T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

SCHOOF ALGORİTMASININ BAZI UYGULAMALARI

YÜKSEK LİSANS TEZİ

ÖZGE ÇELİK

BALIKESİR, HAZİRAN - 2012

T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

SCHOOF ALGORİTMASININ BAZI UYGULAMALARI

YÜKSEK LİSANS TEZİ

ÖZGE ÇELİK

BALIKESİR, HAZİRAN - 2012

KABUL VE ONAY SAYFASI

ÖZGE ÇELİK tarafından hazırlanan " **SCHOOF ALGORİTMASININ BAZI UYGULAMALARI**" adlı tez çalışmasının savunma sınavı 11.06.2012 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Danışman Doç. Dr. Sebahattin İKİKARDEŞ	.....
Üye Doç. Dr. Fırat ATEŞ	.....
Üye Doç. Dr. Özden KORUOĞLU	.....

Jüri üyeleri tarafından kabul edilmiş olan bu tez BAÜ Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Hilmi NAMLI .....

ÖZET

SCHOOF ALGORİTMASININ BAZI UYGULAMALARI
YÜKSEK LİSANS TEZİ
ÖZGE ÇELİK
BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

(TEZ DANIŞMANI: DOÇ. DR. SEBAHATTİN İKİKARDEŞ)

BALIKESİR, HAZİRAN - 2012

ABD hükümetinin kriptolojik istihbarat ajansı, Ulusal Güvenlik Ajansı'nın tavsiye ettiği gibi eliptik eğri kriptografisinin kullanımı giderek artmaktadır. Kriptografide eliptik eğrilerin kullanımının başlıca avantajları; zengin bir matematiksel yapıya sahip olması, üzerinde yaygın olarak çalışılmış ve özellikle benzer kriptografik sistemlerin sağladığı güvenliği daha küçük anahtar boyutlarıyla sağlamasıdır. Bu çalışma süresinin benzer algoritmalardan daha kısa olduğunu göstermektedir.

Bu tezde, sonlu cisimler üzerinde eliptik eğrinin rasyonel noktalarının sayısını bulmaya yarayan ilk deterministik polinom zamanlı algoritma, Schoof Algoritması sunulmuştur. Algoritma, eliptik eğri kriptografisinde bazı uygulamalara sahiptir. Öyle ki, bir eliptik eğrinin üzerindeki noktaların grubunda Ayrık Logaritma Problemi'ni çözmenin zorluğunu sınamak için noktaların sayısını bilmek önemlidir.

Bu algoritma, uygulamalar için MAGMA hesaplama programı kullanılarak, temel kavramları, sonuçları ve ispatı olmak üzere tanıtıldı. Bu tez boyunca eliptik eğrilerle ilgili çeşitli aritmetik Python uygulamaları da verilmiştir.

ANAHTAR KELİMELER: sonlu cisimler üzerinde eliptik eğriler, rasyonel noktalar, Schoof algoritması, kriptografi

ABSTRACT

SOME IMPLEMENTATION OF SCHOOF'S ALGORITHM
MSC THESIS
ÖZGE ÇELİK
BALIKESIR UNIVERSITY INSTITUTE OF SCIENCE
MATHEMATICS

(SUPERVISOR: ASSOC. PROF. DR. SEBAHATTİN İKİKARDEŞ)

BALIKESİR, JUNE 2012

Elliptic curve cryptography plays an increasing role, as the National Security Agency, the cryptologic intelligence agency of the United States government, has recommended its use. The main advantages of the use of elliptic curves in cryptography rely on the fact that these have a rich mathematical structure, widely studied, and specially on the fact that, with quite smaller keys, they provide the same security level as other cryptographic systems. This substantially reduces the running time of the corresponding algorithms.

In this dissertation, the first deterministic polynomial time algorithm to find number of rational points on a given elliptic curve over finite fields, Schoof's Algorithm, is presented. The algorithm has applications in elliptic curve cryptography where it is important to know the number of points to judge the difficulty of solving the discrete logarithm problem in the group of points on an elliptic curve.

The basic notions and results of that algorithm are introduced, including a proof, using the computation program MAGMA to implement algorithm. Throughout this thesis, several implementations in Python are given, including an implementation of the arithmetic of elliptic curves, an implementation of Schoof's algorithm.

