



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년02월15일
(11) 등록번호 10-1707334
(24) 등록일자 2017년02월09일

(51) 국제특허분류(Int. Cl.)
H04L 9/30 (2006.01) G06F 7/72 (2006.01)
(52) CPC특허분류
H04L 9/3066 (2013.01)
G06F 7/725 (2013.01)
(21) 출원번호 10-2015-0154992
(22) 출원일자 2015년11월05일
심사청구일자 2015년11월05일
(56) 선행기술조사문헌
KR1020090070060 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
이한호
인천광역시 연수구 센트럴로 194, 203동 2205호 (송도동, 더샵센트럴파크2)
오희판뚜이
인천광역시 남구 인하로 100 인하대학교
(74) 대리인
양성보

전체 청구항 수 : 총 7 항

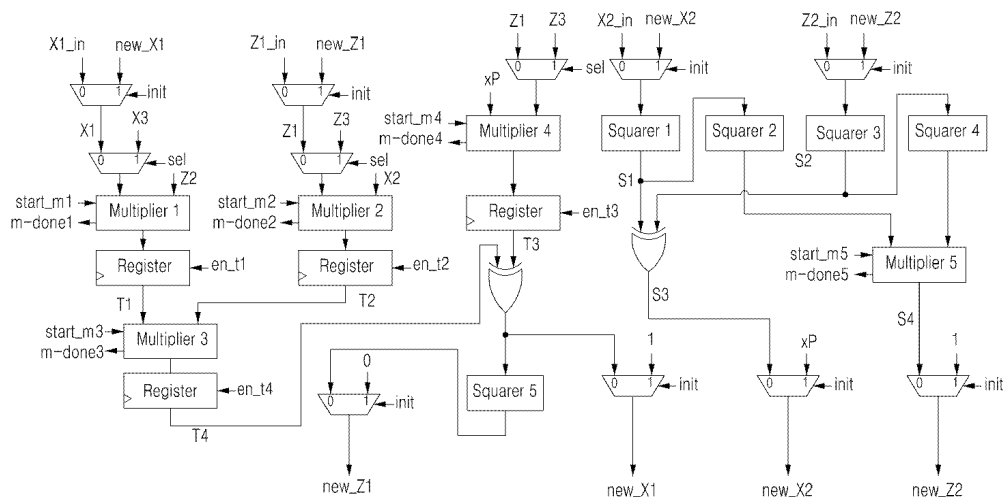
심사관 : 전용혜

(54) 발명의 명칭 효율적인 타원곡선 암호화 연산 장치 및 그 방법

(57) 요약

수정된 Lopez-Dahab 알고리즘, Left-To-Right 알고리즘을 기반으로 불필요한 포인트 덧셈 연산을 줄이는 것을 특징으로 하는 타원곡선 암호화 방법 및 장치가 제시된다. 일 실시예에 따른 타원곡선 암호화 방법은, 아핀 좌표 타원곡선 상의 점을 투영 좌표로 변환하는 단계; 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계; 상기 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정하는 단계; 상기 조정된 수행값을 상기 투영 좌표에서 상기 아핀 좌표로 변환함에 따라 결과값을 획득하는 단계; 및 상기 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출하는 단계를 포함할 수 있다.

대표도



이 발명을 지원한 국가연구개발사업

과제고유번호 1711014156

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 대학ICT 연구센터육성 지원사업

연구과제명 스마트한 주파수 이용을 위한 스펙트럼 및 전파기술

기 여 율 1/1

주관기관 인하대학교 산학협력단

연구기간 2014.06.01 ~ 2015.12.31

명세서

청구범위

청구항 1

타원곡선 암호화 방법에 있어서,

아핀 좌표(Affine Coordinate)에서 타원곡선 상의 점을 투영 좌표(Projective Coordinate)로 변환하는 단계;

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계;

상기 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정하는 단계;

상기 조정된 수행값을 상기 투영 좌표에서 상기 아핀 좌표로 변환함에 따라 결과값을 획득하는 단계; 및

상기 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출하는 단계

를 포함하고,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

Bit-Serial 구조의 곱셈 연산기 및 필드 연산을 사용하는 제어 신호를 사용하여 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하고, 상기 투영 좌표 상에서 포인트 두 배 연산을 수행하는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 2

제1항에 있어서,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

수정된 Lopez-Dahab 알고리즘, Left-To-Right 알고리즘이 사용되어 스칼라 포인트 덧셈 연산을 줄임으로써 상기 스칼라 포인트 곱셈 연산의 수행 시간을 감소시키는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 3

제2항에 있어서,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

키의 비트값에 기초하여 상기 투영 좌표 상에서 상기 스칼라 포인트 덧셈 연산과 상기 스칼라 포인트 곱셈 연산이 결정되는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 4

제3항에 있어서,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

상기 키의 비트값이 0일 경우, 4번의 필드 제곱 연산과 1번의 필드 덧셈 연산 및 1번의 필드 곱셈 연산을 수행하는 포인트 두 배 연산을 수행하고,

상기 키의 비트값이 1일 경우, 상기 포인트 두 배 연산을 수행한 뒤, 4번의 필드 곱셈 연산과 2번의 필드 덧셈 연산 및 1번의 필드 제곱 연산을 수행하는 포인트 덧셈 연산을 수행하는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

제1 곱셈기와 제2 곱셈기를 동시에 계산하고 제1 신호와 제2 신호를 통하여 제3 곱셈기 및 제4 곱셈기를 사용가능(Enable)상태로 전환하고, 상기 제3 곱셈기 및 상기 제4 곱셈기를 병렬로 계산하는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 7

제6항에 있어서,

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는,

상기 제3 곱셈기 및 상기 제4 곱셈기의 출력값을 XOR 게이트를 통하여 XOR 연산을 수행하고, 상기 XOR 연산을 수행한 출력값을 다중화기와 제곱연산기를 통하여 수행값을 획득하는 단계

를 포함하는 타원곡선 암호화 방법.

