이제는 알 때가 됐다! - ECC 알고리즘

김미영
(wickedme02@gmail.com)

ECC(타원곡선 암호화) 알고리즘

Concept	타원곡선의 정의 및 연산 타원곡선의 이산로그 문제에 기반한 공개키 암호화 방식 블록체인(비트코인)에 적용된 타원곡선 암호(ECDSA)
KeyWord	ECC, 타원곡선(Elliptic Curve), 공개키 암호화, ECDSA(Elliptic Curve Digital Signature Algorithm), ECDH

최근 들어 타원곡선 기반 암호화(ECC) 알고리즘이 주목을 받고 있으며, 채택 속도가 빨라지고 있다. 타원곡선 암호화 알고리즘은 비트코인의 소유권 증명, 미국 정보의 내부 통신 보호, Apple 의 iMessage 서비스에도 DNS 정보를 암호화하는데 사용되고 있다. 또한 IoT 경량 암호화로 ECC 알고리즘이 주목받고 있으며, 간편결제 인증에서 ECC 암호화를 채택하는 사례가 등장하고 있다

타원곡선 암호(ECC)는 공개키 암호화 방식 중 하나로 유한체(Finite field) 위에서의 타원곡선의 대수적 구조를 기반으로 한 이산로그문제에 착안해 만들어졌다. 타원곡선 암호는 기존의 공개키 암호에 비해 더 적은 비트로 동일한 안전성을 얻을 수 있고 빠른 속도로 암호화를 처리하며, 키(key) 관리가 용이한 장점이 있다.

타원곡선 암호화(ECC)의 기반이 되는 타원곡선의 개념과 타원곡선을 이용한 암호화 원리, 그리고 끝으로 비트코인에서의 타원곡선 암호 적용에 대해 알아보자 한다.

1. 타원곡선의 정의

일반적으로 타원 곡선 방정식은 다음과 같은 형태의 3 차 방정식을 이용한다.

$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3 \qquad \text{(식 1)}$$

그러나, 실수상의 타원 곡선은 다음과 같은 특별한 범주에 속하는 타원 곡선을 사용한다.

$$y^2 = x^3 + ax + b \qquad \text{(식 2)}$$

타원곡선은 일반적으로 타원 형태의 그래프를 생각하기 쉬우나, 타원곡선은 그림 1 과 같은 형태의 곡선을 가지며 (식 2)의 a 와 b 의 값에 따라 다양한 형태의 타원 곡선을 정의할 수 있다.

타원곡선의 흥미로운 특징은 x 축을 중심으로 대칭되며, 비 수직선에 대해 최대 3 개 지점에서 곡선과 교차한다는 점이다.