KEYWORDS: elliptic curves over finite fields, rational points, Schoof's algorithm, cryptography

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

FONKSİYON LİSTESİ

Sayfa

SEMBOL LİSTESİ

Simge	Adı
$\mathbb{Z}$	:Tam sayılar kümesi
$\mathbb{Z}_n$	:n modunda kalan sınıflarının kümesi
$\mathbb{Z}_n^{\times}$	:n modunda kalan sınıflarının kümesinin çarpımsal grubu
$\boldsymbol{\phi(n)}$	:Euler- ϕ fonksiyonu
$\boldsymbol{\#G}$	:Bir G grubunun eleman sayısı
$\boldsymbol{G^r}$	:Bir G grubunun r kez direkt toplamı
$\oplus$	:Eliptik eğriler üzerinde tanımlı toplama işlemi
$\mathbb{C}$	:Karmaşık sayılar kümesi
$\mathbb{R}$	:Reel sayılar kümesi
$\boldsymbol{Ker\ \psi}$	: ψ dönüşümünün çekirdeği
$\boldsymbol{K}$	:Cisim
$\mathbb{Q}$	:Rasyonel sayılar kümesi
$\boldsymbol{\bar{K}}$	:K cisminin cebirsel kapanışı
$\mathbb{F}_p$	:p elemanlı sonlu cisim
$\mathbb{F}_q$	:Karakteristiği p olan q elemanlı sonlu cisim
$\boldsymbol{\phi_q}$	:Frobenius endomorfizması
$\boldsymbol{C}$	:Cebirsel eğri
$\boldsymbol{\mathcal{O}_C}$	:C cebirsel eğrisinin infinity noktası
$\boldsymbol{E}$	:Eliptik eğri
$\boldsymbol{\mathcal{O}}$	:E eliptik eğrisinin infinity noktası
$\boldsymbol{E/K}$	:Katsayıları K cisminde bulunan E eliptik eğrisi
$\boldsymbol{E(L)}$	:L cismindeki E eliptik eğrisinin noktalarının kümesi
$\boldsymbol{E[n]}$	:E eliptik eğrisi üzerindeki n. mertebeden noktaların kümesi
$\boldsymbol{E_{tors}}$	:E eliptik eğrisinin torsiyon alt grubu
$\boldsymbol{\psi_m}$	:m. bölüm polinomu
$\boldsymbol{\#E}(\mathbb{F}_q)$	:$\mathbb{F}_q$ sonlu cisminde E eliptik eğrisi üzerindeki noktaların sayısı
$\boldsymbol{t}$	:Frobenius izi
$\left(\frac{x}{p}\right)$	:Legendre sembolü

$\boldsymbol{O}(\ldots)$:Büyük- O notasyonu

$\boldsymbol{boy}(\boldsymbol{n})$:İkilik sistemde yazılı bir sayının basamak sayısı(uzunluğu)

$\boldsymbol{Time}(\boldsymbol{n})$:n algoritmasının çalışma süresi

$\mathbf{gcd}\ (\boldsymbol{a}, \boldsymbol{b})$:a ve b 'nin en büyük ortak böleni

$\boldsymbol{pay}(\boldsymbol{a})$:a'nın payı

$[\boldsymbol{n}]\boldsymbol{P}$:P noktasının n sayısı kadar yan yana toplanması

ÖNSÖZ

İki senelik yüksek lisans eğitimim boyunca, her zaman bana sabırla bir şeyler öğretmeye çalışan, beni farklı yenilikler denememem için destekleyen, olaylara her zaman gerçekçi yaklaşmamı sağlayan, uzaktan da olsa hep yanımda olduğunu hissettiğim, değerli danışman hocam Doç. Dr. Sebahattin İKİKARDEŞ'e,

Leuven'de bulunduğum zaman zarfında benim ikinci danışmanım gibi olan sevgili Doç. Dr. Wouter CASTRYCK'e,

değerlerini her seferinde biraz daha anladığım canım annem ve babama,

hayat neşemin kaynağı kardeşime,

içten teşekkürler…

1. GİRİŞ

Uygarlığın başlangıcından bu yana, bir bilgiyi gizli bir şekilde sunmak için çeşitli şifreleme yolları bulmak, alıcıdan başkasının bu bilgiyi anlayamaması için ise kolay hesaplanamayacak şifre çözücüler oluşturmak, her zaman çözülmesi gereken bir problem olmuştur. Öncelikle, askeri ve politik alanda artan bu ihtiyaç, kriptografinin günümüze kadar gelişerek yaygınlaşmasını sağlamıştır. Özellikle, teknolojinin gelişmesi ve yaşamımızın içine bu denli girmesiyle kriptografi, her gün farkında olmadan kullandığımız bir gereklilik haline gelmiştir.