청구항 8

타원곡선 암호화 장치에 있어서,

아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환하는 제1 변환부;

상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 연산부;

상기 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정하는 조정부;

상기 조정된 수행값을 상기 투영 좌표에서 상기 아핀 좌표로 변환함에 따라 결과값을 획득하는 제2 변환부; 및

상기 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출하는 획득부

를 포함하고,

상기 연산부는,

Bit-Serial 구조의 곱셈 연산기 및 필드 연산을 사용하는 제어 신호를 사용하여 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하고, 상기 투영 좌표 상에서 포인트 두 배 연산을 수행하는

타원곡선 암호화 장치.

발명의 설명

기술 분야

[0001] 아래의 설명은 타원곡선을 이용한 암호를 위한 효율적인 연산 장치 및 그 방법에 관한 것이다. 더욱 상세하게는 타원곡선 암호화를 위해 타원곡선 상에서 표현되는 점들에 대한 기하학적인 연산인 스칼라 포인트 곱셈을 처리하는 연산 장치 및 그 방법에 관한 것이다.

배경 기술

[0002]인터넷 및 무선통신이 성장함에 따라 정보 보호를 위한 암호 시스템의 필요성 또한 증가하고 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 시스템들 중 타원곡선 암호화 시스템(Elliptic Curve Cryptosystem, ECC)이 최근 학계나 산업계로부터 많은 관심을 모으고 있다. ECC는 공개 키 암호 시스템으로서 데이터의 암호화에는 공개 키가 사용되고 복호화에는 비밀 키가 사용되는 암호시스템으로서, 타원곡선이라 불리는 수학적 객체 상에서 정의되는 특수한 가산법을 기반으로 하여 암호화/복호화를 구현하는 암호 시스템이다. ECC의 장점은 RSA(Rivest Shamir Adleman)와 같은 기존의 다른 암호 시스템들에 비해 현저히 작은 크기의 키를 사용하면서도 동일한 안전도를 가진다는 점이다.

[0004] 유한체(Galois Field, GF)상의 타원곡선은 수학적식1을 만족하는 해들의 집합이다.

[0005] 수학적식 1:

$$y^2 + xy = x^3 + ax^2 + b$$

[0006]

[0007] 타원곡선 상에서의 연산 결과 Q는 타원곡선 상의 점 P와 스칼라 값 k의 스칼라 포인트 곱셈에 의해 수행된다. 이진 유한체 GF(2^m)상에서 m-bit의 크기를 갖는 스칼라 값 k는 수학적식2와 같이 표현가능 하다.

[0008] 수학적식 2:

$$k = k_0 + k_1 2 + \dots + k_{m-1} 2^{m-1}$$

[0009]

[0010] 스칼라 포인트 곱셈의 결과인 Q는 수학적식3과 같이 표현할 수 있다.

[0011] 수학적식 3:

$$Q = kP = P + \dots + P + P$$

[0012]

[0013] 포인트 곱셈 연산을 계산하기 위한 세 가지 주요 단계는 다음과 같다.

[0014] (1) 점 P를 아핀 좌표(Affine Coordinate)에서 투영 좌표(Projective Coordinate)로 변환한다.

[0015] (2) 투영 좌표 상에서 Q=kP를 계산한다.

[0016] (3) 점 Q를 투영 좌표에서 아핀 좌표로 변환한다.

[0017] 유한체를 이용한 연산에는 곱셈, 제곱연산, 덧셈, 뺄셈, 반전이 있다. 덧셈, 뺄셈 연산은 XOR(exclusive OR)연산을 통해 계산할 수 있다. 이진 유한체 GF(2^m)의 원소인 다항식을 a(x), b(x)라고 하여 수학적식 4 와 같이 나타낼 수 있다.

[0018] 수학적식 4:

$$a(x) = a_{m-1}x^{m-1} + \dots + a_1x + a_0$$

$$b(x) = b_{m-1}x^{m-1} + \dots + b_1x + b_0$$

[0019]

[0020] 이진 유한체 GF(2^m)의 원소인 다항식을 수학적식 4 와 같이 a(x), b(x)라고 할 때, 유한체의 곱셈 연산은 수학적식 5 과 같이 나타낼 수 있다. 이때, 사용된 연산은 부분합 과 기약다항식 f(x)의 modulo연산이다.

[0021] 수학적식 5:

$$c(x) = a(x)b(x) \bmod f(x),$$

$$c(x) = \left(\sum_{i=0}^{m-1} b_i a(x)x^i \right) \bmod f(x).$$

[0022]

[0023] 흔히 사용되는 유한체 곱셈 연산으로는 Digit-Serial 곱셈연산과 Bit-Serial 곱셈연산이 있다. Digit-Serial 곱셈연산은 시스템의 지연을 줄이기 위해 사용되지만 하드웨어의 복잡도가 증가하는 반면, Bit-Serial 곱셈연산은 하드웨어의 복잡도는 줄어들지만 시스템의 지연이 증가하는 특징을 가지고 있다.

[0024] 제곱연산은 같은 입력을 이용하여 곱셈연산을 수행한 것과 같다. Lopez-Dahab 알고리즘은 ECC에 있어서 타원곡선 상에 존재하는 점들의 스칼라 포인트 곱셈 연산을 수행할 때 사용하는 알고리즘이다.

[0025] Montgomery Ladder Method는 Lopez-Dahab 알고리즘을 이용해 타원곡선 상의 스칼라 포인트 곱셈 연산을 할 때 가장 흔히 사용하는 알고리즘으로 아핀 좌표뿐만 아니라 투영 좌표에서도 연산이 가능하다.

[0026] 하기의 표 1은 Montgomery Ladder Method 곱셈 연산 과정의 의사코드를 도시한다.

[0027] 표 1:

Algorithm 1. LD scalar point multiplication
Input: $k = (k_{m-1} \dots k_1 k_0), P(x_P, y_P)$
Output: $Q = kP$
1. $A = 0; B = P;$
2. for $i = 0$ to $i = m-1$ do
if $(k_i = 0)$
$B = A + B;$
$A = 2A;$
else
$A = A + B;$
$B = 2B;$
end if
end for
3. Return $Q = A;$

[0028]

[0029] 이러한 연산을 수행하기 위해서는 포인트 덧셈 연산과 포인트 두 배 연산이 효율적으로 결합되어야 한다.

