



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0061411  
(43) 공개일자 2017년06월05일

(51) 국제특허분류(Int. Cl.)  
G06F 7/72 (2006.01) H04L 9/30 (2006.01)  
H04L 9/32 (2006.01)  
(52) CPC특허분류  
G06F 7/725 (2013.01)  
G06F 7/726 (2013.01)  
(21) 출원번호 10-2015-0166400  
(22) 출원일자 2015년11월26일  
심사청구일자 2015년11월26일

(71) 출원인  
한밭대학교 산학협력단  
대전광역시 유성구 동서대로 125 (덕명동)  
(72) 발명자  
이현빈  
대전광역시 서구 둔산로 155, 109동 703호(둔산동, 크로바아파트)  
김정도  
대전광역시 서구 계룡로 332-8 (갈마동)  
(74) 대리인  
특허법인충정

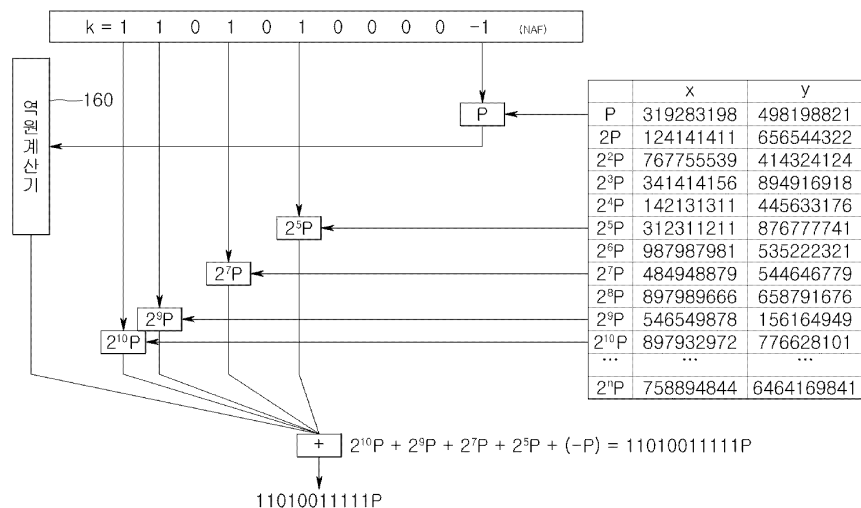
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 **특업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 장치 및 방법**

### (57) 요약

타원 곡선의 스칼라 곱셈 연산 장치는 타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)를 고려하여 특업 테이블의 크기를 결정한 후,  $2^0P$  내지  $2^nP$ 까지의 각각의  $2^iP$ 값을 계산하여 유한체의 곱셈과 분배법칙을 이용하여 특업 테이블을 구성하는 테이블 형성부; 및 타원 곡선 알고리즘에서 사용되는 스칼라 곱셈을  $kP$ 의 하기의 수학적식으로 표현하고,  $k$ 를 바이너리 형태로 변환한 후,  $k$ 의 각 자리수의 계수가 1인 경우의 해당 자리수의  $2^iP$ 값을 특업 테이블에서 스캔하여 로드하여 로드된 각각의 자리수의  $2^iP$ 값을 덧셈 연산하여  $kP$ 에 해당하는 값을 얻는 스칼라 곱셈을 수행하는 스칼라 곱셈 연산부를 포함하는 것을 특징으로 한다.

### 대표도



(52) CPC특허분류

**H04L 9/3066** (2013.01)

**H04L 9/3252** (2013.01)

**H04L 2209/043** (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1425091130

부처명 중소기업청

연구관리전문기관 한국산학연합회

연구사업명 산학협력기술개발

연구과제명 하이브리드 미니 PC 내장 사용자 맞춤형 스마트 키네틱 책상 시스템

기 여 율 1/1

주관기관 한밭대학교산학협력단

연구기간 2014.12.01 ~ 2015.11.30

---

## 명세서

### 청구범위

#### 청구항 1

타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)를 고려하여 룩업 테이블의 크기를 결정한 후,  $2^0P$  내지  $2^nP$ 까지의 각각의  $2^iP$ 값을 계산하여 유한체의 결합과 분배법칙을 이용하여 상기 룩업 테이블을 구성하는 테이블 형성부; 및

타원 곡선 알고리즘에서 사용되는 스칼라 곱셈을  $kP$ 의 하기의 수학식으로 표현하고, 상기  $k$ 를 바이너리 형태로 변환한 후, 상기  $k$ 의 각 자리수의 계수가 1인 경우의 해당 자리수의  $2^iP$ 값을 상기 룩업 테이블에서 스캔하여 로드하여 로드된 각각의 자리수의  $2^iP$ 값을 덧셈 연산하여 상기  $kP$ 에 해당하는 값을 얻는 스칼라 곱셈을 수행하는 스칼라 곱셈 연산부를 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 장치.

[수학식]

스칼라 곱셈 =  $kP$

여기서,  $k$ 는 유한체 상에서 정의되는 임의의 원소이고,  $P$ 는 유한체 상에서 정의된 타원 곡선 위의 임의의 점 ( $P=(x,y)$ )임.

#### 청구항 2

제1항에 있어서,

상기 바이너리 형태로 변환된  $k$ 를 NAF(Non Adjacent Form) 알고리즘을 이용하여 NAF( $k$ )로 변환하는 NAF 변환부를 더 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 장치.

#### 청구항 3

제1항에 있어서,

상기 바이너리 형태로 변환된  $k$ 를 NAF(Non Adjacent Form) 알고리즘을 이용하여 NAF( $k$ )로 변환하는 NAF 변환부를 더 포함하며,

상기 스칼라 곱셈 연산부는 상기  $k$ 의 자리수의 계수가 0인 경우, 다음 자리수를 스캔하고, 상기  $k$ 의 자리수의 계수가 -1인 경우, 역원 계산기를 이용하여  $-2^iP$ 값으로 역원 연산하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 장치.

