KONU-1 : Eliptic Curve

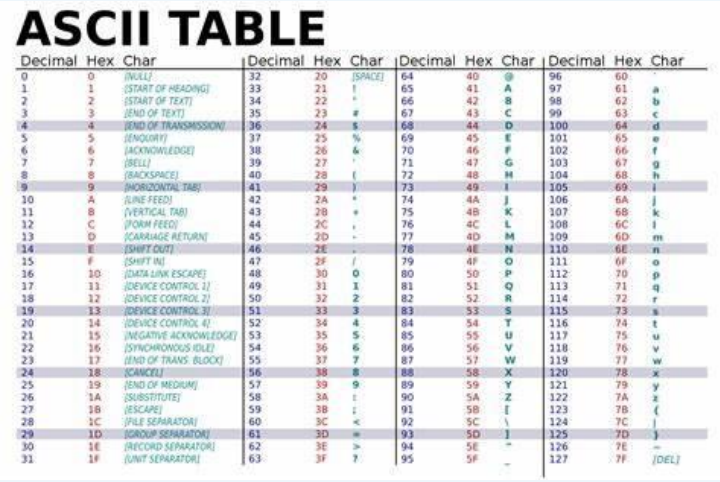
-Eliptic Curve (Kriptoloji) --> Private Key

Eliptik eğri kriptografisi (ECC), kriptografi alanında kullanılan bir şifreleme yöntemidir. ECC, özellikle dijital imzalama, anahtar değişimi ve gizli anahtarlı şifreleme gibi güvenlik uygulamalarında yaygın olarak kullanılır. ECC, diğer şifreleme yöntemlerine kıyasla daha güçlü güvenlik sağlamak için matematiksel bir temele dayanır.

Eski şifreleme yöntemlerinde, ASCII veya Unicode gibi basit mantık kullanılarak şifreleme gerçekleştirilirdi. Örneğin, Sezar şifrelemesi gibi yöntemlerde her harf veya karakter belirli bir sayıya dönüştürülerek şifreleme yapılırdı. Ancak bu tür basit şifreleme yöntemleri kolaylıkla kırılabilir ve güvenlik açıkları içerebilir.

Commented [MW1]: Get the conversation going by adding comments and using Share (above) to send a link to this doc. It's free! No subscription or sign-in necessary.

Açıklama:



ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Bu bizim ascii tablomuzdur, ALİ kelimesini şifrelemek istiyorsak ascii ye göre:

A harfı 65 sayısına eşit
L harfı 76 sayısına eşit

İ harfı 73 sayısına eşit

Şifrelemek için her harfe bir miktar ekleyelim örneğin 5 ekleyelim

A =65 ---> 65+5=70 ---> 70 =F

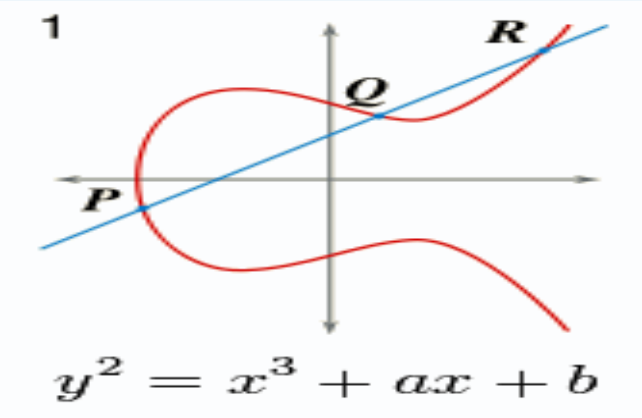
L =76 ---> 76+5=81 ---> 81 =Q

İ =73 ---> 73+5=78 ---> 78 =N

ALİ gibi anlamlı bir kelime (isim) anlamsız olan FQN'e eşit oldu ve bu şekilde gösterilecektir.

Lakin bu şifreleme sistemi istediğimiz kadar güçlü değil, her hangi biri eklediğimiz miktarı tahmin edebilirse veya bulursa şifreleme sistemimiz yıkılacaktır.