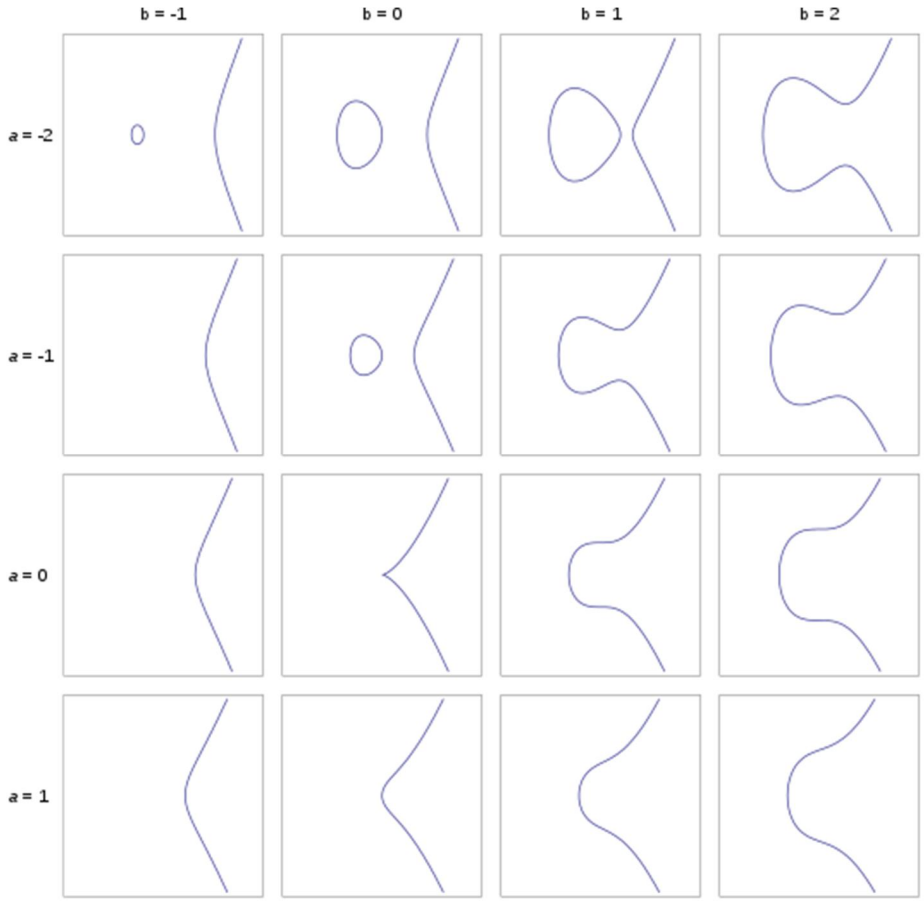

그림 1. 타원곡선 $y^2 = x^3 + ax = b$들의 그래프

(출처: wikipedia)

2. 타원곡선 상에서의 연산

타원 곡선을 이용한 암호화를 이해하기 위해서는 타원곡선상의 덧셈 연산을 이해해야 한다.
타원곡선의 덧셈 연산을 기하학적으로 설명해 보면, 타원곡선상의 P 와 Q 의 덧셈연산은 점 P 와 Q 를 지나는 직선이 타원과 만나는 제 3 의 교점(-R)을 x 축으로 대칭시킨 점을 P+Q=R 로 정의한다. (그림 2. a)

P 와 Q 가 같은 경우에는 즉, P+P 의 연산은 P 점에서 접선을 그었을 때 타원과 만나는 제 3 의 교점(-R)을 x 축으로 대칭시킨 점(R)에 해당한다. (그림 2. b)

또한 무한대 값 "0" 이 가능하여, P+(-P) = 0 으로 P 와 x 축 대칭점 -P 와의 덧셈 연산 결과는 무한대 값을 가진다. (그림 2. c)

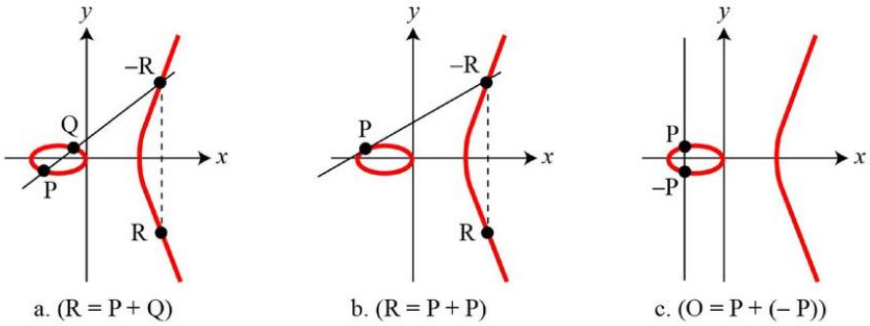

그림 2. 타원곡선상의 덧셈 연산

(출처: http://slidesplayer.org/slide/11329530/)

3. 타원곡선의 이산로그 문제에 기반한 타원곡선 암호화

타원곡선 암호의 원리 및 키생성

타원곡선 암호는 타원곡선 위의 이산로그 문제가 어렵다는 사실을 이용한 공개키 암호 방식이다. 타원곡선 암호화를 위한 타원곡선은 임의의 정수 a, b 에 대해 정의된 다음과 같은 방정식의 해 (X, Y) 의 집합이다.

$$Y^2 = X^3 + aX + b \ \ (mod\ p) \qquad \text{(식 3)}$$

점 P=(x, y)가 타원곡선 상에 있다는 것은 위 방정식을 만족시킨다는 뜻이고, 두 점 P, Q 와 임의의 정수 x 에 대해 다음 방정식을 정의할 수 있다.

$$Q = xG \qquad \text{(식 4)}$$

이때 해 x 를 구하는 것이 타원곡선 이산대수 문제이다. 이로부터 타원곡선 암호에서 사용하는 키 쌍은 다음과 같이 정의할 수 있다.

G : 생성자, 임의의 시작 포인트
x : 개인키, P 보다 적은 소수(Prime)로, 난수 생성기로 생성
Q : 공개키, 개인키로부터 연산

이때 공개키 Q 는 Q = x*G = G+G+.....G (x 번 덧셈) 으로 G 를 x 번 덧셈연산한 값이다.