#### 청구항 4

타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)와 유한체 상에서 정의된 타원 곡선 위의 임의의 점 Base Point  $P=(x,y)$ 와 유한체 상에서 정의되는 임의의 원소인  $k$ 를 입력받는 단계;

상기 파라미터의 크기(n)와 상기  $P$ 를 기초로 룩업 테이블의 크기를 결정하고  $2^0P$  내지  $2^nP$ 까지의 각각의  $2^iP$ 값을 계산하여 유한체의 결합과 분배법칙을 이용하여 상기 룩업 테이블을 구성하는 단계;

타원 곡선 알고리즘에서 사용되는 스칼라 곱셈을  $kP$ 의 하기의 수학식으로 표현하고, 상기  $k$ 를 바이너리 형태로 변환하는 단계; 및

상기  $k$ 의 각 자리수의 계수가 1인 경우의 해당 자리수의  $2^iP$ 값을 상기 룩업 테이블에서 모두 스캔하여 로드하고, 상기 로드된 각각의 자리수의  $2^iP$ 값을 덧셈 연산하여 상기  $kP$ 에 해당하는 값을 얻는 스칼라 곱셈을 수행하는 단계를 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 방법.

[수학식]

스칼라 곱셈 =  $kP$

여기서,  $k$ 는 유한체 상에서 정의되는 임의의 원소이고,  $P$ 는 유한체 상에서 정의된 타원 곡선 위의 임의의 점임.

#### 청구항 5

제4항에 있어서,

상기  $k$ 를 바이너리 형태로 변환하는 단계는,

상기 바이너리 형태로 변환된  $k$ 를 NAF(Non Adjacent Form) 알고리즘을 이용하여 NAF( $k$ ) 형태로 변환하는 단계를 더 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 방법.

#### 청구항 6

제4항에 있어서,

상기  $k$ 를 바이너리 형태로 변환한 후, 상기  $k$ 의 자리수의 계수가 -1인 경우의 해당 자리수의  $2^iP$ 값을 상기 룩업 테이블에서 스캔하여 로드하는 단계; 및

역원 계산기를 이용하여 상기 로드된  $2^iP$ 값을  $-2^iP$ 값으로 역원 연산하며 상기 로드된 각각의 자리수의  $2^iP$ 값과 덧셈 연산을 수행하는 단계를 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 방법.

#### 청구항 7

제4항에 있어서,

상기  $k$ 를 바이너리 형태로 변환한 후, 상기  $k$ 의 자리수의 계수가 0인 경우, 해당 자리수의  $2^iP$ 값을 상기 룩업 테이블에서 스캔하는 작업을 수행하지 않으며 상기 바이너리 형태의  $k$ 의 다음 자리수를 스캔하도록 자리수를 이동하는 단계를 포함하는 것을 특징으로 하는 타원 곡선의 스칼라 곱셈 연산 방법.

### 발명의 설명

### 기술 분야

[0001] 본 발명은 스칼라 곱셈 연산 장치에 관한 것으로서, 특히 기존 스칼라 곱셈 방식을 대신하여  $2^iP$ 에 해당하는 타원 곡선 점의 값( $x, y$ )를 미리 연산하여 룩업 테이블을 구성한 후 룩업 테이블을 활용하여 스칼라 곱셈을 효율적으로 처리하는 룩업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 장치 및 방법에 관한 것이다.

### 배경 기술

[0002] 타원 곡선 스칼라 곱셈은 타원 곡선 암호화 및 페어링에 사용되는 필수적인 연산이다. 타원 곡선 스칼라 곱셈은 타원 곡선 알고리즘의 주요 연산 중 하나이며 암호화와 복호화 동작에서 대부분의 시간을 차지한다.

[0003] 타원 곡선 스칼라 곱셈은  $kP$ 로 표현되고  $kP$ 는 타원 곡선에서 정의되는 점의 덧셈 연산을 통해 처리된다. 여기서,  $k$ 는  $GF(p)$ 상에서 정의되는 임의의 원소이고  $P$ 는  $GF(p)$ 상에서 정의된 타원 곡선 위의 임의의 점이다.

[0004] 타원 곡선 알고리즘의 성능에 직접적 영향을 미치는 스칼라 곱셈은 두 점의 덧셈 연산을 반복적으로 수행한다. 일반적으로 상수는 알고리즘의 보안성을 위해 매우 큰 소수가 사용되므로 큰 오버헤드를 유발한다.

[0005] 타원 곡선 알고리즘은 타원 곡선 이산대수 문제의 어려움을 바탕으로 타 공개키에 비하여 짧은 키 크기에도 동일한 안정성을 제공하기 때문에 전자서명, 키 교환 프로토콜에서 매우 중추적인 역할을 하고 있다.

[0006] 그러나 National Institute Standards and Technology(NIST)에서 2000년대 후반을 기점으로 암호 알고리즘에 대한 키 길이를 해쉬함수(SHA)의 보안 강도를 일정 수준 이상으로 개선된 수준이어야 할 것으로 권고한 바 있다.

[0007] 따라서, 현재 우리나라에서도 기존 160 비트 수준의 키 길이에서 224 비트 이상의 큰 파라미터를 권고하는 추세이다.

[0008] 타원 곡선 알고리즘에서 스칼라 곱셈은 알고리즘 수행 시간의 대부분을 차지하는 핵심 연산으로 파라미터의 증

가로 인하여 큰 오버헤드를 유발하게 된다.