Kriptografide güvenilirliği sağlamak için kullanılan matematiksel problemden biri de ayrık logaritma problemidir. Bu problem, "G bir grup olsun. Bu gruptan alınacak a ve b elemanları için $a^k = b$ olacak şekilde bir k bulunabilir mi?" şeklindedir. Yani şifrelemek istenilen bilgi a, şifreleme işlemi a^k ve karşı tarafın eline geçen şifrelenmiş bilgi ise b'dir. Fark edildiği gibi eğer G grubu ne kadar çok elemana sahip olursa, eşitliği sağlayan değerleri, deneyerek bulmaya çalışmak o kadar çok zaman alır. Yani güvenilir hale gelir. Ama geniş bir G grubuna sahip olmak, güvenilirlik adına avantaj sağladığı kadar, bilgisayarda kaplayacağı alanın da aynı oranda artması onu bir o kadar dezavantajlı hale getirir. İşte kriptografideki bu büyük ikilemle birlikte eliptik eğriler kullanılmaya başlar ve eliptik eğri kriptografisi doğar. Geriye kalan tek problem ise üzerindeki noktaların kümesi kullanılacak doğru eliptik eğriyi bulmaktır.

1985'te Rene Schoof $\mathbb{F}_q$ sonlu cisimleri üzerinde tanımlanmış eliptik eğrilerin noktalarının sayısını bulmaya yarayan, polinom zamanlı çalışan, bir algoritma yayınladı. Öyle ki bu algoritma çok büyük q asalları için bile var olan algoritmalardan çok daha hızlı çalışıyordu. Özellikle, o günlerde kullanılan diğer algoritmalardan biri olan Baby Step-Giant Step metoduyla karşılaştıracak olursak Baby Step- Giant Step metodunda $q^{1/4}$ bit işlerken, Schoof Algoritması için sadece $\log^8 q$ bit gerekiyordu.

Bu tezde, Schoof algoritması matematiksel olarak incelenmiş, cebir hesaplama programı olan MAGMA hesaplama programı ile algoritma baştan programlanmış ve sayısal uygulamalarla algoritma somutlaştırılmıştır.

İkinci bölümde, algoritmanın çalışabilmesi için gerekli olan matematiksel bilgi altyapısı oluşturulmaya çalışılmıştır.

Üçüncü bölümde, torsiyon noktaları ve bölüm polinomları tanımlanmıştır.

Dördüncü bölümde, sonlu cisimler üzerindeki eliptik eğrilerin nokta sayıları için temel yapıtaşı olan Hasse teoremi ve Frobenius endomorfizması verilmiştir.

Beşinci bölümde, sonlu cisimler üzerindeki eliptik eğrilerin nokta sayısını hesaplamak için kullanılan yöntemlerden biri verilmitir.

Altıncı bölümde, bir algoritmanın çalışma zamanı ve bu tahmini zamanın nasıl belirlendiği anlatılmıştır.

Yedinci bölümde, Schoof algoritması, matematiksel olarak tanıtılmış, MAGMA hesaplama programı kullanılarak, sayısal bir uygulamayla somutlaştırılmış ve tahmini çalışma süresi incelenmiştir.

Sekizinci bölüm olan sonuçlar bölümünde ise, 1985'te yayınlanan bu algoritmanın daha sonra nasıl geliştirildiğinden bahsedilmiş ve bir takım çıkarımlarda bulunulmuştur.

