ÖZEL ANAHTAR GENEL ANAHTAR DÖNÜŞTÜRME

Dönüştürmek istediğiniz özel anahtar şu şekildedir:

Özel Anahtar seçelim (Private Key): d = 7

Şimdi, Bitcoin ve diğer kripto paralarda kullanılan standart bir eliptik eğri olan "secp256k1" eğrisini kullanacağız.

Özel anahtara karşılık gelen genel anahtarı aşağıdaki denklemi kullanarak hesaplayın:

Genel Anahtar (Public Key) = d * G

Burada G, eliptik eğri üzerindeki genel başlangıç noktasıdır.

- "secp256k1" eliptik eğrisini yükleyin.
- Özel anahtarı (d) genel başlangıç noktası (G) ile eğri üzerinde çarpın. Bu, eğri üzerindeki ikili noktalarda (x, y) matematiksel işlemler gerçekleştirmenizi gerektirir.

Açıklamalar için, G için aşağıdaki değerleri kullanacağız:
G =
(0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798,
0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)

- Özel anahtarı (d) ve genel başlangıç noktasını (G) eğri üzerinde çarpın, seçili eliptik eğri için belirtilen matematiksel işlemleri kullanarak. Sonuç olarak, genel anahtarı temsil eden yeni bir ikili nokta (x, y) elde edersiniz.
- İşlemin sonucu, genel anahtardır (Public Key) ve dijital imzanızı doğrulamak için kullanılabilir.

2.Adımın detaylı açıklaması:

"secp256k1" eliptik eğrisi üzerinde genel başlangıç noktası (G) ile özel anahtar (d) arasındaki matematiksel işlemleri daha detaylı bir şekilde açıklayabilirim. Aşağıda, adım adım ilerleyelim:

- Genel başlangıç noktası (G) için değerleri hatırlayalım: G = (0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798,
0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)
- Özel anahtar (d) ile G noktasını çarpmak için "secp256k1" eliptik eğrisinde tanımlı matematiksel işlemleri kullanacağız. Bu işlemler şunlardır:

- Eğrinin tanımına göre, iki nokta P ve Q arasındaki toplama işlemi P + Q, belirli koordinatlar üzerinde gerçekleşir. İlgilenen noktaların koordinatları toplanır ve sonucun eğri üzerindeki bir nokta olduğu doğrulanır.
- Eğrinin tanımına göre, bir noktanın kendisiyle toplama işlemi 2P olarak ifade edilir. Bu, koordinatların çiftlenmesi ve sonucun eğri üzerindeki bir nokta olduğunun doğrulanmasıyla gerçekleştirilir.
- Eğrinin tanımına göre, bir noktanın belirli bir sayıyla çarpılması (örneğin, d * P), belirli matematiksel işlemlerle gerçekleştirilir. Bu işlem, çiftlenme, toplama ve sonucun eğri üzerindeki bir nokta olduğunun doğrulanmasıyla yapılır.
- Dolayısıyla, adım adım ilerleyelim:
- G noktasını kendisiyle toplamak için matematiksel işlemleri kullanarak 2G noktasını bulun.
- Bulunan 2G noktasını kendisiyle toplamak için matematiksel işlemleri kullanarak 3G noktasını bulun.
- 3G noktasını kendisiyle toplamak için matematiksel işlemleri kullanarak 4G noktasını bulun.
- Bu şekilde ilerleyerek, döngüyü takip edin ve d * G'ye ulaşana kadar G noktasını kendisiyle toplamaya devam edin.

Sonuç olarak, d * G noktası, özel anahtarın genel anahtara dönüştürüldüğü noktayı temsil eder. Bu işlem, eliptik eğri üzerindeki matematiksel işlemlerin iteratif olarak uygulanmasıyla gerçekleştirilir.

G değeri:

G değeri, "secp256k1" olarak adlandırılan özel bir eliptik eğri üzerindeki genel başlangıç noktasıdır. Bu nokta, Bitcoin ve diğer kripto para birimlerinde kullanılan standart bir eğri olan secp256k1'in bir parçasıdır.

secp256k1 eğrisi, aşağıdaki denklemi sağlayan noktaların kümesidir:

y^2 = x^3 + 7

Burada (x, y), eğri üzerindeki noktaların koordinatlarını temsil eder. Ancak bu denklem tek başına bir eğriyi belirtmez, çünkü başlangıç noktası (G) gibi diğer parametreleri de içermesi gerekmektedir.

Bitcoin'in tasarımında, secp256k1 eğrisi seçildi ve başlangıç noktası (G) şu şekilde belirlendi:

G =
(0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798,
0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)

Bu değerler, secp256k1 eğrisi üzerindeki belirli bir noktanın (x, y) koordinatlarını ifade eder ve genel başlangıç noktası olarak kabul edilir.

Eğri üzerindeki bu belirli başlangıç noktası (G), genellikle hesaplama ve işlem kolaylığı sağlamak amacıyla kullanılır. Örneğin, genel anahtarın hesaplanması veya işlem imzası oluşturma süreçlerinde bu nokta kullanılır.

Bu nedenle, G değeri secp256k1 eğrisindeki bir özelliktir ve Bitcoin protokolü tarafından belirlenmiş bir değeri ifade eder.

G degeri sabitmidir:

Evet, G değeri secp256k1 eğrisinde sabit bir noktadır. Bu, Bitcoin ve diğer kripto para birimlerinde kullanılan ECDSA (Elliptic Curve Digital Signature Algorithm) protokolünün bir parçasıdır.

Genel başlangıç noktası (G), secp256k1 eğrisindeki önceden belirlenmiş bir noktadır ve secp256k1'in tanımına dahildir. Bu başlangıç noktası sabit bir değere sahiptir ve genellikle secp256k1 eğrisindeki diğer matematiksel işlemlerde kullanılır.