[0009] 이는 전자서명 생성 및 전자서명 인증 시간의 증가, 인터넷 또는 IoT(Internet of Things) 환경에서의 품질 저하로 이어진다.

## 발명의 내용

### 해결하려는 과제

[0010] 이와 같은 문제점을 해결하기 위하여, 본 발명은 기존 스칼라 곱셈 방식을 대신하여  $2^n P$ 에 해당하는 타원 곡선 점의 값(x,y)를 미리 연산하여 록업 테이블을 구성한 후 록업 테이블을 활용하여 스칼라 곱셈을 효율적으로 처리하는 록업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 장치 및 방법을 제공하는데 그 목적이 있다.

### 과제의 해결 수단

- [0011] 상기 목적을 달성하기 위한 본 발명의 특징에 따른 타원 곡선의 스칼라 곱셈 연산 장치는,
- [0012] 타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)를 고려하여 록업 테이블의 크기를 결정한 후,  $2^0 P$  내지  $2^n P$ 까지의 각각의  $2^n P$ 값을 계산하여 유한체의 곱셈과 분배법칙을 이용하여 록업 테이블을 구성하는 테이블 형성부; 및
- [0013] 타원 곡선 알고리즘에서 사용되는 스칼라 곱셈을  $kP$ 의 하기의 수학식으로 표현하고, k를 바이너리 형태로 변환한 후, k의 각 자리수의 계수가 1인 경우의 해당 자리수의  $2^n P$ 값을 록업 테이블에서 스캔하여 로드하여 로드된 각각의 자리수의  $2^n P$ 값을 덧셈 연산하여  $kP$ 에 해당하는 값을 얻는 스칼라 곱셈을 수행하는 스칼라 곱셈 연산부를 포함하는 것을 특징으로 한다.
- [0014] [수학식]
- [0015] 스칼라 곱셈 =  $kP$
- [0016] 여기서, k는 유한체 상에서 정의되는 임의의 원소이고, P는 유한체 상에서 정의된 타원 곡선 위의 임의의 점( $P=(x,y)$ )임.
- [0017] 본 발명의 특징에 따른 타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)와 유한체 상에서 정의된 타원 곡선 위의 임의의 점 Base Point  $P=(x,y)$ 와 유한체 상에서 정의되는 임의의 원소인 k를 입력받는 단계;
- [0018] 파라미터의 크기(n)와 상기 P를 기초로 록업 테이블의 크기를 결정하고  $2^0 P$  내지  $2^n P$ 까지의 각각의  $2^n P$ 값을 계산하여 유한체의 곱셈과 분배법칙을 이용하여 록업 테이블을 구성하는 단계;
- [0019] 타원 곡선 알고리즘에서 사용되는 스칼라 곱셈을  $kP$ 의 하기의 수학식으로 표현하고, k를 바이너리 형태로 변환하는 단계; 및
- [0020] k의 각 자리수의 계수가 1인 경우의 해당 자리수의  $2^n P$ 값을 록업 테이블에서 모두 스캔하여 로드하고, 로드된 각각의 자리수의  $2^n P$ 값을 덧셈 연산하여  $kP$ 에 해당하는 값을 얻는 스칼라 곱셈을 수행하는 단계를 포함하는 것을 특징으로 한다.
- [0021] [수학식]
- [0022] 스칼라 곱셈 =  $kP$
- [0023] 여기서, k는 유한체 상에서 정의되는 임의의 원소이고, P는 유한체 상에서 정의된 타원 곡선 위의 임의의 점임.

### 발명의 효과

- [0024] 진술한 구성에 의하여, 본 발명은 스칼라 곱셈( $kP$ )에서 가장 많은 자원을 차지하는 k를 바이너리 형태로 변환한 후 2의 n승에 해당하는 P가 미리 테이블화되어 스칼라 곱셈시 활용함으로써 연산 속도를 향상시키는 효과가 있다.
- [0025] 본 발명은 스칼라 곱셈시 NAF 알고리즘을 적용하여 계수가 0인 경우 해당 자리수의 연산을 회피할 수 있어 스칼라

라 곱셈의 연산 속도를 향상시키는 효과가 있다.

### 도면의 간단한 설명

- [0026] 도 1은 타원 곡선 알고리즘에서의 두 점의 덧셈 연산을 나타낸 도면이다.
- 도 2는 본 발명의 실시예에 따른 록업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 장치의 구성을 나타낸 도면이다.
- 도 3은 본 발명의 실시예에 따른 록업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 방법을 나타낸 도면이다.
- 도 4는 본 발명의 실시예에 따른 록업 테이블의 구성을 나타낸 도면이다.
- 도 5는 본 발명의 실시예에 따른 k를 NAF 변환한 일례를 나타낸 도면이다.
- 도 6은 본 발명의 실시예에 따른 스칼라 곱셈을 수행하는 방법을 나타낸 도면이다.
- 도 7은 본 발명의 실시예에 따른 역원 계산기의 개념을 나타낸 도면이다.

### 발명을 실시하기 위한 구체적인 내용

- [0027] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0028] 타원 곡선 알고리즘은 타원 곡선의 이산대수문제의 어려움을 기반으로 하여 타 공개키 알고리즘에 비해 적은 파라미터로 강력한 보안성을 제공하는 공개 키 알고리즘이다.
- [0029] 타원 곡선 알고리즘은 소수체 또는 이진체에서 정의되는데 정의되는 유한체에 따라 타원 곡선 생성 방정식이 약간씩 다르다.
- [0030] 소수체에서의 타원 곡선 알고리즘은 다음의 [수학식 1]에서 생성되는 점(x,y)들을 이용하는 알고리즘으로 '공개 키', '메시지' 등이 모두 타원 곡선 상의 점(x,y)에 비유된다.