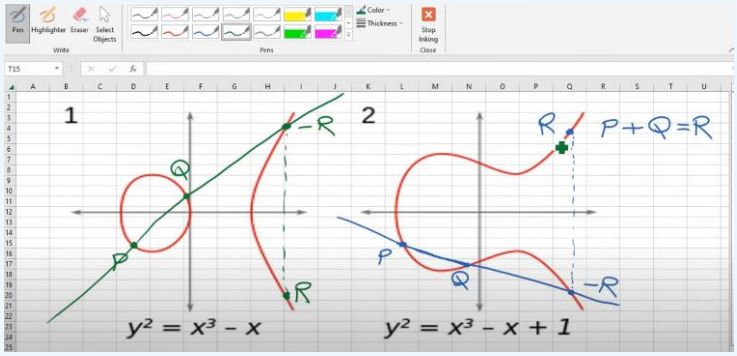
Bu nedenle zaman geçtikçe daha güçlü şifreleme sistemleri keşifedilmektedir, orneğin bugunku konumuz olan elliptic curve cryptography (ECC) .



ECC, daha güçlü bir şifreleme yöntemi olup Sezar şifrelemesiyle doğrudan ilişkili değildir. ECC, matematiksel olarak karmaşık bir algoritma olup eğriler üzerinde çalışır. ECC'de, bir eğri üzerindeki noktaların matematiksel özelliklerinden yararlanarak şifreleme ve anahtar değişimi işlemleri gerçekleştirilir.

ECC, daha az kaynak kullanımı gerektirirken daha yüksek güvenlik düzeyi sağlayabilen bir şifreleme yöntemidir. Bu nedenle, ECC günümüzde yaygın olarak kullanılan kriptografik algoritmalar arasında yer almaktadır.

ECC'de kullanılan eğri, matematiksel bir denklemle tanımlanır. Bu denklemdeki sabit değerler (a ve b) eğrinin şeklini belirlerken, "x" ise bu denkleme uygulanabilen değerlerin kümesini belirler. ECC, eğri üzerinde matematiksel işlemler gerçekleştirerek şifreleme işlemlerini gerçekleştirir.



ECC'deki matematiksel özelliklerden biri, eğri üzerinde iki noktanın (p ve q) toplandığında üçüncü bir nokta (r) elde edilmesidir.

Bu işlem genellikle "nokta toplama" veya "nokta ekleme" olarak adlandırılır. Matematiksel olarak, iki nokta $p(x_1, y_1)$ ve $q(x_2, y_2)$ eğri üzerinde bulunur. Nokta toplama işlemi, bu noktaların x ve y koordinatlarını kullanarak aşağıdaki formülü uygulayarak gerçekleştirilir:

$s = (y_2 - y_1) / (x_2 - x_1)$

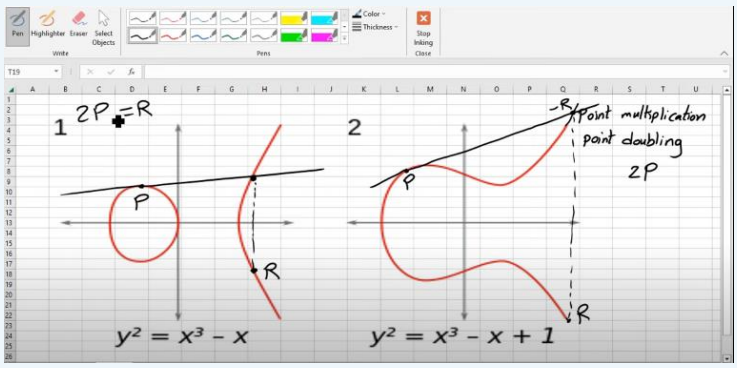
Bu formül, iki noktanın eğri üzerindeki bir doğru üzerinde buluşma eğrisi (slope) olan "s" değerini hesaplar. Bu doğru daha sonra eğri ile kesiştiği üçüncü bir nokta (r) bulmak için kullanılır.

Üçüncü noktanın koordinatlarını bulmak için, aşağıdaki formülleri kullanabiliriz:

$x_3 = s^2 - x_1 - x_2$ $y_3 = s(x_1 - x_3) - y_1$

Bu şekilde, $p(x_1, y_1)$ ve $q(x_2, y_2)$ noktalarının toplamı olarak $r(x_3, y_3)$ bulunur.

