 정보 없이는 매우 풀기 어려운 수학적 문제를 바탕으로 만들어진 것이다. 키를 만드는 사람은 이 문제(공개키)를 일반에 공개하고 특정한 정보(비밀키)는 자신만이 알 수 있도록 숨긴다. 그러면 어떤 사람이든 이 문제를 이용해 메시지를 암호화하더라도 키를 만든 사람만이 이 문제를 풀어 원래의 메시지를 해독할 수 있다.
- [0020] 본 발명의 실시예들은 이러한 공개키 암호 시스템에 기반하면서도 특히 이러한 수학적 문제에 타원곡선암호라는 방식을 채용하고 있다. 타원곡선암호(Elliptic Curve Cryptography, ECC)는 타원곡선 이론에 기반한 공개키 암호 방식의 일종으로, 유한체(finite field)에서 해법이 어렵다고 알려진 타원곡선 이산로그문제(Elliptic Curve Discrete Logarithm Problem, ECDLP)를 바탕으로 만들어진 암호이다. 타원곡선 이산로그문제는 타원곡선상의 임의의 한 점 P에 정수 k를 곱한 값이 Q=kP일 때, 점 Q와 P를 알고 있더라도 정수 k를 계산하기 어려움을 의미한다. 따라서, 타원곡선암호 시스템의 핵심 연산은 스칼라 곱셈(scalar multiplication), 즉 Q=kP를 구하는 것이며, 이러한 스칼라 곱셈은 타원곡선암호 시스템의 안전성과 효율성에 가장 큰 영향을 준다.
- [0021] 공간 상에서 타원곡선은 (x, y)의 점들로 구성되어지고, x 및 y 좌표간의 관계로 표현된 타원곡선의 형태를 갖는다. 스칼라 곱셈을 연산은 동일한 두 점의 합을 구하는 더블링(doubling) 연산과 서로 다른 두 점의 합을 구하는 애디션(addition) 연산을 반복적으로 이용함으로써 연산이 수행된다. 그런데, 타원곡선 스칼라 곱셈에 사용되는 타원곡선 더블링(ECDBL) 연산과 타원곡선 애디션(ECADD) 연산은 키 비트(key bit)에 따라 실행되는 연산에 차이가 생기기 때문에 단순전력분석(Simple Power Analysis, SPA)에 의해 분석이 많이 진행되었다.
- [0022] 타원곡선 스칼라 곱셈 알고리즘에 대한 단순전력분석이 진행되면서 이를 방어하기 위한 많은 대응방법들이 제안되었다. 이러한 대응방법들 중, 몽고메리 알고리즘(Montgomery algorithm)은 몽고메리에 의해서 처음으로 제안된 알고리즘으로 몽고메리 폼(Montgomery form) 상에서의 스칼라 곱셈 연산을 x 좌표만을 사용하여 연산함으로써 연산을 빠르게 할 수 있는 방법이다. 그 이후, Lopez and Dahab과 Izu and Takagi등에 의해서 일반적인 바이어슈트라스 폼(Weierstrass form)에서의 x 좌표만을 이용한 몽고메리 래더 알고리즘(Montgomery ladder algorithm)의 연산식이 제안되었다. 몽고메리 래더 알고리즘은 x 좌표만을 사용함으로써 타원곡선에서의 스칼라 곱셈을 빠르고 효율적으로 할 수 있고, 포인트(point) 압축면에서도 매우 효율적인 방법으로 알려져 있다.
- [0023] 도 2는 타원곡선암호의 알고리즘을 설명하기 위해 예시한 몽고메리 래더 알고리즘의 의사 코드(pseudo code)로

서, 다음과 같은 단계로 수행된다.

[0024] 우선 입력 값으로 타원곡선상의 한 점 P와 상수(스칼라를 의미한다.) d를 입력받는다. 도 2의 몽고메리 래더 알고리즘은 매 루프(loop)마다 타원곡선 더블링(ECDBL) 연산과 타원곡선 애디션(ECADD) 연산이 반복적으로 수행된다. 따라서, 몽고메리 래더 알고리즘은 스칼라 곱셈의 매 루프마다 동일한 연산을 수행하므로 단순전력분석에 안전할 수 있다. 이러한 몽고메리 래더 알고리즘은 바이어슈트라스 폼에서 점 P, Q와 P-Q 세 점의 X 좌표만을 이용하여 P+Q의 x 좌표를 구할 수 있다는 특징이 있다.

[0025] 그러나, 이러한 타원곡선 스칼라 곱셈을 위해 이상에서 제안된 다양한 방법들 역시 많은 연산이 필요함으로 인해 수행 속도가 느려지고, 과도한 시스템 자원을 요구한다는 문제점이 여전히 내재되어 있다. 따라서, 본 발명의 실시예들은 스칼라 연산에 필요한 연산 수와 각각의 연산에 요구되는 자원을 줄일 수 있는 방법을 제안하고자 한다. 이하에서 도면을 참조하여 본 발명의 다양한 실시예들을 상세하게 설명한다.