### 수학식 1

[0031] 
$$y^2 \equiv x^3 + ax + b \pmod{p}$$

- [0032] 여기서, x,y가 수학식 1을 만족하는 값일 때 타원 곡선 점을 나타내고, a, b는 임의의 양의 정수이고, 타원 곡선 방정식에 사용되며, p는 소수로 modulo 연산자이다.

[0033]  $a, b \in GF(p)$  는 관계를 가지며, GF(P)는 [0 내지 p-1]의 원소를 갖는 유한체를 나타낸다.

- [0034] 이러한 특징으로 인하여 타원 곡선 알고리즘에서 사용되는 연산은 점과 점을 이용하여 다른 점을 찾는 연산인테 이를 두 점의 덧셈이라 한다.

- [0035] 도 1은 타원 곡선 알고리즘에서의 두 점의 덧셈 연산을 나타낸 도면이다.

- [0036] 타원 곡선 알고리즘에서 사용되는 두 점의 덧셈은 도 1에 도시된 바와 같이, 3가지 형태로 구분된다.

- [0037] (1) 점 P와 점 Q가 서로 다른 점일 때의 연산으로 점의 덧셈 연산이라고 나타내며, 다음의 [수학식 2]와 같다.

## 수학식 2

$$P = (x_1, y_1), Q = (x_2, y_2) \text{ then } P + Q = R(x_3, y_3)$$

$$x_3 = \lambda^2 - (x_1 + x_2), y_3 = \lambda(x_1 - x_3) - y_1$$

$$\text{where } \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

[0038]

[0039]

여기서,  $x_1, y_1, x_2, y_2, x_3, y_3$ 는 임의의 타원 곡선점을 나타내고,  $\lambda$ 는 기울기를 나타낸다.

[0040]

(2) 점 P와 점 Q가 같은 점일 때의 연산으로 점의 두 배 연산이라고 나타내며, 다음의 [수학식 3]과 같다.

## 수학식 3

$$P = (x_1, y_1), P = (x_2, y_2) \text{ then } P + P = R(x_3, y_3)$$

$$x_3 = \lambda^2 - (x_1 + x_2), y_3 = \lambda(x_1 - x_3) - y_1$$

$$\text{where } \lambda = \frac{3x_1^2 + a}{2y_1}$$

[0041]

[0042]

(3) 점 P와 점 Q가 서로 대칭 점 또는 역원인 경우는 두 점의 덧셈 결과값이  $\infty$ 이고, 이를 무한원점이라 한다.

[0043]

암호학은 양의 정수만을 사용하므로 도 1에 도시된 그래프의 경우 암호학의 이해를 돕기 위하여 사용된 그래프이다.

[0044]

타원 곡선 알고리즘에서의 가장 중요한 연산은 스칼라 곱셈이며, 스칼라 곱셈은 어떠한 점  $P = (x, y)$ 에 상수  $k$ 만큼 곱하는 연산이다.

[0045]

스칼라 곱셈은 타원 곡선 알고리즘에서 가장 빈번하게 사용되고 핵심이 되는 연산으로  $kP = P + P + P \dots + P$  형태이다. 즉,  $k$ 가 양의 정수일 때 점 P를  $k$ 번의 점의 덧셈 연산을 나타내고  $kP$ 로 표기된다.

[0046]

$k$ 는 유한체( $GF(p)$ )상에서 정의되는 임의의 원소이고,  $P$ 는  $GF(p)$  상에서 정의된 타원 곡선 위의 임의의 점이다.  $kP$ 는 타원 곡선에서 정의되는 두 점의 덧셈 연산을 통해 처리된다.

[0047]

이때,  $k$ 는 매우 큰 소수이기 때문에 타원 곡선 알고리즘에서 가장 많은 자원을 사용한다.

[0048]

유한체에서 정의되는 타원 곡선 점들은 유한체의 분배법칙, 결합법칙을 만족한다.

[0049]

따라서, 어떠한 점  $P$ 가 있고  $k$ 가 100일 때,  $kP$ 는 적절히 작은 수의 스칼라 곱셈으로 분배되어 계산된 후 결합할 수 있다.

[0050]

일례로,  $100P = P + P + P \dots + P$ ( $P$ 를 100번 더한 값)

[0051]

$$100P = 50P + 50P$$

[0052]

$$100P = 10P + 20P + 30P + 40P$$

[0053]

본 발명은 스칼라 곱셈  $kP$ 를  $2^n$ 승의 스칼라들로 분배하여 계산하는 것이다.

[0054]

다시 말해,  $k$ 를 바이너리 형태로 변환한 후, 스칼라 곱셈할 때 2의  $n$ 승에 해당하는  $P$ 가 미리 테이블화되어 있다면 스칼라 곱셈( $kP$ )가 더욱 세분화될 수 있다.