[0030] 점 $A(x_A, y_A)$, $B(x_B, y_B)$ 가 타원곡선 위의 두 점이라고 할 때, 점 A와 B의 x좌표의 합인 x_{A+B} 와 x_{A+A} 는 수학적식6과 같이 표현 가능하다.

[0031] 수학식 6:

$$x_{A+B} = x_P + x_B(x_A + x_B)^{-1} + (x_B(x_A + x_B)^{-1})^2,$$

$$x_{A+A} = x_A^2 + b/x_A^2,$$

[0032]

[0033] 투영 좌표를 사용하여 x_A, x_B 를 $x_A = X_A/Z_A$, $x_B = X_B/Z_B$ 로 다시 표현하면 수학식6은 수학식7로 표현할 수 있다.

[0034] 수학식 7:

$$Z_{A+B} = (X_A Z_B + X_B Z_A)^2,$$

$$X_{A+B} = x_P Z_{A+B} + X_A X_B Z_A Z_B,$$

$$Z_{A+A} = X_A^2 Z_A^2,$$

$$X_{A+A} = X_A^4 + b Z_A^4,$$

[0035]

[0036] 수학식7과 같이 새롭게 표현된 좌표와 Left-To-Right 알고리즘을 이용하여 스칼라 포인트 곱셈 연산을 수행한다.

[0037] 하기의 표 2는 Left-To-Right 곱셈 연산 과정의 의사코드를 도시한다.

[0038] 표 2:

Algorithm 2. Left-to-right Algorithm

Input: $k = (k_{m-1} \dots k_1 k_0)$, $P(x_P, y_P)$

Output: $Q = kP$

1. $Q = \infty$

2. *for* $i = m - 1$ *to* $i = 0$ *do*

2.1 $Q = 2P$

2.2 *if* $(k_i = 1)$

$Q = Q + P$

3. **Return** (Q)

[0039]

발명의 내용

해결하려는 과제

- [0040] 실시예들은 타원곡선 암호화 장치 설계에 있어 종래의 Lopez-Dahab 알고리즘과 Left-To-Right 알고리즘을 개선하여 새로운 효율적인 Lopez-Dahab, Left-To-Right 알고리즘을 제안하고, 제안한 알고리즘을 이용하여 불필요한 반복을 줄여 낮은 하드웨어 복잡도를 가지는 효율적인 타원곡선 암호화 장치 및 그 방법을 제공한다.
- [0041] 실시예들은 Bit-Serial 구조의 스칼라 포인트 곱셈기를 사용하고 필드 연산을 제어하는 제어신호를 사용하여 적은 수준의 하드웨어 복잡도를 갖는 타원곡선 암호화 장치를 제공한다.

과제의 해결 수단

- [0042] 일 실시예에 따르면, 타원곡선 암호화 방법은, 아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환하는 단계; 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계; 상기 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정하는 단계; 상기 조정된 수행값을 상기 투영 좌표에서 상기 아핀 좌표로 변환함에 따라 결과값을 획득하는 단계; 및 상기 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출하는 단계를 포함할 수 있다.
- [0043] 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 수정된 Lopez-Dahab 알고리즘, Left-To-Right 알고리즘이 사용되어 스칼라 포인트 덧셈 연산을 줄임으로써 상기 스칼라 포인트 곱셈 연산의 수행 시간을 감소시키는 단계를 포함할 수 있다.
- [0044] 또 다른 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 키의 비트값에 기초하여 상기 투영 좌표 상에서 상기 스칼라 포인트 덧셈 연산과 상기 스칼라 포인트 곱셈 연산이 결정되는 단계를 포함할 수 있다.
- [0045] 또 다른 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 상기 키의 비트값이 0일 경우, 4번의 필드 제곱 연산과 1번의 필드 덧셈 연산 및 1번의 필드 곱셈 연산을 수행하는 포인트 두 배 연산을 수행하고, 상기 키의 비트값이 1일 경우, 상기 포인트 두 배 연산을 수행한 뒤, 4번의 필드 곱셈 연산과 2번의 필드 덧셈 연산 및 1번의 필드 제곱 연산을 수행하는 포인트 덧셈 연산을 수행하는 단계를 포함할 수 있다.
- [0046] 또 다른 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 상기 필드 곱셈 연산을 계산하기 위하여 Bit-Serial 구조를 갖는 필드 곱셈 연산기를 사용하는 단계를 포함할 수 있다.
- [0047] 또 다른 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 제1 곱셈기와 제2 곱셈기를 동시에 계산하고 제1 신호와 제2 신호를 통하여 제3 곱셈기 및 제4 곱셈기를 사용가능(Enable)상태로 전환하고, 상기 제3 곱셈기 및 상기 제4 곱셈기를 병렬로 계산하는 단계를 포함할 수 있다.
- [0048] 또 다른 일측에 따르면, 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 단계는, 상기 제3 곱셈기 및 상기 제4 곱셈기의 출력값을 XOR 게이트를 통하여 XOR 연산을 수행하고, 상기 XOR 연산을 수행한 출력값을 다중화기와 제곱연산기를 통하여 수행값을 획득하는 단계를 포함할 수 있다.
- [0049] 일 실시예에 따르면, 타원곡선 암호화 장치는, 아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환하는 제1 변환부; 상기 투영 좌표 상에서 상기 점에 대한 스칼라 포인트 곱셈 연산을 수행하는 연산부; 상기 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정하는 조정부; 상기 조정된 수행값을 상기 투영 좌표에서 상기 아핀 좌표로 변환함에 따라 결과값을 획득하는 제2 변환부; 및 상기 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출하는 획득부를 포함할 수 있다.