ECC'de nokta toplama işlemi, şifreleme ve anahtar değişimi işlemlerinde önemli bir rol oynar. Bu işlem, ECC'nin güvenliği ve etkinliği için temel bir matematiksel özelliktir.



"2P = R" ifadesi, bir noktanın kendisiyle toplanması yerine, bir noktanın iki katının hesaplanmasını ifade eder. İşte nokta iki katlamasının nasıl gerçekleştirildiği:

Verilen bir nokta P(x, y) üzerinde çalışıyoruz ve P'nin iki katını hesaplamak istiyoruz.

- İlk adımda, eğri üzerindeki teğet çizgisinin eğimini hesaplarız. Bunun için aşağıdaki formülü kullanırız:

$s = (3x^2 + a) / (2y)$

Burada "a" eğrinin sabit bir değeridir.

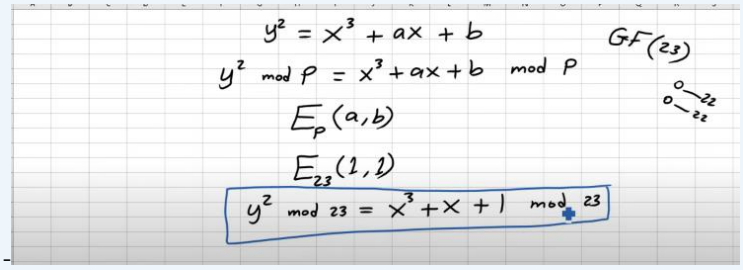
- İkinci adımda, yeni noktanın koordinatlarını hesaplarız. İşte formülleri:

$x_3 = s^2 - 2x$ $y_3 = s(x - x_3) - y$

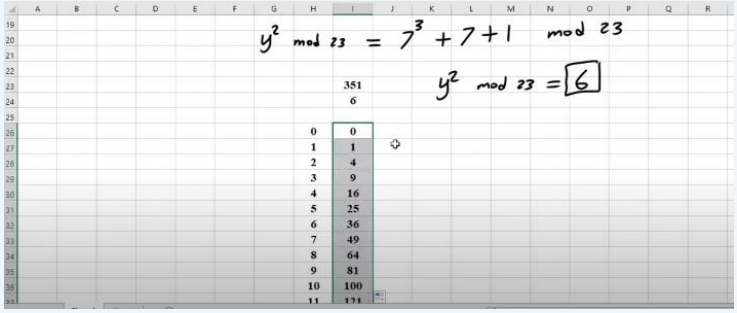
Burada (x_3, y_3) iki katlanmış noktanın koordinatlarıdır.

Sonuç olarak, P noktasının iki katı R noktasını elde etmiş oluruz.

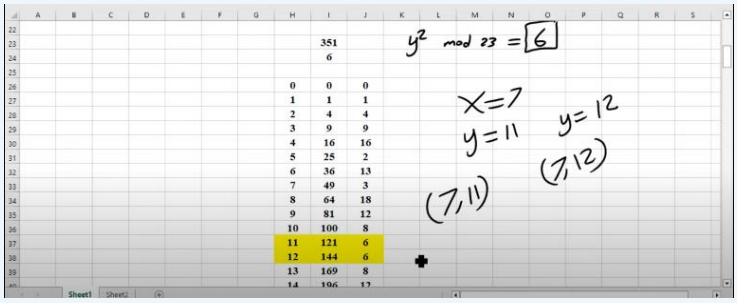
Bu işlem, nokta toplama işleminden farklıdır, çünkü burada aynı noktanın kendisiyle toplama yapılmaz. Noktanın iki katı, eğri üzerindeki matematiksel işlemler kullanılarak hesaplanır.