Q=xG 수식에서 x 와 G 를 이용해서 Q 를 구하기는 쉽지만, G 와 Q 를 안다고 해서, x 값을 유추해 내기가 굉장히 어려운 타원곡선 이산대수 문제를 이용한다. G 는 타원 곡선상의 임의의 점이며 x*G 는 G 를 타원곡선상에서 x 번 덧셈 연산한 것을 의미하며, 그림 3 은 x*G 의 연산 과정을 기하학적으로 도식화하여 보여준다.

앞에서 설명한 타원곡선상의 덧셈 연산에서 P+P 연산을 상기시켜보자. 2G=G+G 는 점 G 에서의 접선이 타원곡선과 만나는 제 3 의 점을 x 축으로 대칭시킨 지점이다. 4G = 2G+2G 는 2G 에 해당하는

점에서 마찬가지로 접선을 그어 타원 곡선과 만나는 점의 x 축 대칭 점이다. G 의 상수 배 연산은 이를 반복적으로 수행하여 표현할 수 있으며, 타원곡선상에서 이루어지는 특성을 보인다.

타원 곡선은 공개키 암호 체계를 수학적으로 수행하는 한 가지 방법으로써 타원곡선을 이용하여, RSA, ElGamal, Diffie-Hellman 을 구현할 수 있다.

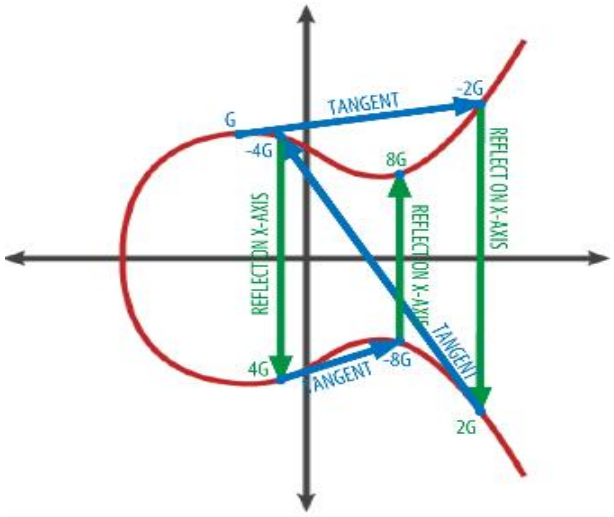

그림 3. 타원곡선상의 Q=x*G 연산

타원곡선을 이용한 Diffie-Hellman (ECDH)

ECDH 는 Diffie-Hellman 키 교환을 타원곡선에 적용한 방법이다. Diffie-Hellman 키 교환의 핵심은 서로 통신하는 쌍방이 같은 키를 공유하는 기법이다.

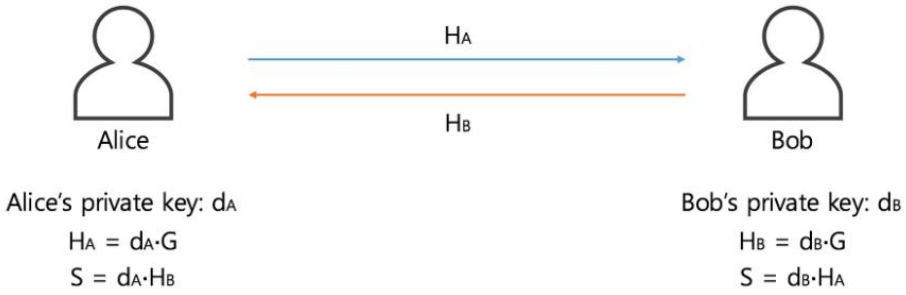

그림 4. 타원곡선을 이용한 DH

그림 4.에서 Alice 와 Bob 은 개인키로 각각 dA 와 dB 를 결정하고, 타원곡선 덧셈연산을 통해 계산한 HA 와 HB 를 상호 교환한다. Alice 는 Bob 로부터 받은 HB 에 개인키 dA 를 타원곡선 덧셈 연산하여 비밀키 S 를 생성하고, Bob 은 Alice 로부터 받은 HA 에 개인키 dB 를 타원곡선 덧셈 연산을 통해 동일한 비밀키 S 를 생성할 수 있다. 따라서 비밀키 교환없이 쌍방이 동일한 비밀키를 공유 가능하다.

타원곡선을 이용한 ElGamal

아래 그림 5.에서 처럼 Bob 은 개인키 d 를 생성하고, 개인키 d 로부터 타원곡선 상에서 계산된 공개키 e_2 를 Alice 에게 전송한다. Alice 는 평문 P 를 Bob 의 공개키를 이용해서 (C_1, C_2) 암호문을 생성하여

Bob 에게 전달하면, Bob 은 개인키 d 를 이용해서 평문 P 를 복호화 한다. 이때 암호화와 복호화 과정에서 사용되는 곱셈연산은 위에서 설명한 타원곡선 덧셈연산 방식에 의해 수행된다.