발명의 효과

- [0050] 실시예들에 따르면, 타원곡선 암호화 장치 설계에 있어서 제안하는 알고리즘을 사용함으로써, 불필요한 스칼라

포인트 덧셈 연산을 줄일 수 있는 효율적인 타원곡선 암호화 장치 및 그 방법을 제공할 수 있다.

[0051] 실시예들에 따르면, 타원곡선 암호화 장치 설계에 있어서 스칼라 포인트 곱셈 연산부에서 Bit-Serial 구조와 필드 연산 제어신호를 사용함으로써, 낮은 하드웨어 복잡도를 가지며 시스템의 지연을 줄일 수 있는 효율적인 타원곡선 암호화 장치 및 그 방법을 제공할 수 있다.

도면의 간단한 설명

[0052] 도 1은 일 실시예에 따른 타원곡선 암호화 장치의 구성을 나타내는 도면이다.

도 2는 일 실시예에 따른 스칼라 포인트 곱셈 연산부를 나타내는 도면이다.

도 3은 일 실시예에 따른 필드 곱셈 연산부를 나타내는 도면이다.

도 4는 일 실시예에 따른 스칼라 포인트 곱셈 연산 방법의 흐름도를 나타내는 도면이다.

도 5는 일 실시예에 따른 타원곡선 암호화 장치의 구성을 나타낸 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0053] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0054] 일 실시예에 따른 타원곡선 암호화 장치는 다양한 암호 시스템을 위한 타원곡선 암호화 장치에 동일하게 적용될 수 있다. 아래에서는 GF()의 크기를 갖는 이진 유한체를 기준으로 설명하기로 한다.

[0055] 일 실시예에 따른 타원곡선 암호화 장치는 타원곡선 암호화 장치를 위한 알고리즘으로서 키의 길이가 163비트인 경우 최적화 된 낮은 하드웨어 복잡도를 가지는 수정된 Lopez-Dahab 알고리즘, Left-To-Right 알고리즘이 사용될 수 있다.

[0056] 도 1은 일 실시예에 따른 타원곡선 암호화 장치의 구성을 나타내는 도면이다.

[0057] 타원곡선 암호화 장치는 아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환할 수 있다. 타원곡선 암호화 장치는 투영 좌표 상에서 점에 대한 스칼라 포인트 곱셈 연산을 수행할 수 있다. 타원곡선 암호화 장치는 스칼라 포인트 곱셈 연산을 수행한 수행값을 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정할 수 있다. 타원곡선 암호화 장치는 조정된 수행값을 투영 좌표에서 아핀 좌표로 변환함에 따라 결과값을 획득할 수 있다. 타원곡선 암호화 장치는 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출할 수 있다.

[0058] 본 발명에서는 제안된 알고리즘을 사용하는 타원곡선 암호화 장치는 불필요한 스칼라 포인트 곱셈 연산을 줄임으로써, 효율적으로 연산을 수행한다.

[0059] 하기의 표 3은 제안하는 알고리즘의 곱셈 연산 과정의 의사코드를 도시한다.

표 3:

<i>Algorithm 3. Modified LD and Left-to-Right algorithm</i>	
Input: $k = (k_{m-1} \dots k_1 k_0), P(x_P, y_P)$	
Output: $Q = kP$	
1. From affine coordinate to projective coordinate:	
$X_2 = x_P, Z_2 = 1, X_3 = x_P^2 + b, Z_3 = x_P^2$	
if $(k_0 = 1) X_1 = x_P, Z_1 = 1$	
else $X_1 = 0, Z_1 = 0$	
end if	
2. Calculate in projective coordinate:	
2.1 for $i = 1$ to $i = m - 1$ do	
$S_1 = X_2^2, S_2 = Z_2^2, S_3 = S_1^2, S_4 = S_2^2, X_2 = S_3 + S_4, Z_2 = M_1(S_1, S_2)$	
2.2 if $(k_i = 1)$	
$T_1 = M_2(X_1, Z_2), T_2 = M_3(X_2, Z_1), Z_1 = (T_1 + T_2)^2, T_3 = M_4(x_P, Z_1), T_4 = M_5(T_1, T_2), X_1 = T_3 + T_4$	
end if	
end for	
3. Adjust the result:	
$T_1 = M_2(X_3, Z_2), T_2 = M_3(X_2, Z_3), Z_1 = (T_1 + T_2)^2, T_3 = M_4(x_P, Z_3), T_4 = M_5(T_1, T_2), X_1 = T_3 + T_4$	
4. Convert from projective to affine coordinate:	
if $(Z_1 = 0) X_1 = x_P, Z_1 = x_P + y_P$	
else $X_1 = X_1/Z_1, X_2 = X_2/Z_1$	
$T_2 = M_2(X_1 + x_P X_2 + x_P), T_3 = M_3(x_P + x_P, x_P^{-1}), T_4 = T_2 + x_P^2 + y_P, T_2 = M_2(T_3, T_4), Z_1 = T_2 + y_P$	
end if	
5. Return output value:	
$x_Q = X_1, y_Q = Z_1$	

제안하는 알고리즘은 네 가지 단계를 거쳐 포인트 곱셈 연산을 수행한다.

(1) 타원곡선 상의 점 P를 아핀 좌표(Affine Coordinate)에서 투영 좌표(Projective Coordinate)로 변환한다.

(2) 투영 좌표 상에서 포인트 곱셈 연산을 수행한다.

(3) Lopez-Dahab 알고리즘과 비교하여 정확한 값을 얻는다.

(4) 투영 좌표에서 아핀 좌표로 변환하고 결과값 Q를 얻는다.

제안하는 수정된 Lopez-Dahab 알고리즘과 Left-To-Right 알고리즘은 스칼라 포인트 곱셈 연산을 수행함에 있어서 많은 수의 스칼라 포인트 덧셈 연산을 줄이는 장점을 갖는다.

제안하는 알고리즘의 첫 번째 단계에서 타원곡선 상의 점 P를 아핀 좌표에서 투영 좌표로 변환하고 $X_2 = x_P, Z_2 = 1$ 로 초기 값을 설정할 수 있다. X_1 과 Z_1 값은 키 값의 LSB인 k_0 에 의해 결정될 수 있다.