X=7 seçelim

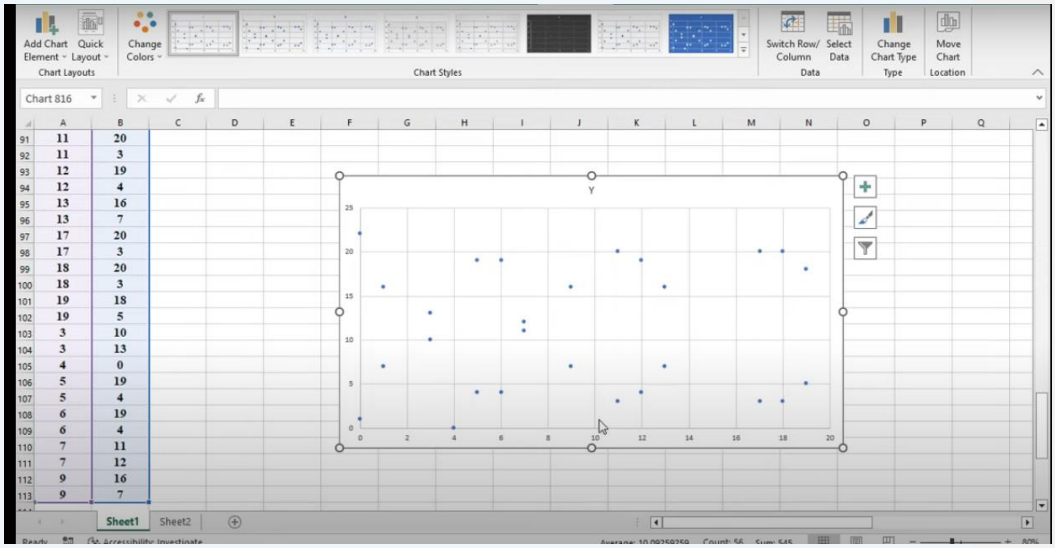


Mod 23 olduğu için y değerleri 22 ye kadar çekiyoruz.



Bu şekilde 1.sütün 1-22 olan değerler oldu 2.sütün o değerlerin kareleri ve 3.sütün o değerlerin modları bizde modu 6 olan değerleri arıyoruz.

Ondan (7,11) ve (7,12) noktaları elliptic curve grafımızın içinde vardır. Tüm x ve y noktaları bulursak grafımız bu şekilde olacaktır:



Burda matematiğin karmaşıklığı bitirelim bizim için önemli olan private key hakında konuşalım ve konuları birbiryle bağlayalım.

Elitik eğri kriptografisi (ECC)'nin karmaşıklığı ve private key (özel anahtar) arasında bir ilişki vardır. ECC, daha güçlü bir şifreleme yöntemi olarak kabul edilir, çünkü diğer şifreleme yöntemlerine kıyasla daha az kaynak kullanımı gerektirirken daha yüksek güvenlik sağlayabilir demiştik. Bu, ECC'nin matematiksel karmaşıklığından kaynaklanır.

Private key, ECC şifrelemesinde kullanılan kritik bir bileşendir. ECC'de, bir kullanıcının herhangi bir işlemi gerçekleştirmek için bir public-private key çiftine ihtiyacı vardır. Public key (genel anahtar), diğer kullanıcılarla paylaşılırken, private key (özel anahtar) sadece sahibi tarafından bilinir ve gizli tutulur.

Private key, ECC'deki matematiksel işlemlerle ilişkilidir. Özel anahtarın güvenliği, ECC'nin matematiksel özelliklerine dayanır. Private key, ECC'de noktaların işlem gördüğü eğri üzerinde matematiksel hesaplamalar gerçekleştirilerek oluşturulur.

Private key'in uzunluğu, ECC'nin güvenliği üzerinde etkili olan bir faktördür. Daha uzun ve karmaşık bir private key, güvenlik düzeyini artırır ve kaba kuvvet saldırılarına karşı daha dayanıklı hale getirir. Bu nedenle, ECC'de kullanılan private key'in yeterli uzunlukta ve rastgele oluşturulmuş olması önemlidir.

ECC'nin matematiksel karmaşıklığı, private key'in güvenliği ve ECC algoritmasının dayanıklılığına katkıda bulunur. ECC, daha az kaynak kullanarak daha güçlü güvenlik sağlayabilen bir şifreleme yöntemi olduğu için, matematiksel karmaşıklığıyla private key'in güvenliğini destekler.