[0026] 도 3은 본 발명의 일 실시예에 따른 타원곡선압호를 위한 연산 방법을 도시한 흐름도이다. 설명의 편의를 위해 이하의 본 발명의 실시예들은 타원곡선 함수로서 자코비 4차 형태(Jacobi quartic form)의 타원곡선을 예시하고 있다. 자코비 4차 형태의 타원곡선은 x의 차수가 4차인 다음의 수학식 1과 같이 표현될 수 있으며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 이러한 자코비 4차 타원곡선 이외에 이에 상응하는 다양한 타원곡선이 본 발명의 기술적 사상을 구현하는데 활용될 수 있음을 알 수 있다.

수학식 1

$$E(F_p): y^2 = x^4 + 2ax^2 + 1 \quad (a \in F_p, a^2 \neq 1)$$

[0027]

[0028] 즉, $\text{char}(F_p) \neq 2, 3$ 이고 소수 p를 선택하였다고 가정하면, 유한체 F_p 상의 자코비 4차 타원곡선은 상기된 수학식 1과 같이 정의될 수 있다. 이 타원곡선 위의 점 $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ 및 $P_3 = P_1 + P_2 = (x_3, y_3)$ 과 두 점의 차가 다음의 수학식 2와 같이 유지되면 이하에서 소개될 수학식 3 내지 수학식 5를 만족한다.

수학식 2

$$P = P_1 - P_2 = (x, y)$$

[0029]

[0030] 특히 수학식 2는 몽고메리 래더 알고리즘에 관한 도 2의 알고리즘 역시 만족하는 것을 나타내고 있으며, 매 루프(loop)마다 두 개의 점의 차는 일정한 값 P를 유지한다.

[0031] 이상의 내용에 기초하여 도 3을 순서대로 설명하면 다음과 같다.

[0032] 310 단계에서 타원곡선상의 점을 표현하는 좌표 공간 중 제 1 좌표축만을 사용하여 상기 점에 대한 스칼라 곱셈을 설정한다. 본 발명의 실시예에서는 자코비 4차 타원곡선에서 몽고메리 래더 알고리즘을 적용함에 있어서, x^2 좌표만을 이용하여 스칼라 곱셈 연산을 수행한다. 즉, (x, y)의 2차원 좌표 공간에서 하나의 좌표축인 x 좌표축의 좌표값만을 이용하게 된다.

[0033] 보다 구체적으로 스칼라 곱셈에서 활용되는 애디션 연산과 더블링 연산은 각각 다음의 수학식 3과 같이 정의된다.

수학식 3

$$\text{Addition : } x_3 = \frac{x_1 y_2 + y_1 x_2}{1 - x_1^2 x_2^2}, \quad y_3 = \frac{(y_1 y_2 + 2a x_1 x_2)(x_1^2 x_2^2 + 1) + 2x_1 x_2(x_1^2 + x_2^2)}{(1 - x_1^2 x_2^2)^2}$$

$$\text{Doubling : } x_3 = \frac{2x_1 y_1}{1 - x_1^4}, \quad y_3 = \left(\frac{x_1^2 + 1}{1 - x_1^4} \right)^2 (x_1^4 + 2a x_1^2 + 1 + y_1^2) - x_3^2 - 1$$

[0034]

[0035]

320 단계에서는 310 단계를 통해 설정된 스칼라 곱셈을 제곱 형태(square form)로 변환한다. 즉, 상기 수학식 3을 x 좌표의 제곱에 관한 식으로 정리하면 다음의 수학식 4와 같다.

수학식 4

$$\text{Addition : } x_3^2 = \frac{4x_1^2(x_1^4 + 2a x_1^2 + 1)}{(1 - x_1^4)^2}$$

$$\text{Doubling : } x_3^2 = \frac{(x_1^2 - x_2^2)^2}{x^2(1 - x_1^2 x_2^2)^2}$$

[0036]

[0037]

이러한 변환을 통해 x^2 좌표를 이용하여 스칼라 곱셈 연산의 주요 연산인 애디션 연산과 더블링 연산을 표현할 수 있게 되었다. 이와 같이 x^2 좌표를 이용한 표현 방법을 사용할 경우 기존의 x 좌표를 이용한 스칼라 곱셈 연산보다 제곱 연산이 줄어드는 효과를 얻을 수 있다. 타원곡선에서 스칼라 곱셈 과정에서 두 점의 합을 계산하기 위해서는 이상의 수학식 3 내지 수학식 4에서 볼 수 있듯이 곱셈, 나눗셈 및 덧셈 연산이 필요하다. 이 중에서 가장 많은 시간과 자원이 소모되는 연산은 나눗셈 연산에 해당한다. 왜냐하면 이러한 나눗셈 연산에는 역원(inverse) 연산이 포함되기 때문이다. 따라서, 타원곡선암호 방법에서 스칼라 곱셈의 성능을 향상시키기 위해서는 연산의 효율을 위해 이러한 역원 연산을 가능한 배제시킬 필요가 있다.