$k_0=1$ 이면 $X_1 = x_p$, $Z_1 = 1$ 이고 $k_0 = 0$ 이면 X_1 과 Z_1 은 0이다.

[0069] 투영 좌표 상에서 포인트 곱셈 연산을 수행하는 단계에서는, 이후 키의 비트 값인 k_i 에 따라 투영 좌표 상에서 스칼라 포인트 덧셈 연산과 스칼라 포인트 곱셈 연산이 결정될 수 있다. 개선된 알고리즘에 의한 이 단계의 장점은 키의 비트 값인 k_i 가 0일 경우 불필요한 스칼라 포인트 덧셈 연산을 줄일 수 있다. 그러므로 스칼라 포인트 곱셈 연산을 수행하는데 소요되는 시간이 확연히 줄어들게 된다.

[0070] 표3에 나온 것과 같이 타원곡선 암호화 장치는 키의 비트 값이 0일 경우, 네 번의 필드 제곱 연산($S_1=X_2^2$, $S^2=Z_2^2$, $S_3=S_1^2$, $S_4=S_2^2$)과 한 번의 필드 덧셈 연산($X_2=S_3+S_4$), 한 번의 필드 곱셈 연산($Z_2=S_1S_2$)을 수행하는 포인트 두 배 연산만을 수행할 수 있다. 타원곡선 암호화 장치는 키의 비트 값이 1일 경우 포인트 두 배 연산을 수행한 뒤, 네 번의 필드 곱셈 연산($T_1=X_1Z_2$, $T_2=X_2Z_1$, $T_3=x_pZ_1$, $T_4=T_1T_2$)과 두 번의 필드 덧셈 연산(T_1+T_2 , T_3+T_4), 한 번의 필드 제곱 연산($((T_1+T_2)^2)$)을 수행하는 포인트 덧셈 연산을 추가로 수행할 수 있다.

[0071] 타원곡선 암호화 장치는 모든 연산의 반복이 종료되면 점과 점에 대한 스칼라 포인트 곱셈 연산을 수행한 수행 값 X_1 값과 기존의 Lopez-Dahab 알고리즘의 X_1 값 사이의 차이가 발생함을 판단할 수 있다. 타원곡선 암호화 장치는 기존의 Lopez-Dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통해 얻은 값과 수정된 Lopez-Dahab 알고리즘의 수행값을 비교함으로써 수행값 X_1 값을 정확하게 수정해주는 과정을 수행할 수 있다.

[0072] 일 실시예에 따른 타원곡선 암호화 장치에 있어서, 타원곡선 암호화 장치의 구조는 알고리즘의 반복과정에서 다섯 개의 필드 곱셈기를 사용할 수 있다.

[0073] 타원곡선 암호화 장치는 수정된 투영 좌표 상에서 점에 대한 스칼라 포인트 곱셈 연산을 수행하고, 투영 좌표 상에서 포인트 두 배 연산을 수행할 수 있다. 타원곡선 암호화 장치는 제안된 구조에서 두 개의 제곱 연산($S_1 = X_2^2$, $S_2 = Z_2^2$)은 제곱 연산 장치1(Squarer1)과 제곱 연산 장치3(Squarer3)에 의해 병렬로 수행되며 S_3 과 S_4 의 계산은 제곱 연산 장치2(Squarer2)와 제곱 연산 장치4(Squarer4)에 의해 동시에 수행될 수 있다. 제곱 연산 장치2(Squarer2)의 결과 S_2 와 제곱 연산 장치4(Squarer4)의 결과 S_4 는 곱셈기5에서 곱셈 연산($Z_2 = S_2 \cdot S_4$)을 통해 계산될 수 있다. 제곱 연산 장치1(Squarer2)과 제곱 연산 장치3(Squarer3)의 결과는 163비트 XOR게이트에서 XOR연산($X_2 = S_1 \oplus S_3$)에 의해 계산될 수 있다.

[0074] 도 3을 참고하면, 필드 곱셈 연산부를 나타낸 것이다.

[0075] 타원곡선 암호화 장치의 스칼라 포인트 곱셈 장치를 설계함에 있어서 가장 많이 사용되는 모듈은 필드 곱셈기이다. 필드 곱셈 연산을 계산하기 위해 Bit-Serial 방법을 사용할 경우 Digit-Serial 방법을 사용한 경우보다 클럭 사이클 수는 많지만 낮은 하드웨어 복잡도를 갖는다. 일 실시예에 따른 타원곡선 암호화 장치는 Bit-Serial 방법을 사용한 곱셈기와 필드 연산 제어신호를 이용하여 낮은 하드웨어의 복잡도를 가지면서 시스템의 지연의 적은 타원곡선 암호화 장치 및 그 방법을 제공한다.

[0076] 도 2는 일 실시예에 따른 스칼라 포인트 곱셈 연산부를 나타내는 도면이다.

[0077] 도2에서 도시한 것처럼 곱셈기1은 $T_1(T_1 = X_1 \cdot Z_2)$, 곱셈기2는 $T_2(T_2 = X_2 \cdot Z_1)$ 를 동시에 계산하

고 m_done1신호와 m_done2신호를 통해 곱셈기3과 곱셈기4를 사용 가능(enable)상태로 바꾸며 곱셈기3과 곱셈기4는 $T_3(T_3 = x_p \cdot Z_1)$ 와 $T_4(T_4 = T_1 \cdot T_2)$ 를 병렬로 계산할 수 있다.

[0078] 타원곡선 암호화 장치는 제안한 알고리즘의 수행시간을 두 배로 줄일 수 있는 방법을 제공할 수 있다. 곱셈기3과 곱셈기4의 출력값은 163비트 XOR 게이트를 통해 XOR 연산 $(T_3 \oplus T_4)$ 을 하며 결과값은 다중화기(Multiplexer)와 제곱연산장치5(Squarer5)를 통해 새로운 X_1 , Z_1 값을 출력할 수 있다.