- Brute Force (Exhaustive search)

Brute force, kaba kuvvet saldırısı olarak da bilinen bir şifre kırma yöntemidir. Bu yöntemde, şifrelenmiş verileri çözmek veya bir anahtarın değerini bulmak için tüm olası kombinasyonları denemek amacıyla sistematik bir şekilde tekrarlanan denemeler yapılır.

Kaba kuvvet saldırısı, herhangi bir güvenlik mekanizmasının zayıf noktasını kullanarak, şifreleme algoritmasının veya anahtarın yanlış tahmin edilmesine dayanır. Şifrelenmiş verilerin veya anahtarın doğru değerini bulmak için tüm olası kombinasyonlar denendiğinden, kaba kuvvet saldırıları genellikle zaman alıcı ve kaynak yoğun işlemlerdir.

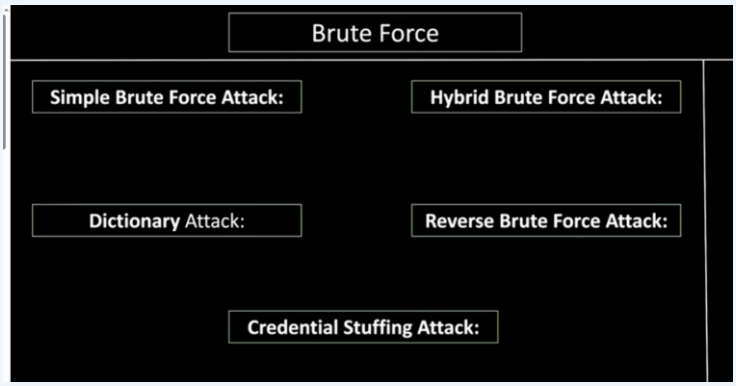
ECC, güçlü bir şifreleme yöntemi olduğu için, kaba kuvvet saldırılarına karşı dirençlidir. ECC'nin güvenliği, eğri üzerindeki noktaların matematiksel özellikleri ve doğru şekilde oluşturulmuş uzun ve rastgele private key'lerle sağlanır. Uzun ve rastgele bir private key kullanıldığında, kaba kuvvet saldırısı için tüm olası anahtar kombinasyonlarının denemesi neredeyse imkansız hale gelir.

Ancak, ECC'nin güvenliği, private key'in uzunluğu ve rastgele oluşturulması gibi faktörlere bağlıdır. Kısa veya tahmin edilebilir bir private key kullanılması durumunda, kaba kuvvet

saldırılarına karşı daha savunmasız olunabilir. Bu nedenle, ECC kullanırken, yeterli güvenlik seviyesini sağlamak için güçlü private key'lerin kullanılması önemlidir.

Kaba kuvvet saldırıları, şifreleme algoritmalarını test etmek, zayıf noktalarını tespit etmek veya bir sistemin güvenliğini değerlendirmek için kullanılabilir. Ancak, yasalara aykırı veya yetkisiz şekilde kullanıldığında, başkalarının gizli bilgilerine erişmek veya şifreli sistemlere izinsiz erişim sağlamak amacıyla kullanılan bir saldırı yöntemi olarak kabul edilir. Bu nedenle, kaba kuvvet saldırıları yasa dışı ve etik dışı olarak değerlendirilir.

Türleri:



Ama yine şifreleme geliştiği gibi şifre kırma teknikleri de gelişti.

Bu tür saldırıların gerçekleştirilmesi genellikle zaman alıcı ve kaynak yoğun olduğundan, yapay zeka (AI) teknikleri bazen brute force saldırılarında kullanılabilir. İşte AI'nın brute force saldırılarıyla ilişkili kullanımı hakkında bazı bilgiler:

- Parola Kırma: Brute force saldırılarının en yaygın kullanım alanlarından biri, kullanıcı hesaplarının parolalarını kırmaktır. AI, brute force saldırılarında kullanılan kombinasyonları tahmin etmek ve analiz etmek için kullanılabilir. Örneğin, AI tabanlı bir model, parolaların genellikle belirli desenlere, yaygın kelimelere veya kullanıcı bilgilerine dayandığını öğrenebilir ve bu bilgileri kullanarak parola tahmininde bulunabilir.
- Şifre Kırma: AI, brute force saldırılarıyla şifrelenmiş verileri kırmak için de kullanılabilir. Örneğin, bir şifreleme algoritması veya anahtarın olası kombinasyonlarını tahmin etmek için yapay sinir ağları veya genetik algoritmalar gibi AI teknikleri kullanılabilir. Bu, brute force saldırılarını daha etkili ve hızlı hale getirebilir.
- Hızlandırma ve Optimizasyon: Brute force saldırıları genellikle çok uzun sürebilir, çünkü tüm olası kombinasyonlar denenir. AI, bu süreci hızlandırmak veya optimize