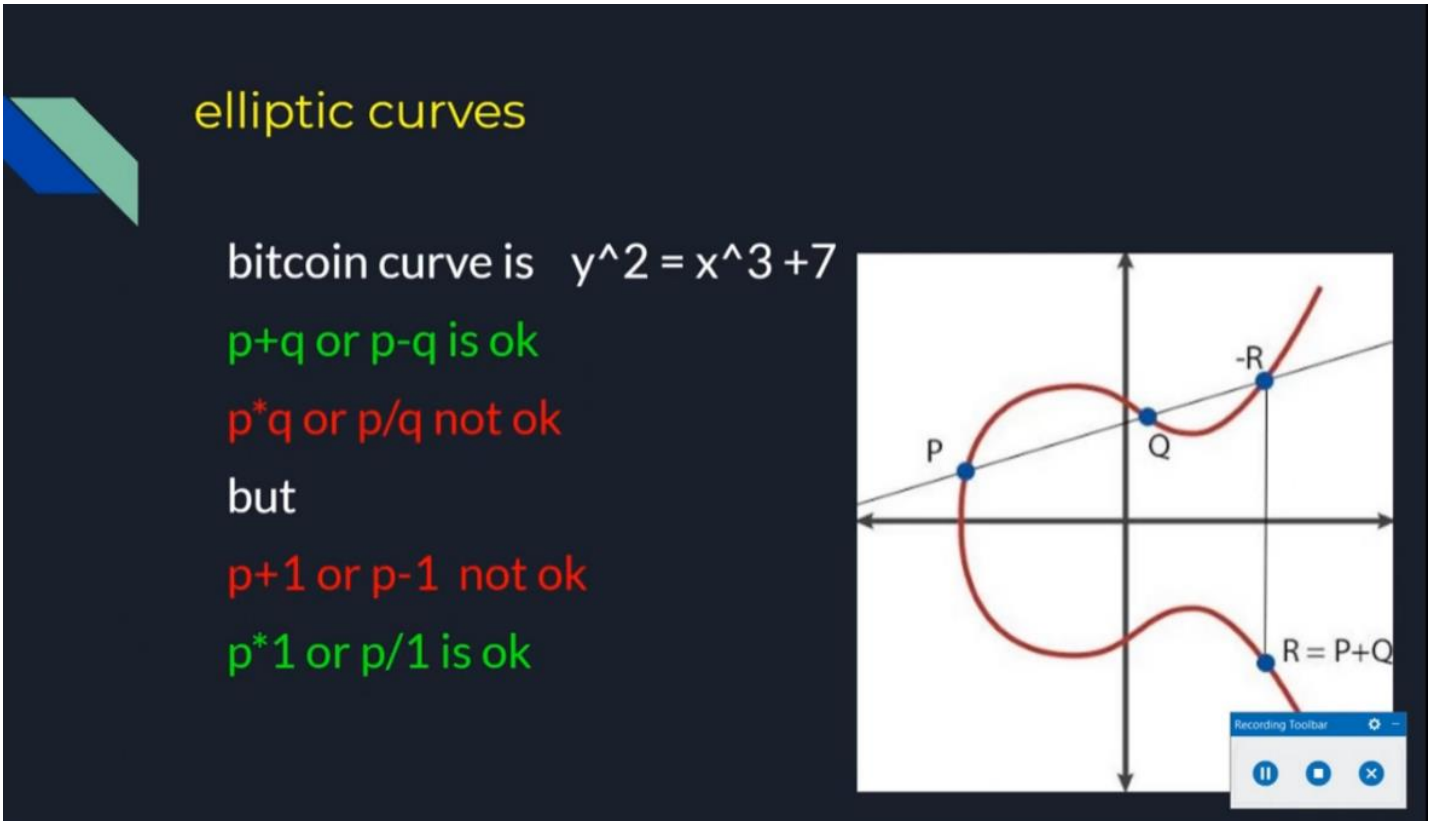
G değeri, aşağıdaki koordinatları ifade eder:

G = (0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798, 0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)

Bu değerler, secp256k1 eğrisi üzerinde belirli bir noktanın (x, y) koordinatlarını temsil eder. G, secp256k1 eğrisinde bir başlangıç noktası olarak kabul edilir ve ECDSA protokolünün uygulanmasında kullanılan bir parametredir.

G sabit bir değer olduğundan, özel anahtardan genel anahtarın hesaplanması için her zaman aynı nokta kullanılır. Bu, genel anahtarın özel anahtarla eşleştirilebilmesini ve doğrulanabilmesini sağlar.

Sonuç olarak, G değeri secp256k1 eğrisindeki sabit bir başlangıç noktasını temsil eder ve ECDSA protokolü tarafından belirlenmiş bir değeri ifade eder.




```
const EC = require('elliptic').ec;
 // ECDSA kütüphanesini kullanarak elliptic eğrisini yükleme
const ec = new EC('secp256k1');
// Private key (d) olarak 7'yi belirleyelim
const privateKey = '7';

// Private key'i hexadecimal formatından Buffer'a dönüştürelim
const privateKeyBuffer = Buffer.from(privateKey, 'hex');

// Private key'i kullanarak public key'i hesaplayalım
const key = ec.keyFromPrivate(privateKeyBuffer);
const publicKey = key.getPublic('hex');

console.log('Public Key:', publicKey);
```

Output:

Public Key:
04c759f8319fc7d9ee29a20f35b5cc3b04210f68f2f6aef23bc587c738a19
055b08b12d240a6b6d08dc35df09d33a30a4d671e881e94df79cb7f95c2
28dc9a53c