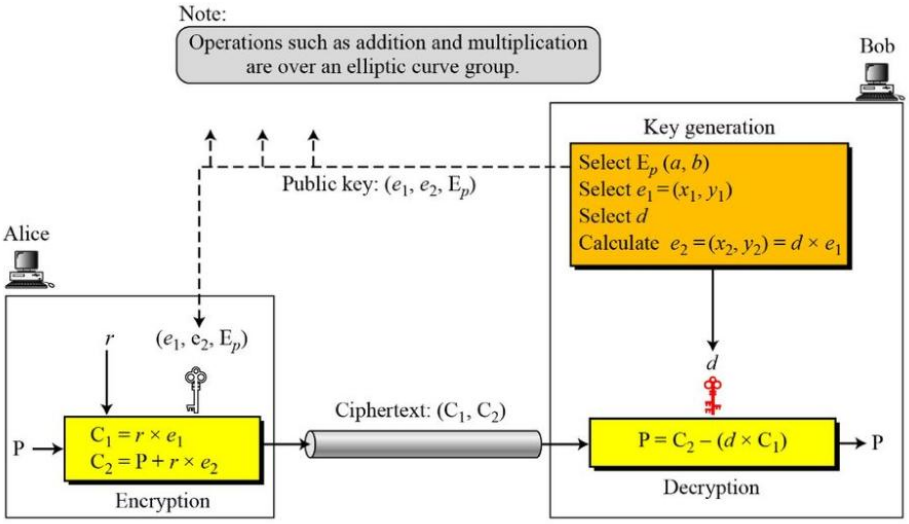

그림 5. 타원곡선을 이용한 ElGamal 암호 체계

(출처: http://slidesplayer.org/slide/11329530/)

이산대수 기반의 ElGamal 과 타원곡선을 이용한 ElGamal 의 키 생성, 암호화, 복호화 방식을 비교하면 다음과 같다.

	ElGamal 공개키 암호	타원곡선 공개키 암호
공개키	$(g, p, y = g^x \bmod p)$	$(G, p, Y = xG)$
개인키	(x)	(x)
암호화	$(c_1, c_2) = (g^{x'} \bmod p,\ y^{x'} m \bmod p)$	$(C_1, C_2) = (x'G, x'Y + M)$
복호화	$c_2\ c_1^{-x} \bmod p$	$C_2 - x\,C_1$

(x'는 임의의 난수이다.)

타원곡선 암호화의 보안성

ECC 는 RSA 보다 매우 짧은 길이의 키를 사용하면서도 비슷한 수준의 안정성을 제공하는 것이 특징이다. 타원곡선암호화의 160 비트 키 길이의 암호강도는 이산대수의 특성을 이용한 비대칭키 알고리즘인 RSA 1024 길이의 키 강도에 해당한다. 키 값이 커질수록 RSA 보다 암호화 레벨이 급격하게 높아진다. 타원곡선 암호화 방식은 더 적은 비트로 동일한 안전성을 얻을 수 있으나, 연산은 더 복잡하다는 단점이 있다.

Symmetric Key Size (bits)	RSA Key Size (bits)	Elliptic Curve Key Size (bits)	Key size ratio
80	1024	160	7 : 1
112	2048	224	9 : 1
128	3072	256	12 : 1
192	7680	384	20 : 1
256	15360	512	30 : 1

Table 1. RSA 와 ECC 키 길이 비교 (NIST 권장 키 길이)

(출처: https://wiki.openssl.org/index.php/Elliptic_Curve_Cryptography)

4. 블록체인(비트코인)의 타원곡선 암호(ECDSA)

비트코인등 블록체인 기반 기술에서는 키 쌍의 생성에 타원곡선 암호(ECDSA) 알고리즘을 사용해 키 길이는 256 비트 이상을 사용한다. 비트코인은 미국국립표준기술원(NIST)에서 개발한 secp256k1 표준에 정의된 타원 곡선을 사용한다. secp256k1 에서의 타원 곡선 수식은 다음과 같다. 위에서 설명한 타원곡선 암호화 방식에 따라 개인키와 개인키로부터 계산된 공개키를 생성한다.

$$y^2 \bmod p = (x^3 + 7) \bmod p \qquad \text{(식 5)}$$