[0038]

330 단계에서는 320 단계를 통해 변환된 제곱 형태의 스칼라 곱셈을 $(X^2 : Z^2)$ 의 투영 좌표(projective coordinate)의 형태로 변환한다. 이러한 투영 좌표 변환은 나눗셈 연산(역원 연산)을 소거시키는 역할을 한다.

따라서, 상기 수학식 4는 투영 좌표계 $(X/Z, Y/Z^2)$ 에 의해 다음의 수학식 5와 같이 표현될 수 있다.

수학식 5

$$\text{ECADD}^{x^2} : X_3^2 = Z^2((A+B)/2)^2, \quad Z_3^2 = X^2((A-B)/2)^2$$

$$\text{where } A = (X_1^2 + Z_1^2)(Z_2^2 - X_2^2), \quad B = (X_1^2 - Z_1^2)(Z_2^2 + X_2^2)$$

$$\text{ECDBL}^{x^2} : X_3^2 = A((X_1^2 + Z_1^2) + (a-1)A/2), \quad Z_3^2 = (X_1^2 + Z_1^2)^2(X_1^2 - Z_1^2)^2$$

$$\text{where } A = 4X_1^2 Z_1^2 = (X_1^2 + Z_1^2)^2 - (X_1^2 - Z_1^2)^2$$

[0039]

[0040]

이 때, 타원곡선 애디션 연산 ECADD^{x^2} 의 연산량은 '4M+2S'로 표현될 수 있으며, 타원곡선 더블링 연산

ECDBL^{x^2} 의 연산량은 '2M+2S+1C'로 표현될 수 있다. 여기서, M은 곱셈(multiplication)을 의미하고, S는 제곱(square)을 의미하며, C는 상수(constant)를 의미한다.

[0041] 340 단계에서는 330 단계를 통해 변환된 투영 좌표 형태의 스칼라 곱셈을 이용하여 제 1 좌표축에 대한 좌표값을 산출한다. 즉, 340 단계는 kP의 x 좌표의 제공값을 산출한다.

[0042] 이상에서 기술한 310 단계 내지 340 단계는 설정된 스칼라 곱셈을 보다 적은 연산을 통해 수행 가능하도록 변환하고, 그 변환식에 따른 연산 결과를 산출하는 과정에 해당하므로, 전자적 형태의 정보 및 수학식을 처리할 수 있는 처리기(processor) 및 이러한 연산에 필요한 기억공간(memory)을 통해 구현될 수 있으며, 필요에 따라서는 처리기와 기억공간 간의 데이터 처리를 적절하게 제어할 수 있는 제어기(controller)가 활용될 수도 있을 것이다. 이러한 처리기, 기억공간 및 제어기는 본 발명이 속하는 기술분야의 활용 환경이나 동작 환경을 고려하여 통상의 기술자에 의해 적절하게 선택될 수 있을 것이다. 나아가, 이러한 스칼라 곱셈의 설정 및 변환 과정에는 이상에서 예시된 하드웨어들을 제어하기 위한 부가적인 소프트웨어 코드(code)도 활용될 수 있을 것이다.

[0043] 한편, 350 단계에서는 상기된 과정에 이어 y 좌표를 복원하는 과정을 기술하고 있는 것으로 이러한 350 단계는 추가적으로 수행될 수 있다. 이상의 310 단계 내지 340 단계를 통해 본 발명의 실시예는 제안된 스칼라 곱셈을 통해 kP의 x 좌표의 제공값을 얻을 수 있다. 그런데, 타원곡선암호 시스템에서는 암호학적으로 x 좌표뿐만 아니라, y 좌표의 값이 필요하기 때문에 이를 복원할 필요가 있다.

[0044] 따라서, 350 단계에서는 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 좌표 공간상의 제 2 좌표축에 대한 좌표값을 산출한다. 즉, 제 1 좌표축인 x 축에 대한 스칼라 곱셈으로부터 제 2 좌표축인 y 축에 대한 좌표값을 산출한다. 보다 구체적으로 y 좌표값을 복원하기 위해서 다음의 수학식 6이 사용된다.