[0079] 일 실시예에 따른 타원곡선 암호화 장치는 수정된 Lopez-Dahab 알고리즘, Left-To-Right 알고리즘을 사용함으로써 불필요한 계산과정을 줄이고 반복해서 사용하는 모듈을 재사용함으로써 낮은 하드웨어 복잡도를 가진다는 점이다.

[0080] 타원곡선 암호화 장치는 투영 좌표를 아핀 좌표로 변환함에 따라 결과값을 반환할 수 있다.

[0081] 도 4는 일 실시예에 따른 스칼라 포인트 곱셈 연산 방법의 흐름도를 나타내는 도면이다.

[0082] 타원곡선 암호화 장치는 아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환할 수 있다(410). 타원곡선 암호화 장치는 타원곡선 상의 점 P를 아핀 좌표에서 투영 좌표로 변환하고, $X_2 = x_p$, $Z_2 = 1$ 로 초기 값을 설정할 수 있다. X_1 과 Z_1 값은 키 값의 LSB인 k_0 에 의해 결정될 수 있다. $k_0=1$ 이면 $X_1 = x_p$, $Z_1 = 1$ 이고 $k_0 = 0$ 이면 X_1 과 Z_1 은 0이다.

[0083] 타원곡선 암호화 장치는 키의 비트 값인 k_i 에 따라 투영 좌표 상에서 스칼라 포인트 덧셈 연산과 스칼라 포인트 곱셈 연산이 결정될 수 있다. 타원곡선 암호화 장치는 키의 비트 값이 0일 경우, 네 번의 필드 제곱 연산($S_1=X_2^2$, $S^2=Z_2^2$, $S_3=S_1^2$, $S_4=S_2^2$)과 한 번의 필드 덧셈 연산($X_2=S_3+S_4$), 한 번의 필드 곱셈 연산($Z_2=S_1S_2$)을 수행하는 포인트 두 배 연산만을 수행할 수 있다. 타원곡선 암호화 장치는 키의 비트 값이 1일 경우, 알고리즘은 포인트 두 배 연산을 수행한 뒤, 네 번의 필드 곱셈 연산($T_1=X_1Z_2$, $T_2=X_2Z_1$, $T_3=x_pZ_1$, $T_4=T_1T_2$)과 두 번의 필드 덧셈 연산(T_1+T_2 , T_3+T_4), 한 번의 필드 제곱 연산($((T_1+T_2)^2)$)을 수행하는 포인트 덧셈 연산을 추가로 수행할 수 있다.

[0084] 타원곡선 암호화 장치는 투영 좌표 상에서 점과 스칼라 포인트 곱셈 연산을 수행함에 따라 획득된 수행값 X_1 과 종래의 Lopez-Dahab 알고리즘의 X_1 값 사이의 차이가 발생한다. 이에 따라 타원곡선 암호화 장치는 종래의 Lopez-Dahab 알고리즘에서 스칼라 포인트 곱셈 연산을 통해 얻은 값과 수정된 Lopez-Dahab 알고리즘에서 스칼라 포인트 곱셈 연산을 통하여 획득된 수행값을 비교하여 X_1 값을 정확하게 수정해주는 과정을 수행할 수 있다.

[0085] 예를 들면, 타원곡선 암호화 장치는 다섯 개의 필드 곱셈기를 사용할 수 있다. 타원곡선 암호화 장치는 투영 좌표 상에서 점에 대한 스칼라 포인트 곱셈 연산을 수행하고, 투영 좌표 상에서 포인트 두 배 연산을 수행할 수 있다. 타원곡선 암호화 장치는 제안된 구조에서 두 개의 제곱 연산($S_1 = X_2^2$, $S_2 = Z_2^2$)은 제곱 연산 장치1(Squarer1)과 제곱 연산 장치3(Squarer3)에 의해 병렬로 수행되며 S_3 과 S_4 의 계산은 제곱 연산 장치2(Squarer2)와 제곱 연산 장치4(Squarer4)에 의해 동시에 수행될 수 있다. 제곱 연산 장치2(Squarer2)의 결과 S_2 와 제곱 연산 장치4(Squarer4)의 결과 S_4 는 곱셈기5에서 곱셈 연산($Z_2 = S_2 \cdot S_4$)을 통해 계산될 수

있다. 제곱 연산 장치1(Squarer2)과 제곱 연산 장치3(Squarer3)의 결과는 163비트 XOR게이트에서 XOR연산 ($X_2 = S_1 \oplus S_3$)에 의해 계산될 수 있다.

[0086] 타원곡선 암호화 장치는 스칼라 포인트 곱셈 장치를 필드 곱셈기로 사용할 수 있다. 타원곡선 암호화 장치는 Bit-Serial 방법을 사용한 곱셈기와 필드 연산 제어신호를 이용할 수 있다.

[0087] 곱셈기1은 $T_1(T_1 = X_1 \cdot Z_2)$, 곱셈기2는 $T_2(T_2 = X_2 \cdot Z_1)$ 를 동시에 계산하고 m_done1신호와 m_done2신호를 통해 곱셈기3과 곱셈기4를 사용 가능(enable)상태로 바꾸며 곱셈기3과 곱셈기4는 $T_3(T_3 = x_p \cdot Z_1)$ 와 $T_4(T_4 = T_1 \cdot T_2)$ 를 병렬로 계산할 수 있다.

[0088] 타원곡선 암호화 장치는 곱셈기3과 곱셈기4의 출력값을 163비트 XOR 게이트를 통해 XOR 연산 ($T_3 \oplus T_4$)을 수 행할 수 있으며, 결과값은 다중화기(Multiplexer)와 제곱연산장치5(Squarer5)를 통해 새로운 X_1 , Z_1 값을 출력할 수 있다.

[0089] 타원곡선 암호화 장치는 조정된 수행값을 투영 좌표에서 아핀 좌표로 변환함에 따라 결과값을 획득할 수 있다 (420). 타원곡선 암호화 장치는 결과값 Q를 획득할 수 있다. 이를 통하여 타원곡선 암호화 장치는 아핀 좌표 로 변환됨에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출할 수 있다.