2. ÖNBİLGİLER

2.1 Sayılar Teorisi

Bu kısımda verilen bilgiler [1] nolu kaynaktan derlenmiştir. *n* pozitif bir tamsayı ve $\mathbb{Z}$ nin *n* modülüne göre kalan sınıflarının kümesi

$$\mathbb{Z}_n = \{\bar{1}, \bar{2}, \dots, \overline{n-1}\}$$

olsun. Bu küme toplama işlemine göre bir gruptur.

$$\mathbb{Z}_n{}^{\times} = \{\bar{a} |\ 1 \leq a \leq n,\ \gcd(a,n) = 1\}$$

olsun. $\mathbb{Z}_n{}^{\times}$, çarpma işlemine göre bir gruptur. Bütünlüğü bozmadan kolaylık olması için bundan sonra $a = \bar{a}$ kullanacağız.

$\bar{a} \in \mathbb{Z}_n{}^{\times}$ alalım. *a'nın mertebesi,* $a^k \equiv 1 (mod\ n)$ şeklindeki en küçük $k > 0$ tamsayısıdır. *a*'nın mertebesi, Euler-ϕ fonksiyonu olan $\phi(n)$ 'i böler.

p bir asal sayı ve $a \in \mathbb{Z}_p{}^{\times}$ olsun. *a* nın mertebesi, $p-1$ i böler. Eğer *g* nin mertebesi $p-1$ e eşit ise *g ye mod p* de *ilkel kök* denir. Öyle ki. Eğer *g mod p* bir ilkel kök ise, o halde her bir tamsayının *mod p* eşleneği 0 veya *g*'dir. Örneğin, 3 *mod* 7 'de bir ilkel köktür ve

$$\{1,3,9,27,81,243\} \equiv \{1,3,2,6,4,5\}\ (mod\ 7)$$

olur. Burada *mod p* de ilkel köklerin sayısı $\phi(p-1)$ dir. Özellikle *mod p* de ilkel kök her zaman vardır, bu yüzden $\mathbb{Z}_p{}^{\times}$ bir devirli gruptur.

g'nin *mod p* 'de bir ilkel kök olup olmadığını bulabilmek için basit bir kriter vardır. $p-1$'in faktörizasyonunu bildiğimizi varsayarsak: eğer her bir $q|p-1$ asal sayısı için $g^{(p-1)/q} \not\equiv 1\ (mod\ p)$ ise *g mod p* de bir ilkel köktür.

Şimdi, sayı teori için çok kullanışlı olan bir teorem verelim.

2.1.1 Teorem(Çin Kalan Teoremi):

$n_1, n_2, \dots, n_r$ pozitif tamsayılar olsun. Öyle ki $i \neq j$ olduğunda $\gcd(n_i, n_j) = 1$ ve $a_1, a_2, \dots, a_r$ birer tamsayı olsun. Burada her *i* için

$$x \equiv a_i\ (mod\ n_i)$$

olacak şekilde bir *x* vardır. Bu *x* tamsayısı tek bir şekilde $mod\ n_1 n_2 \dots n_r$ ile saptanır.

Örnek olarak, $n_1 = 4, n_2 = 3, n_3 = 5$ ve $a_1 = 1,\ a_2 = 2,\ a_3 = 3$ olsun. Bu durumda $x = 53$ sonucu için

$$x \equiv 1\ (mod\ 4), \qquad x \equiv 2\ (mod\ 3), \qquad x \equiv 3\ (mod\ 5)$$

sağlanır ve *x* sonucu için $x \equiv 53 (mod\ 60)$ 'dir.

Çinlilerin Kalan Teoremi'ni ifade etmenin başka bir yolu ise, eğer $i \neq j$ için $\gcd(n_i, n_j) = 1$ ise,

$$\mathbb{Z}_{n_1, n_2, \dots, n_r} \simeq \mathbb{Z}_{n_1} \oplus \dots \oplus \mathbb{Z}_{n_r}$$

olur. Bu toplam gruplarının bir izomorfizmasıdır. Aynı zamanda halkaların da bir izomorfizmasıdır.

2.2 Grup Teori

Bu tezde kullanılan gruplar toplamsal değişmeli gruplardır. Bunun için gruplarda toplama işlemi kullanılacaktır. Yani bir G grubunun işlemi + ve bu işleme göre etkisiz elemanı *0*'dır. Her $g \in G$ için,

$$0 + g = g + 0 = g$$

ve her bir $g \in G$ nin toplama işlemine göre tersi $-g$ için ,

$$(-g) + g = g + (-g) = 0$$

sağlanır. Bir *n* pozitif tamsayısı için

$$ng = g + g + \cdots + g \quad (n\ kez\ toplam)$$

ve bir $n < 0$ için ise

$$ng = -(|n|g) = -(g + g + \cdots + g) \quad (n\ kez\ toplam)$$

olur.