수학식 6

$$ECADD^x : X_3 = Z((A+B)/2), Z_3 = X((A-B)/2) \\ \text{where } A = (X_1^2 + Z_1^2)(Z_2^2 - X_2^2), B = (X_1^2 - Z_1^2)(Z_2^2 + X_2^2)$$

[0045]

[0046] 수학식 6은 y 좌표를 복원하는 데 필요한 연산량을 줄이기 위해 도입된 것으로, $ECADD^x$ 의 연산량은 '4M+4S'이다.

[0047] 이제 도 4를 참조하여 y 좌표를 복원하는 방법이 제시된다. 도 4는 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 방법에서 제 1 좌표축에 대한 투영 좌표 형태의 스칼라 곱셈으로부터 제 2 좌표축에 대한 좌표값을 복원하는 방법을 설명하기 위한 도면이다.

[0048] 도 4에서도 역시 자코비 4차 타원곡선을 예시하고 있으며, 타원곡선 상의 점 $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ 및 $P_3 = P_1 + P_2 = (x_3, y_3)$ 과 두 점의 차 $P = P_1 - P_2 = (x, y)$ 는 일정한 값으로 유지된다. 즉, 타원곡선상에 위치한 2개의 점들의 차이가 P로서 유지되는 특징을 갖는다.

[0049] 도 4를 참조하면, 우선 몽고메리 래더 알고리즘에서 $i=1$ 일 때의 루프 종료 후, $Q[0]=((k-k_0)/2)P$ 이고, $Q[1]=((k-k_0)/2)P+P$ 를 얻을 수 있다.

[0050] 그런 다음, $ECADD^{x^2}$ 를 이용하여 $Q[0]$, $Q[1]$ 및 $-P=(-x, y)$ 로부터 $Q[0]-P$ 를 산출한다. 이어서, $ECADD^{x^2}$ 를 이용하여 $Q[0]-P$, $Q[1]$ 및 $2P$ 로부터 $2Q[0]=(k-k_0)P$ 를 산출하고, $Q[0]$, $Q[1]$ 및 P 로부터 $2Q[0]+P=(k-k_0)P+P$ 를 각각 산출한다.

[0051] 그 결과, kP는 다음과 같이 산출된다. 만약 $k_0=0$ (마지막 비트가 0임을 의미한다.)이라면 $2Q[0]=kP-k_0P$ 로서 y 좌표를 복원하고, 만약 $k_0=1$ (마지막 비트가 1임을 의미한다.)이라면 $2Q[0]=kP-k_0P+P$ 로서 y 좌표를 복원한다.

[0052] 이러한 복원 절차를 통해 y 좌표를 산출하는 방법은 다음의 수학식 7과 같이 표현된다.

수학식 7

$$Y_1 = \frac{X_3 - X_1 Z_1 Y_2}{X_2 Z_2}$$

[0053]

[0054]

상기 수학식 7의 연산량은 '1I+4M'과 같이 표현될 수 있다. 여기서 I는 역원(inverse)를 의미한다.

[0055]

도 5는 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 방법을 설명하기 위한 알고리즘을 예시한 의사 코드로서, 제곱 형태 및 투영 좌표의 형태로 변환된 스칼라 곱셈을 이용하여 타원곡선상의 점에 대한 x 좌표값

을 산출하고, $ECADD^{x^2}$, $ECDBL^{x^2}$ 및 $ECADD^x$ 를 사용하여 y 좌표를 복원하는 본 발명의 실시예를 전체적으로 도시하고 있다.

[0056]

도 5에서 본 발명의 예시 알고리즘은 타원곡선상의 점 P와 스칼라 d를 입력받는다. 이제 본 알고리즘은 for 구

문의 루프를 통해 각각 x 좌표의 제곱값에 대한 타원곡선 더블링 연산($ECDBL^{x^2}$)과 타원곡선 애디션 연산($ECADD^{x^2}$)을 수행한다. 여기서, 타원곡선 애디션 연산과 타원곡선 더블링 연산은 각각 다른 저장공간을 활용하여 각각 다른 점들에 대한 연산을 수행하므로 병렬적으로 수행될 수 있다.

[0057]

이어서 도 5의 알고리즘은 line 8, 9를 통해 ECADD 연산이 (x,z)의 값을 산출하므로 '4M'의 연산량을 갖는다. 마지막으로 $ECRE^{(x,y)}$ 연산을 통해 y 좌표를 산출한다.

[0058]

상기된 본 발명의 실시예들에 따르면 제곱 형태 및 투영 좌표의 형태로 변환된 스칼라 곱셈을 이용하여 타원곡선상의 점에 대한 좌표값을 산출함으로써 비대칭 암호 시스템의 암호화 및 복호화 과정에 상대적으로 적은 수의 연산이 필요하게 되고, 그로 인해 그 수행 속도가 향상된다. 더불어 암호화 및 복호화를 위해 요구되는 시스템 자원을 감소시킴으로써, 낮은 컴퓨팅 파워를 갖는 환경에서도 효과적이고 신뢰도 높은 암호화 및 복호화 시스템을 구현하는 것이 가능하다.