- [0055] 일례로  $600P \rightarrow 600_{10} = 1001011000_2$
- [0056]  $600P \rightarrow (1*2^9P) * (1*2^6P) * (1*2^4P) * (1*2^3P)$
- [0057] 여기에서 1을 곱한 값은 자기 자신이기 때문에  $2^nP$ 가 계산되어 있다면 별도의 연산없이 해당 값을 로드하여 최종적으로 더하면 된다(계수가 0일 경우 무시함).
- [0058] 도 2는 본 발명의 실시예에 따른 룩업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 장치의 구성을 나타낸 도면이고, 도 3은 본 발명의 실시예에 따른 룩업 테이블을 이용한 타원 곡선의 스칼라 곱셈 연산 방법을 나타낸 도면이고, 도 4는 본 발명의 실시예에 따른 룩업 테이블의 구성을 나타낸 도면이다.
- [0059] 본 발명의 실시예에 따른 룩업 테이블을 이용한 타원 곡선 스칼라 곱셈 가속화 장치(100)는 입력부(110), 테이블 형성부(120), NAF 변환부(130), 스칼라 곱셈 연산부(140), 저장부(150) 및 역원 계산기(160)를 포함한다.
- [0060] 입력부(110)는  $k$ 값,  $a$ 값,  $p$ 값,  $P$ 값을 입력받는데(S100),  $(k, a, p, P) \in GF(p)$ 에 속해야 한다.  $GF(p)$ 는  $[0$  내지  $p-1]$ 의 원소를 갖는 유한체를 나타내고,  $k$ 는 스칼라 곱셈에서  $kP$ 에서  $k$ 값을 나타내는 것으로 예를 들어 타원 곡선 알고리즘이 160-bit 파라미터에서 구현이 되었다고 가정하면, 1 내지  $2^{160}$ 을 의미한다.  $a$ 는 전술한 수학식 3에서 임의의 양의 정수이고,  $p$ 는 소수이며 모듈러 연산자를 나타내며,  $P$ 는 Base Point  $P=(x,y)$ 인 타원 곡선의 한 점을 나타낸다.
- [0061]  $k$ 는  $k \in GF(p)$ 와  $[1 < k < n-1]$ 를 만족하는 임의의 소수이며,  $n$ 은 원소의 위수를 의미한다.
- [0062] 이하에서는 설명의 편의를 위해서 타원 곡선 알고리즘이 160-bit 파라미터에서 구현된다고 가정한다.
- [0063] 테이블 형성부(120)는 도 4와 같은 룩업 테이블(122)을  $P$  내지  $2^{160}P$ 까지의  $2^nP$ 값을 모두 포함하도록 구성한다(S102).  $n$ 은 타원 곡선을 구성하는 파라미터(Parameter)의 크기를 의미한다.
- [0064] 테이블 형성부(120)는 타원 곡선을 구성하는 파라미터(Parameter)의 크기( $n$ )를 고려하여 룩업 테이블(122)의 크기를 결정한 후,  $2^0P$  내지  $2^nP$ 까지의 각각의  $2^nP$ 값을 계산하여 유한체의 결합과 분배법칙을 이용하여 룩업 테이블(122)을 구성한다.
- [0065] 이러한 룩업 테이블(122)은 유한체의 결합법칙과 분배법칙을 활용한다. 도 4에 도시된 바와 같이,  $P$ 는 Base Point  $P=(x,y)$ 로  $x = 319283198$ ,  $y = 498198821$ 로 미리 설정되는 값이다.
- [0066] 테이블 형성부(120)는 예를 들면,  $2P = P + P$ 의 경우, 유한체의 결합법칙과 분배법칙과 전술한 [수학식 2]와 [수학식 3]을 이용하여  $2P$ 의  $x$ ,  $y$ 값을 계산하며, 나머지  $2^nP$ 의  $x$ ,  $y$ 값도 동일한 원리로 계산된다.
- [0067] NAF 변환부(130)는 정수  $k$  즉 1 내지  $2^{160}$  중 하나를 입력받아 NAF(Non Adjacent Form) 알고리즘을 이용하여 NAF( $k$ ) 형태로 변환한다(S104).
- [0068] NAF 알고리즘은 공지된 기술로 바이너리 형태의 어떠한 수에서 1을 0 또는 -1로 치환하여 1의 개수를 줄이는 알고리즘으로 기존의 타원 곡선 알고리즘에서 덧셈 후 곱셈 알고리즘을 사용할 때 점의 덧셈 연산을 회피하기 위해서 사용하는 알고리즘이다.
- [0069] NAF 알고리즘의 일례로 나타내면, 다음의 [표 1]과 같다.

표 1

| NAF Result           | Demical Operation   |
|----------------------|---------------------|
| $(0 \ 1 \ 1 \ 1)_2$  | $4 + 2 + 1 = 7$     |
| $(1 \ 0 \ -1 \ 1)_2$ | $8 - 2 + 1 = 7$     |
| $(1 \ -1 \ 1 \ 1)_2$ | $8 - 4 + 2 + 1 = 7$ |
| $(1 \ 0 \ 0 \ -1)_2$ | $8 - 1 = 7$         |

[0070]