[0090] 도 5는 일 실시예에 따른 타원곡선 암호화 장치의 구성을 나타낸 블록도이다.

[0091] 타원곡선 암호화 장치는 제1 변환부(510), 연산부(520), 조정부(530), 제2 변환부(540) 및 획득부(550)를 포함 할 수 있다.

[0092] 제1 변환부(510)는 아핀 좌표에서 타원곡선 상의 점을 투영 좌표로 변환할 수 있다.

[0093] 연산부(520)는 투영 좌표 상에서 점에 대한 스칼라 포인트 곱셈 연산을 수행할 수 있다. 연산부(520)는 수정된 Lopez-Dahab알고리즘, Left-To-Right알고리즘이 사용되어 스칼라 포인트 덧셈 연산을 줄임으로써 스칼라 포인트 곱셈 연산의 수행 시간을 감소시킬 수 있다.

[0094] 연산부(520)는 키의 비트값에 기초하여 상기 투영 좌표 상에서 스칼라 포인트 덧셈 연산과 스칼라 포인트 곱셈 연산이 결정될 수 있다. 연산부(520)는 키의 비트값이 0일 경우, 4번의 필드 제곱 연산과 1번의 필드 덧셈 연 산 및 1번의 필드 곱셈 연산을 수행하는 포인트 두 배 연산을 수행하고, 키의 비트값이 1일 경우, 포인트 두 배 연산을 수행한 뒤, 4번의 필드 곱셈 연산과 2번의 필드 덧셈 연산 및 1번의 필드 제곱 연산을 수행하는 포인트 덧셈 연산을 수행할 수 있다.

[0095] 연산부(520)는 필드 곱셈 연산을 계산하기 위하여 Bit-Serial 구조를 갖는 필드 곱셈 연산기를 사용할 수 있다.

[0096] 연산부(520)는 제1 곱셈기와 제2 곱셈기를 동시에 계산하고 제1 신호와 제2 신호를 통하여 제3 곱셈기 및 제4 곱셈기를 사용가능(Enable)상태로 전환하고, 제3 곱셈기 및 제4 곱셈기를 병렬로 계산할 수 있다. 연산부(52 0)는 제3 곱셈기 및 제4 곱셈기의 출력값을 XOR 게이트를 통하여 XOR 연산을 수행하고, XOR 연산을 수행한 출력 값을 다중화기와 제곱연산기를 통하여 수행값을 획득할 수 있다.

[0097] 조정부(530)는 스칼라 포인트 곱셈 연산을 수행한 Lopez-dahab 알고리즘의 스칼라 포인트 곱셈 연산을 통하여 획득한 값과 비교하여 조정할 수 있다.

[0098] 제2 변환부(540)는 조정된 수행값을 투영 좌표에서 아핀 좌표로 변환함에 따라 결과값을 획득할 수 있다.

[0099] 획득부(550)는 아핀 좌표로 변환함에 따라 획득된 결과값을 기반으로 암호화된 암호값을 산출할 수 있다.

[0100] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로 세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴 퓨터, FPGA(field programmable gate 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(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 기술분야에서 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.

[0101] 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, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)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.

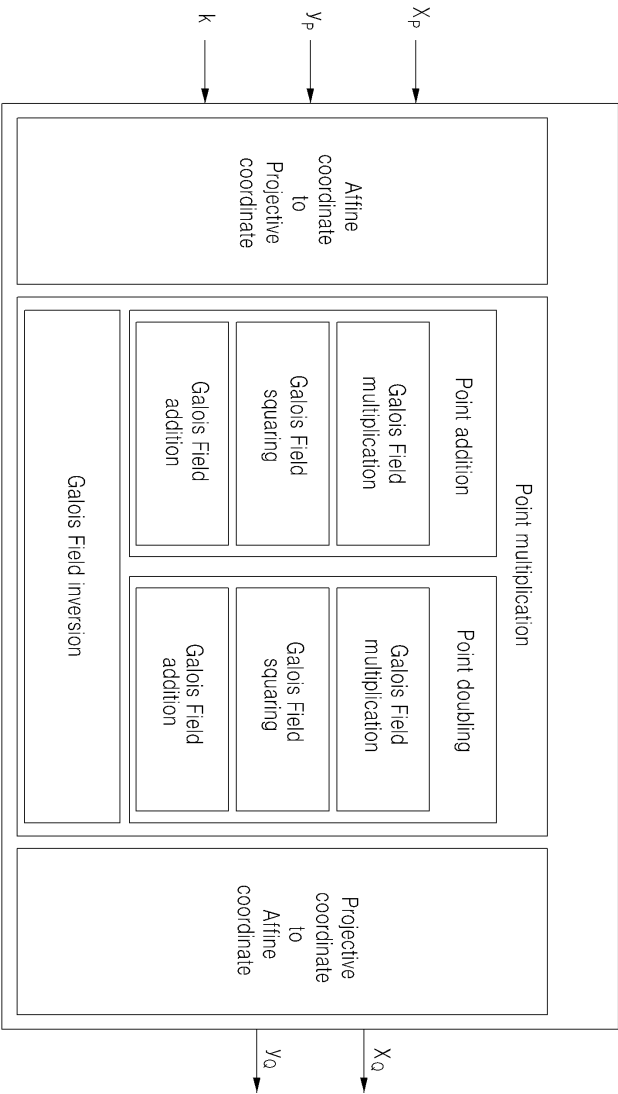
[0102] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 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.