[0059]

이렇게 개선된 본 발명의 실시예들에서 제안하는 타원곡선암호를 위한 연산 방법을 기존에 알려진 연산 방법과 비교하면 다음과 같다.

표 1

Jacobi 4차 곡선 알고리즘	필요공간	연산량	M(S=0.8M)
Billet & Joye	3	10M+ 3S+ 1C	12.4M
Bernstein & Lange	6	8M+ 3S+ 1C	10.4M
Hisil et al.(Z=1)	6	7M+ 3S+ 1C(6M+ 3S+ 1C)	9.4M(8.4M)
$ECADD^{x^2}$	2	4M+ 2S(3M+ 2S)	5.6M(4.6M)

[0060]

상기 표 1에서 본 발명의 실시예에서 제안하고 있는 타원곡선 애디션 연산($ECADD^{x^2}$)을 수행하기 위해 필요공간(데이터 저장을 위한 레지스터를 의미한다.)이 2로 감소하였고, 연산량도 감소함으로 인해 연산 속도가 향상될 수 있음을 알 수 있다.

[0061]

표 2

Jacobi 4차 곡선 알고리즘	필요공간	연산량	M(S=0.8M)
Bernstein & Lange	3	1M+ 9S+ 1C	8.2M
Hisil et al.(Z=1)	6	3M+ 4S	6.2M
Hisil et al.(Z=1)	6	2M+ 5S+ 1C	6M
$ECDBL^{x^2}$	2	2M+ 2S+ 1C	3.6M

[0062]

[0063] 이어서, 상기 표 2에서도 본 발명의 실시예에서 제안하고 있는 타원곡선 더블링 연산($ECDBL^{x^2}$)을 수행하기 위해 필요공간(데이터 저장을 위한 레지스터를 의미한다.)이 2로 감소하였고, 연산량도 감소함으로 인해 연산 속도가 향상될 수 있음을 알 수 있다.

[0064] 이러한 필요공간의 감소는 앞서 설명한 도 5에 예시된 알고리즘을 참조하면 보다 분명히 확인할 수 있다. 즉, 도 4에서 타원곡선 애디션 연산은 각각 Q[0], Q[1]을 저장하기 위한 레지스터(register)를 사용하게 되고, 타원곡선 더블링 연산은 각각 Q[0], Q[2]을 저장하기 위한 레지스터(register)를 사용하게 된다.

[0065] 이제 이상에서 제안한 타원곡선암호를 위한 연산 방법을 활용하여 실질적인 공개키 암호 시스템에 적용하는 실시예들을 제안하고자 한다.

[0066] 도 6은 본 발명의 다른 실시예에 따른 타원곡선암호를 위한 연산 방법에 대해 각각 암호문 송신자 및 암호문 수신자 측의 처리 과정을 설명하기 위한 도면으로, 시간 순서에 따라 암호화 및 복호화 과정에서 이상에서 제안된 스칼라 곱셈 연산이 활용되는 방법을 도시하고 있다.

[0067] 수신자(20)의 측면에서의 활용 과정은 다음과 같다. 우선, 21 단계에서 수신자(20)는 비밀키를 입력받는다. 이러한 입력은 수신자(20)가 직접 입력할 수도 있으며, 필요에 따라서는 수신자(20)의 암호 시스템에서 자동으로 생성할 수도 있을 것이다.

[0068] 다음으로 22 단계에서는 변환된 스칼라 곱셈을 이용하여 타원곡선상의 점에 대한 제 1 좌표축 및 제 2 좌표축에 대한 좌표값을 산출한다. 여기서 변환된 스칼라 곱셈이란, 앞서 설명한 제곱 형태의 변환 및 투영 좌표의 형태로 변환된 스칼라 곱셈을 의미한다. 즉, 22 단계는 본 발명의 실시예들을 통해 제안한 스칼라 곱셈을 이용하여 x, y 좌표값을 산출한다.

[0069] 23 단계에서 수신자(20)는 21 단계를 통해 입력받은 비밀키를 이용하여 22 단계를 통해 산출한 제 1 좌표축 및 제 2 좌표축에 대한 좌표값으로부터 공개키를 생성한다. 이렇게 생성된 공개키는 앞서 도 1을 통해 설명한 공개키 관리 시스템에 등록될 수 있으며, 이후에 암호문을 작성하고자 하는 송신자(10)에 의해 열람될 수 있다.