etmek için kullanılabilir. Örneğin, derin öğrenme teknikleri veya paralel hesaplama, brute force saldırılarını daha verimli hale getirebilir.
- Saldırı Algılama ve Savunma: AI, brute force saldırılarını algılamak ve savunmak için kullanılabilir. AI tabanlı güvenlik sistemleri, normalden farklı davranış kalıplarını tanımlayabilir ve potansiyel brute force saldırılarını tespit edebilir. Bu, sistemlere karşı yapılan saldırıları otomatik olarak engelleyebilir veya saldırının etkisini en aza indirebilir.

Yapay zeka, brute force saldırılarına karşı hem saldırıda bulunanların hem de savunma yapanların işini kolaylaştırabilir. Ancak AI tekniklerinin kullanımı, etik ve yasal sorunları içerebilir. Saldırılar veya savunma amaçlarıyla AI tekniklerinin kullanımı, ilgili yasal düzenlemelere ve etik kurallara uygun olmalıdır.

-Bitcoin public key ---> SHA 256

Bitcoin'deki public key (genel anahtar), SHA-256 (Secure Hash Algorithm 256-bit) algoritmasıyla türetilir. İşlem aşağıdaki adımlarla gerçekleştirilir:

- Öncelikle, kullanıcının Bitcoin cüzdanı tarafından oluşturulan bir çift anahtar vardır: private key (özel anahtar) ve public key (genel anahtar).
- Public key, ECC (elliptic curve cryptography) algoritması kullanılarak eliptik eğri üzerinde hesaplanır. ECC algoritmasında, private key, rastgele bir sayıdır ve public key, bu private key'in eğri üzerindeki noktalarla çarpımıyla hesaplanır.
- Public key, 256 bit uzunluğunda bir dizedir. Ancak bu noktada, public key doğrudan SHA-256 algoritmasına tabi tutulmaz.
- Bitcoin'de public key, özel bir format olan DER (Distinguished Encoding Rules) formatına dönüştürülür. DER formatı, genellikle ASN.1 (Abstract Syntax Notation One) ile kullanılır ve genel anahtarı standart bir yapıya yerleştirir.
- DER formatına dönüştürülen public key, SHA-256 algoritması ile bir hash değeri oluşturmak için işleme tabi tutulur. SHA-256, genellikle Bitcoin'deki blok zinciri işlemlerinin doğrulanması ve bütünlüğünün sağlanması için kullanılan bir hash algoritmasıdır.
- Sonuç olarak, SHA-256 algoritması ile elde edilen hash değeri, genellikle daha kısa bir form olan Base58 veya Base64 gibi bir kodlama formatı kullanılarak adres oluşturma işlemine tabi tutulur. Bu şekilde Bitcoin adresi elde edilir.

Özetle, Bitcoin'deki public key, özel anahtarın eliptik eğri üzerinde çarpımıyla hesaplanır ve ardından SHA-256 algoritmasıyla bir hash değeri oluşturulur. Bu hash değeri, daha sonra Bitcoin adresine dönüştürülerek kullanılır.

Motaz ahmed abdulateef(Rawi) tarafından yazıldı 10.07.2023

Kaynaklar:

- "Elliptic Curve Cryptography Step by Step in Arabic" - محاضرات التشفير بالعربي(https://youtu.be/-Ie7Xpx3pIQ)
- "خوارزمية Diffie-Hellman بالعربي"(https://youtu.be/KOKmY0TtYww)
- "Elliptic Curves - Computerphile" (https://youtu.be/tnFZreeThZg)
- (https://youtu.be/kr5_FVsWnus)

Yazımsal ve mantıksal hatalar (open ai-chat gpt v-3.5) tarafından düzeltildi.