- [0071] 7을 바이너리 형태로 표현하는 경우,  $111_2$ 가 된다. 여기서,  $111_2$ 을 NAF 형태로 표현한다면,  $(100-1)_2$ 로 표현할 수 있으며, -1은 1에 해당되는 P값의 덧셈의 역원으로 대체할 수 있다.
- [0072] 덧셈의 역원은 예를 들어  $P = (x, y)$ 일 때,  $-P = (x, -y)$ 로 나타낼 수 있다.
- [0073] 본 발명은 0의 경우 계산을 회피하므로 NAF 알고리즘을 이용하여 1의 개수를 줄이고 0의 개수를 늘린다면 스칼라 곱셈 연산의 효율을 증대할 수 있는 것이다.
- [0074] 스칼라 곱셈 연산부(140)는 룩업 테이블(122)을 구성한 후, 입력부(110)로부터 정수 k를 입력받는 경우, 스칼라 곱셈(kP)를 수행하게 된다.
- [0075] 예를 들면, 룩업 테이블(122)을 구성한 후,  $k = 1695_{10}$ 일 때, k는 NAF 변환부(130)를 통해 도 5와 같이 변환될 수 있다.
- [0076] 도 5와 같이, NAF 변환을 수행하면 스칼라 연산 횟수가 8번에서 5번으로 줄어든다.
- [0077] 스칼라 곱셈 연산부(140)는 변환된 NAF값과 룩업 테이블(122)을 이용하여 스칼라 곱셈 연산을 수행하게 된다.
- [0078] 스칼라 곱셈 연산부(140)는  $NAF(k)_{Length} \geq 0$  동안 스칼라 곱셈 연산을 수행하게 되는데(S106),  $NAF(k_i)$ 의  $i=1$ 일 때  $k_i=1$ 인 경우, 룩업 테이블(122)을 스캔하여  $2^iP$ 값을 로드하여 저장부(150)에 저장된 kP값과 더하여 저장부(150)에 누적하여 저장하며, i 변수를 i++를 수행한다(S108, S110). 여기서, i는 NAF(k) 필드의 위치를 나타내는 변수이다.
- [0079] 스칼라 곱셈 연산부(140)는  $NAF(k_i)$ 의 해당 자리수의 계수가 0인 경우( $k_i=0$ ), i 변수만을 i++를 수행한 후(S112), 단계 S106로 진행된다.
- [0080] 도 6을 참조하여 설명하면,  $NAF(k_i)$ 의  $i=1$ 일 때  $2^{10}P$ 값을 룩업 테이블(122)에서 스캔하고,  $NAF(k_i)$ 의  $i=2$ 일 때  $k_i=1$ 인 경우,  $2^9P$ 값을 룩업 테이블(122)에서 스캔하여  $2^{10}P$ 값과 누적하고,  $NAF(k_i)$ 의  $i=4$ 일 때  $k_i=1$ 인 경우,  $2^7P$ 값을 룩업 테이블(122)에서 스캔하여  $2^{10}P$ 값 +  $2^9P$ 값과 누적하며,  $NAF(k_i)$ 의  $i=6$ 일 때  $k_i=1$ 인 경우,  $2^5P$ 값을 룩업 테이블(122)에서 스캔하여  $2^{10}P$ 값 +  $2^9P$ 값 +  $2^7P$ 값과 누적한다.
- [0081] 스칼라 곱셈 연산부(140)는 NAF(k)에서 첫 번째 자리부터 스캔하여 계수가 1일 경우 해당 자리수에 해당하는  $2^nP$ 값을 가져오며, 해당 자리수가 0일 경우 룩업 테이블(122)에서 스캔하는 작업을 수행하지 않고, i++하여 다음 자리수를 스캔하기 위하여 단계 S106로 진행한다(S112).
- [0082] 스칼라 곱셈 연산부(140)는  $NAF(k_i)$ 의  $i=11$ 일 때  $k_i=-1$ 인 경우, 해당 자리수에 해당하는  $2^nP$ 값을 로드하여 역원 계산기(160)를 이용하여 역원값을 계산한다(S114, S116). 여기서, 역원은 해당 자리수의  $P = (x, y)$ 일 때,  $-P = (x, -y)$ 를 의미한다.
- [0083] 스칼라 곱셈 연산부(140)는  $NAF(k_i)$ 의  $i=11$ 일 때, 해당 자리수의 계수가 -1이므로 룩업 테이블(122)에서 가져온  $2^nP$ 를 역원 계산기(160)에서 역원 연산하여  $-2^nP$ 값을 계산한다.
- [0084] 스칼라 곱셈 연산부(140)는  $NAF(k_i)$ 의 계수가 1인  $i=1, 2, 4, 6, 11$ 일 때 룩업 테이블(122)에서 로드된 모든  $2^nP$ 값을 차례대로 두 점의 덧셈 연산을 수행하면 최종적으로 kP에 해당하는 값을 얻을 수 있다.
- [0085] 본 발명은 타원 곡선을 구성하는 파라미터(Parameter)의 크기(n)를 고려하여 룩업 테이블(122)의 크기를 결정 한 후,  $2^0$  내지  $2^nP$ 에 해당하는 각각의  $P=(x,y)$ 를 계산하여 룩업 테이블(122)을 구성한다.
- [0086] 타원 곡선 알고리즘으로 구현된 보안 시스템은 파라미터가 변경되지 않는다면 한번 구성된 룩업 테이블(122)을 재사용이 가능하기 때문에 타원 곡선 알고리즘의 공개키 생성 및 전자 서명 알고리즘에 유용하다.
- [0087] 스칼라 곱셈 연산부(140)는 스칼라 곱셈을 수행하여 kP에 해당하는 값을 얻어서 암호화, 복호화하는데 필요한 공개키 생성, 전자서명, 보안 인증 등에 활용할 수 있다.
- [0088] 이상에서 설명한 본 발명의 실시예는 장치 및/또는 방법을 통해서만 구현이 되는 것은 아니며, 본 발명의 실시

예의 구성에 대응하는 기능을 실현하기 위한 프로그램, 그 프로그램이 기록된 기록 매체 등을 통해 구현될 수도 있으며, 이러한 구현은 앞서 설명한 실시예의 기재로부터 본 발명이 속하는 기술분야의 전문가라면 쉽게 구현할 수 있는 것이다.