[0070] 이제, 송신자(10)의 측면으로 넘어오면, 송신자(10)는 11 단계를 통해 수신자(20)가 생성한 공개키를 획득(공개키 열람을 의미한다.)한다. 다음으로 12 단계에서 송신자(10)는 이러한 공개키를 이용하여 평문으로부터 암호문을 생성한다. 이렇게 생성된 암호문은 수신자(20)의 비밀키로만 복원이 가능하며, 설사 제 3자에 의해 획득되더라도 복호화가 불가능하다.

[0071] 다시 수신자(20)의 측면으로 돌아오면, 수신자(20)는 24 단계에서 송신자(10)에 의해 암호화된 암호문을 수신한다. 마지막으로 수신자(20)는 25 단계에서 자신의 비밀키를 이용하여 암호문으로부터 평문을 복호화한다.

[0072] 상기된 본 발명의 다른 실시예에 따르면 공개키 암호 시스템의 암호화 및 복호화 과정에 상대적으로 적은 수의 연산이 필요하게 되고, 그로 인해 그 수행 속도가 향상되며, 적은 시스템 자원만으로도 타원곡선 암호화 시스템을 구현하는 것이 가능하다.

[0073] 도 7은 본 발명의 일 실시예에 따른 타원곡선암호를 위한 연산 장치(700)를 도시한 도면으로 입력부(721), 연산부(722) 및 출력부(723)를 포함한다. 또한, 스칼라 곱셈을 실질적으로 구현하는 애디션 연산과 더블링 연산을 위해 선택적으로 저장부(724)를 포함할 수 있다.

[0074] 입력부(721)는 타원곡선상의 점을 표현하는 좌표값을 입력받는다. 이러한 입력부(721)는 전자적인 형태의 데이

터를 수신할 수 있는 입력 장치로서 구현될 수 있으며, 필요에 따라 물리적인 인터페이스를 포함할 수 있다.

[0075] 연산부(722)는 본 발명의 실시예들이 제안하는 변환된 스칼라 곱셈 연산 방법을 이용하여 입력부(721)를 통해 입력받은 좌표값으로부터 타원곡선에 대한 스칼라 곱셈 결과값을 산출한다. 이러한 연산부(722)는 앞서 도 3을 통해 설명한 310 단계 내지 350 단계에 대응하는 것으로 여기서는 구체적인 설명을 생략한다.

[0076] 다만 본 실시예의 연산 장치를 하드웨어로 구현함에 있어서, 연산부(722)는 열거된 연산을 수행하기 위한 모듈러 곱셈기(modular multiplier), 모듈러 덧셈기(adder) 및 뺄셈기(subtractor)의 조합으로 구현될 수 있다. 이러한 하드웨어의 설계는 본 발명이 속하는 기술분야에서 회로 설계에 관한 통상의 지식을 가진 자에 의해 구현 가능한 것이다.

[0077] 출력부(723)는 연산부(722)를 통해 산출된 스칼라 곱셈 결과값을 출력한다. 이러한 출력부(723)는 전자적인 형태로 가공된 데이터를 출력할 수 있는 입력 장치로서 구현될 수 있으며, 필요에 따라 물리적인 인터페이스를 포함할 수 있다.

[0078] 한편, 연산부(722)에서 설정된 스칼라 곱셈은 타원곡선 애디션 연산과 타원곡선 더블링 연산을 포함하므로, 타원곡선 애디션 연산과 타원곡선 더블링 연산을 수행하기 위해 각각 적어도 2개의 저장부(724)를 활용할 수 있다. 앞서 표 1 및 표 2를 통해 설명한 바와 같이 스칼라 곱셈의 2가지 연산은 각각 2개의 레지스터를 필요로 한다. 나아가 본 실시예의 연산 장치에는 상기된 하드웨어 장치들을 제어하기 위한 제어기(controller)와 입출력 데이터들을 전송하기 위한 데이터 버스(bus)가 포함될 수 있다.

[0079] 상기된 본 발명의 실시예에 따르면 스칼라 곱셈 연산 과정에 상대적으로 적은 수의 연산이 필요하게 되고, 그로 인해 그 수행 속도가 향상되며, 적은 시스템 자원만으로도 타원곡선 암호화 시스템을 구현하는 것이 가능하다.

[0080] 한편, 본 발명은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

[0081] 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현하는 것을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.