[0103] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

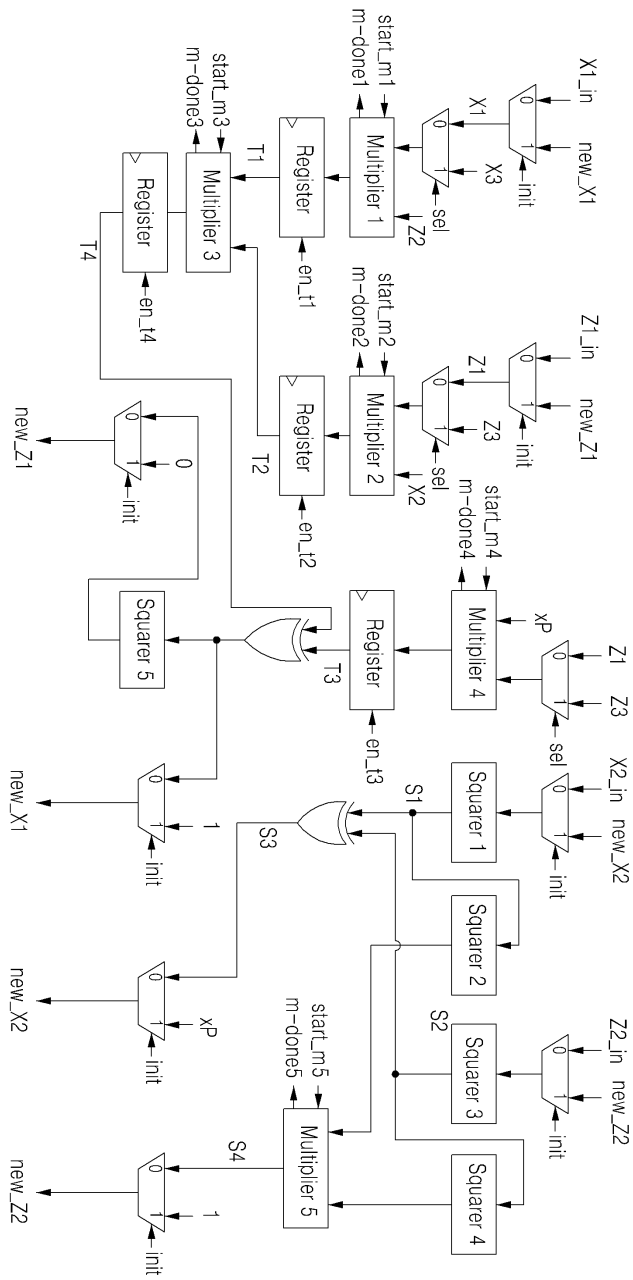
[0104] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

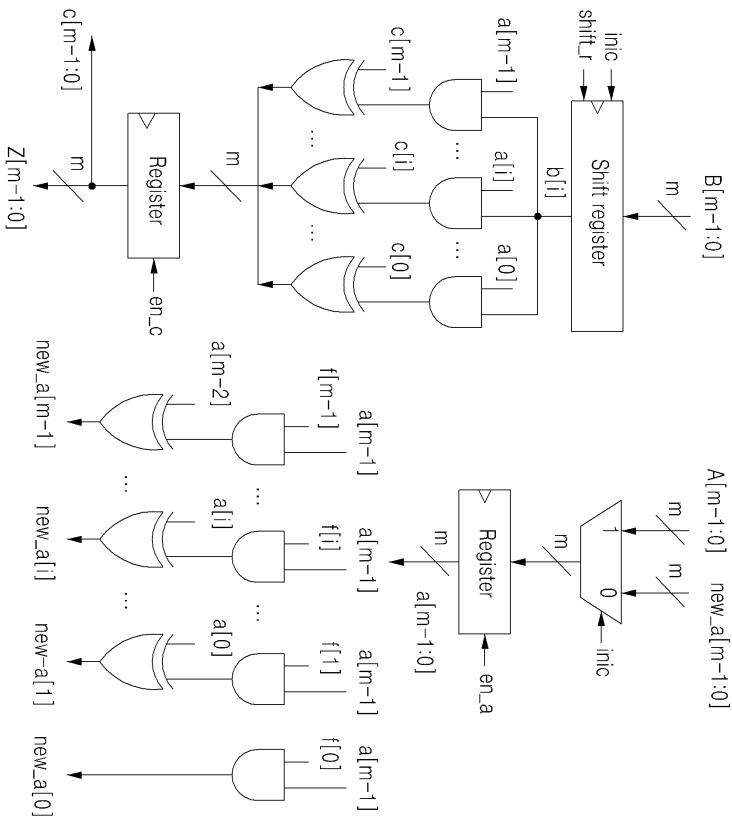
도면1



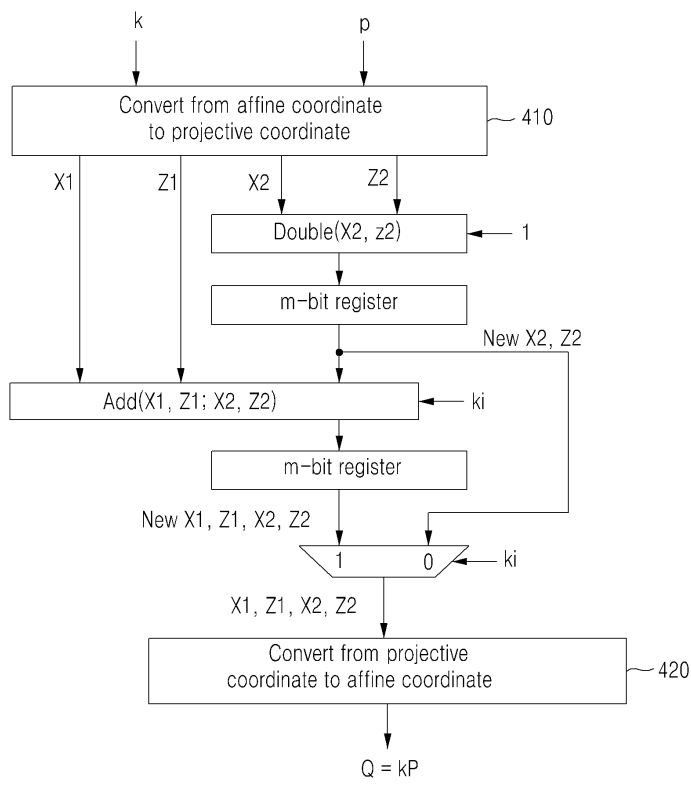
도면2



도면3



도면4



도면5