[0089]

이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

### 부호의 설명

[0090]

100: 스칼라 곱셈 연산 장치

110: 입력부

120: 테이블 형성부

130: NAF 변환부

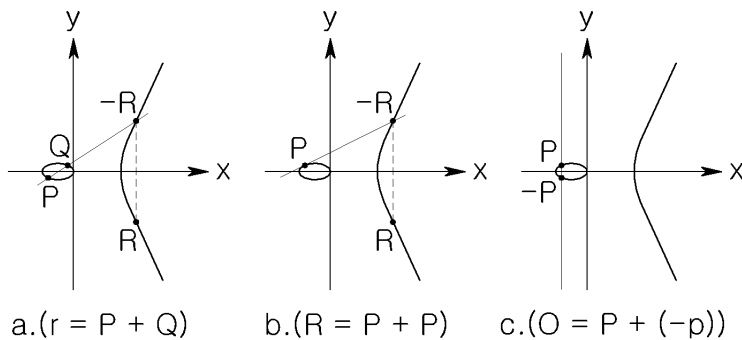
140: 스칼라 곱셈 연산부

150: 저장부

160: 역원 계산기

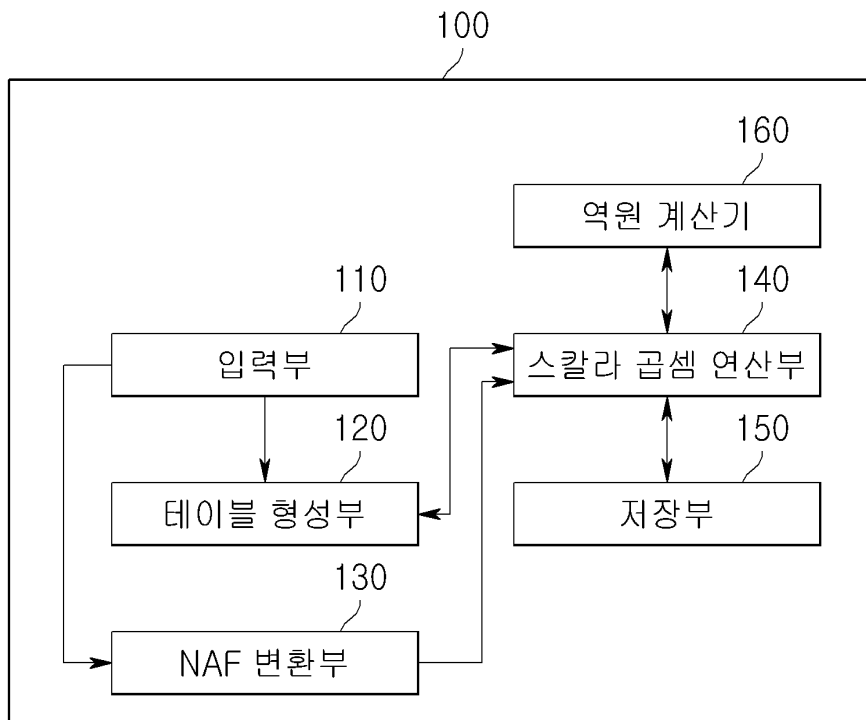
### 도면

#### 도면1

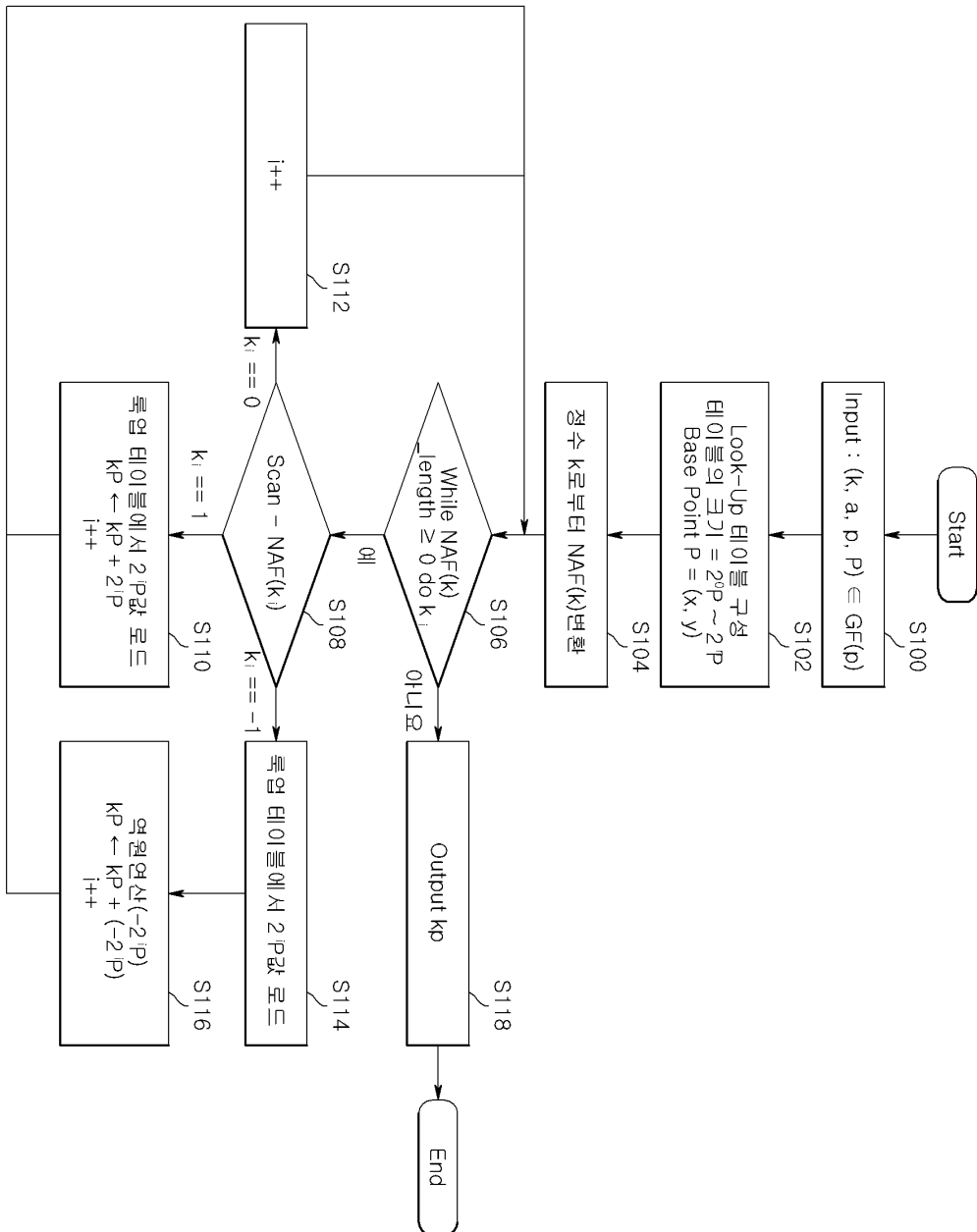


(두점의 덧셈 연산)