[0082] 이상에서 본 발명에 대하여 그 다양한 실시예들을 중심으로 살펴보았다. 본 발명에 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. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

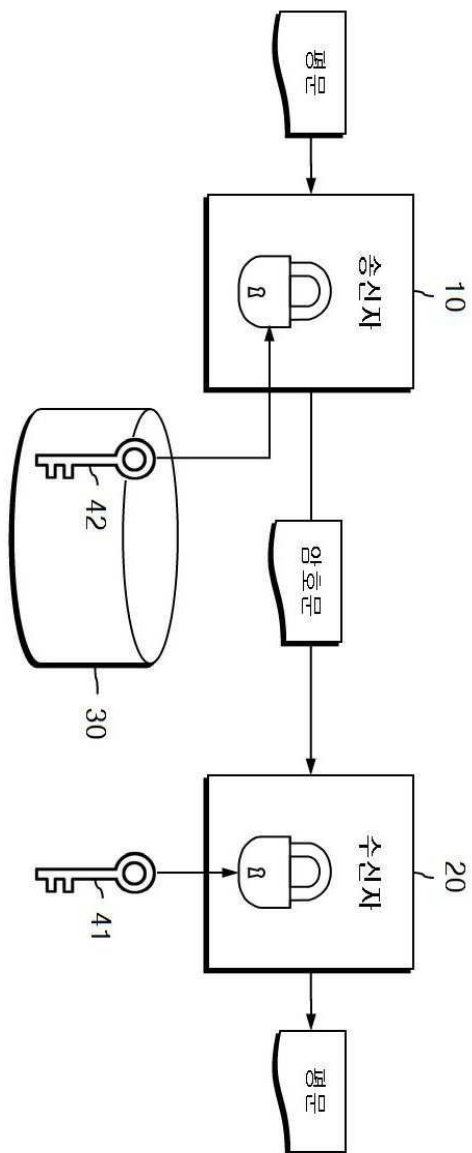
부호의 설명

[0083]

10 : 송신자 (암호화 측)	20 : 수신자 (복호화 측)
30 : 공개키 관리 시스템 (공개키 디렉터리)	
41 : 비밀키	42 : 공개키
700 : 타원곡선암호화 장치	
721 : 입력부	722 : 연산부
723 : 출력부	724 : 저장부

도면

도면1



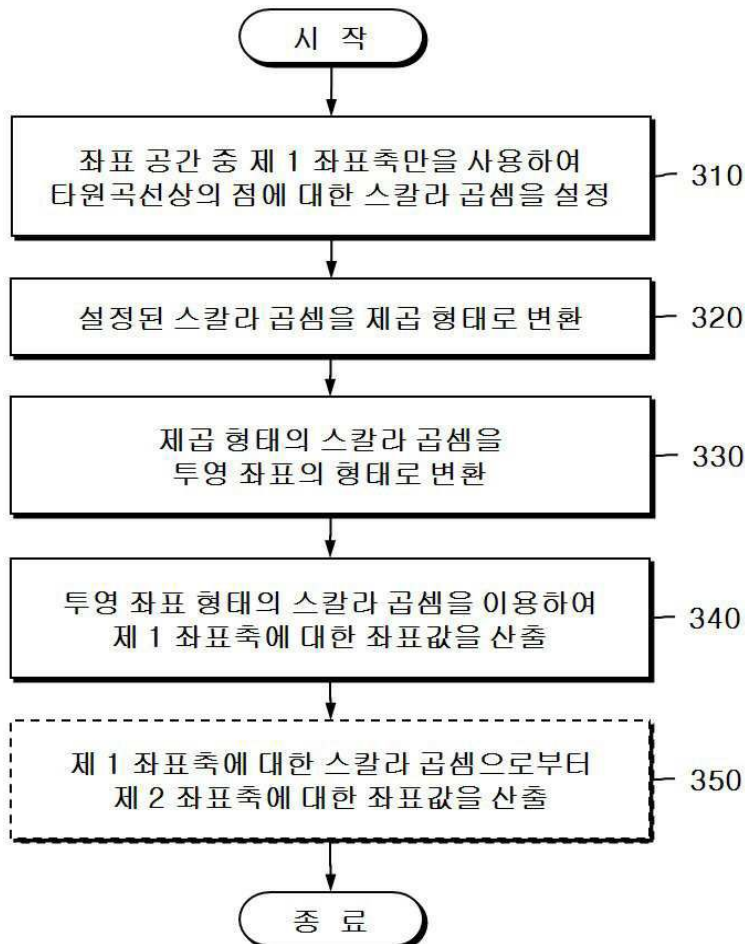
도면2

INPUT: 상수 $d = d_{n-1}2^{n-1} + \dots + d_0 (d_{n-1} = 1)$, 점 P
 OUTPUT: dP

```

1:  $Q[1] = 2P$ 
2: for  $i \leftarrow n-2$  downto 0
3:    $Q[2] = ECDBL^x(Q[d[i]])$ 
4:    $Q[1] = ECADD^x(Q[0], Q[1])$ 
5:    $Q[0] = Q[2-d[i]]$ 
6:    $Q[1] = Q[1+d[i]]$ 
7: Return ( $Q[0]$ )
    
```

도면3



도면4

1. $i=1$ 일 때의 루프 종료 후 $Q[0] = ((k-k_0)/2)P$, $Q[1] = ((k-k_0)/2)P + P$
2. $ECADD^{x^2}$ 를 이용하여 $Q[0] - P$ 계산 ($-P = (-x, y)$)
3. $ECADD^{x^2}$ 를 이용하여 $2Q[0] = (k-k_0)P$, $2Q[0] + P = (k-k_0)P + P$ 계산
4. If $k_0 = 0$, $2Q[0] = kP - k_0P$ 로 y 좌표 복원
5. If $k_0 = 1$, $2Q[0] = kP - k_0P + P$ 로 y 좌표 복원

도면5

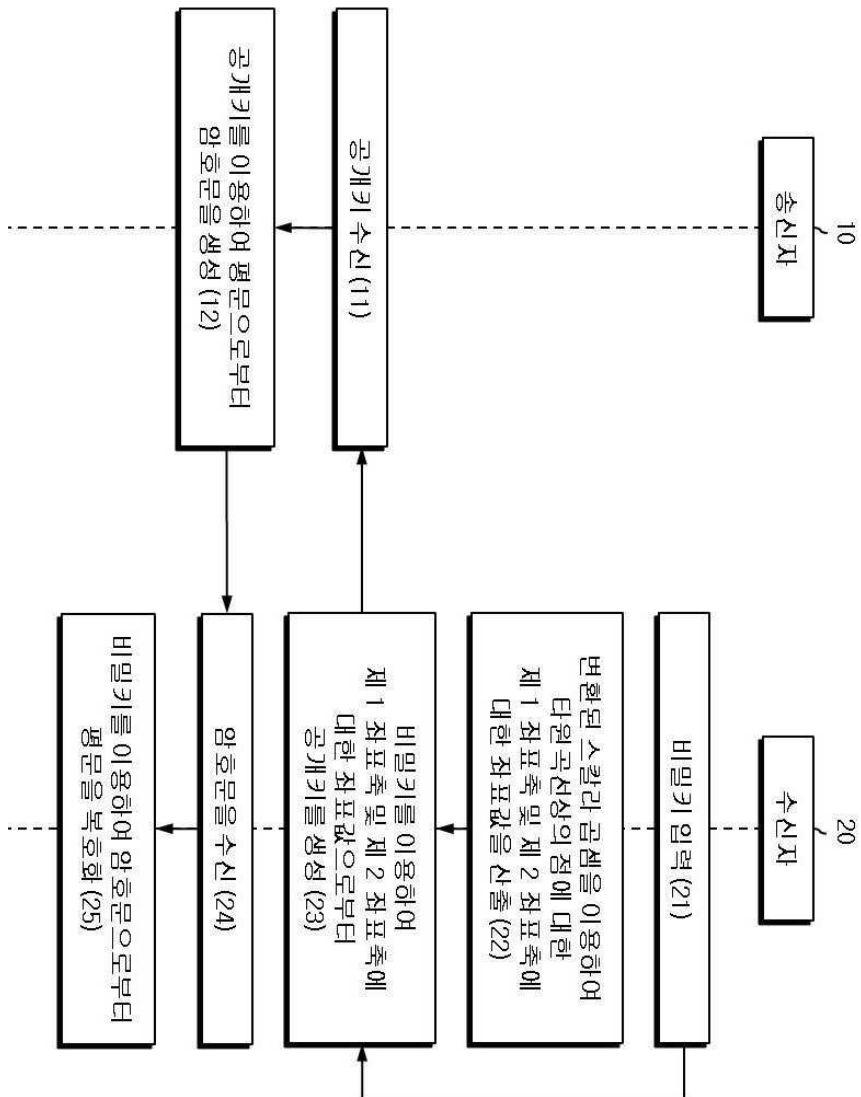
INPUT: 상수 $d = d_{n-1}2^{n-1} + \dots + d_0 (d_{n-1} = 1)$, 점 P
OUTPUT: dP

```

1:  $Q[0] = P, Q[1] = 2P$ 
2: for  $i \leftarrow n-2$  downto 1
3:    $Q[2] = ECDBL^{x^2}(Q[0])$ 
4:    $Q[1] = ECADD^{x^2}(Q[0], Q[1])$ 
5:    $Q[0] = Q[2-d[i]]$ 
6:    $Q[1] = Q[1+d[i]]$ 
7:  $Q[2] = ECADD^{x^2}(Q[0], -P)$ 
8:  $Q[2] = ECADD^x(Q[1], Q[2])$ 
9:  $Q[1] = ECADD^x(Q[0], Q[1])$ 
10:  $Q[0] = Q[2-k_0]$ 
11:  $ECRE^{(x,y)}(Q[0])$ 
12: Return ( $Q[0]$ )

```

도면6



도면7