도면2



도면3



도면4

$$2^3P = 2 * 2^2P \text{ or } 2P + 2P$$

$$2^3P = 2 * 2^3P \text{ or } 2^3P + 2^3P$$

$$\vdots$$

$$2^{160}P = 2 * 2^{159}P \text{ or } 2^{159}P + 2^{159}P$$

122

|                  |           |            |
|------------------|-----------|------------|
|                  | x         | y          |
| P                | 319283198 | 498198821  |
| 2P               | 124141411 | 656544322  |
| 2 <sup>2</sup> P | 767755539 | 414324124  |
| 2 <sup>3</sup> P | 341414156 | 894916918  |
| 2 <sup>4</sup> P | 142131311 | 445633176  |
| 2 <sup>5</sup> P | 312311211 | 876777741  |
| 2 <sup>6</sup> P | 987987981 | 535222321  |
| 2 <sup>7</sup> P | 484948879 | 544646779  |
| 2 <sup>8</sup> P | 897989666 | 658791676  |
| 2 <sup>9</sup> P | 546549878 | 156164949  |
| ...              | ...       | ...        |
| 2 <sup>n</sup> P | 758894844 | 6464169841 |

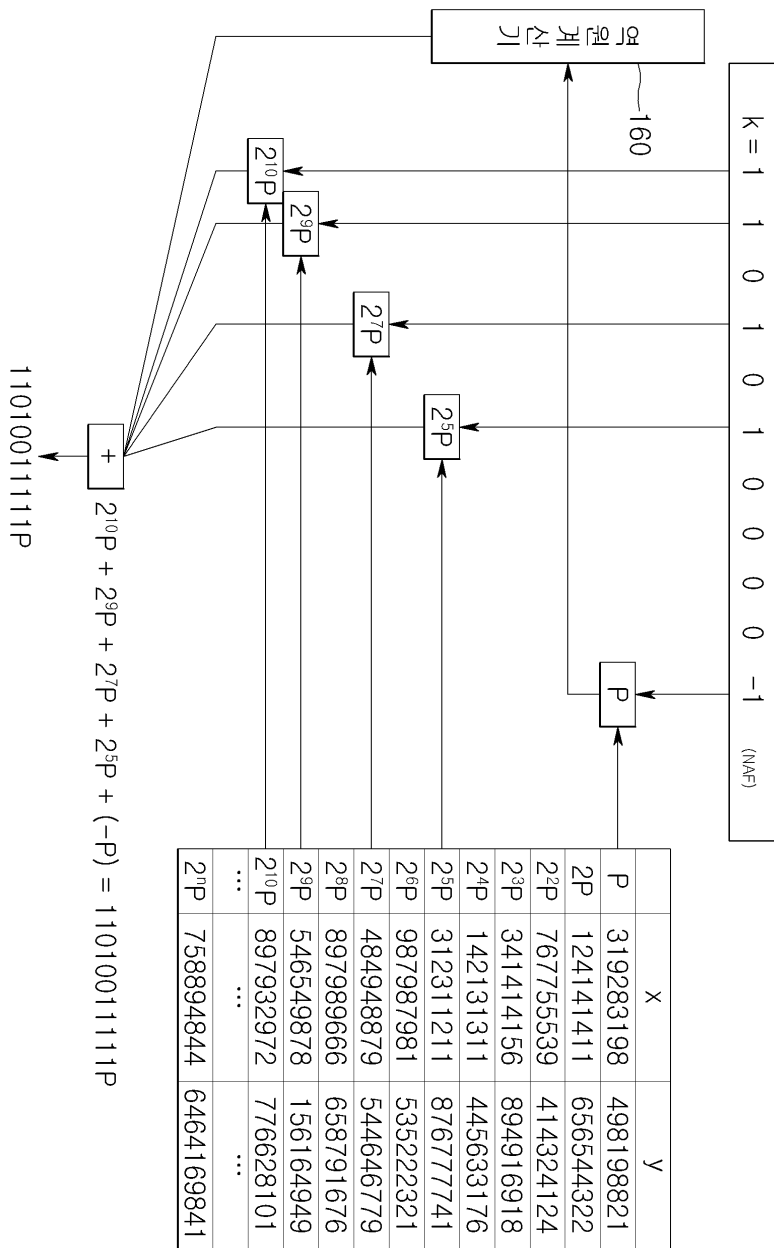
2<sup>n</sup>승 Look-Up테이블의 구성

도면5

|                                                          |       |
|----------------------------------------------------------|-------|
| $k = 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ \boxed{1 \ 1 \ 1 \ 1 \ 1}$  | (2)   |
| $k = 1 \ 1 \ 0 \ 1 \ 0 \ \boxed{1 \ 0 \ 0 \ 0 \ 0 \ -1}$ | (NAF) |

k = 1695<sub>(10)</sub>일때 k의 변환

도면6



도면7