Daha önce de söylediğimiz gibi bu tezde kullanılan gruplar değişmeli gruplardır. Yani her $g, h \in G$ için

$$g + h = h + g$$

olur.

Eğer G bir sonlu grup ise, G *'nin mertebesi,* sahip olduğu eleman sayısıdır. Bir $g \in G$ elemanının *mertebesi* ise

$$kg = 0$$

eşitliğini sağlayan sıfırdan büyük en küçük tamsayıdır. g elemanının mertebesi k olsun, o halde

$$ig = jg \Leftrightarrow i \equiv j\ (mod\ k)$$

olur.

Sıradaki teorem mertebeler konusunda önemli bir sonuçtur.

2.2.1 Teorem(Lagrange Teoremi):

G bir sonlu grup olsun.

- *H* ,*G*'nin bir alt grubu olsun. O halde *H*'nin mertebesi, *G*'nin mertebesini böler.
- $g \in G$ olsun. *g* elemanının mertebesi, *G*'nin mertebesini böler.

$\#G/\#H$ oranına *H'nin G'deki indeksi* denilir. Genelleyecek olursak, *G* grubunun bir *H* alt grubunun indeksi, $g_i \in G$ elemanları ile *G*'nin

$$G = \bigcup_{i=1}^{n} (g_i + H)$$

olacak şekilde tek bir şekilde yazılabilmesini sağlayan en küçük *n* tamsayısıdır. Örneğin,

$$\mathbb{Z} = (0 + 3\mathbb{Z}) \cup (1 + 3\mathbb{Z}) \cup (2 + 3\mathbb{Z})$$

olduğundan $3\mathbb{Z}$'nin $\mathbb{Z}$'deki indeksi 3'tür.

Devirli grup, $\mathbb{Z}$'ye veya herhangi bir *n* için $\mathbb{Z}_n$'ye izomorf olan gruptur. Bu iki grup tek bir eleman tarafından üretilebilme özelliğine sahiptir. Mesela, $\mathbb{Z}_4$, 1 tarafından üretilir. Hatta 3 tarafından da $\{0, 3, 3+3, 3+3+3\}$ şeklinde üretilir.

Sıradaki teorem Lagrange teoreminin tersinin sonlu devirli gruplar için geçerli olduğunu söyler.

2.2.2 Teorem:

G, mertebesi *n* olan sonlu devirli bir grup olsun. $d > 0$ ve *d, n*'yi bölüyor olsun.

- *G,* mertebesi *d* olan tek bir alt gruba sahiptir.
- *G,* mertebesi *d* 'yi bölen *d* tane elemana sahiptir ve *G,* mertebesi *d* olan $\phi(d)$ tane elemana sahiptir (Burada $\phi(d)$ Euler-ϕ fonksiyonudur.).

Örneğin, $\mathbb{Z}_6$, mertebesi 3 olan $\{0, 2, 4\}$ alt grubuna sahiptir. $2, 4 \in \mathbb{Z}_6$ elemanlarının mertebeleri 3'tür.

G_1 ve G_2 gibi iki grubun *direkt toplamı*,

$$G_1 \oplus G_2 = \{(g_1, g_2) \mid g_1 \in G_1, g_2 \in G_2\}$$

olacak şekilde G_1 ve G_2 gruplarının elemanlarının oluşturduğu sıralı ikililerin bir kümesi olarak tanımlanır. Sıralı ikililer,

$$(g_1, g_2) + (h_1, h_2) = (g_1 + h_1, g_2 + h_2)$$

şeklinde toplanabilirler. Bu da, $G_1 \oplus G_2$yi etkisiz elemanı (0,0) olan bir grup yapar. İkiden daha fazla sayıdaki grubun direkt toplamını belirtmek için benzer şekilde bir *G* grubunun *r* kez direkt toplamını G^r ile gösteriyoruz. Özel olarak, toplama işlemi altında grup olan $\mathbb{Z}^r$, tamsayıların *r*-demetlerinin bir kümesini belirtir.

Şimdi bazı önemli temel teoremleri verelim.

G_1 ve G_2 gruplarını ele alalım. Eğer her $g, h \in G_1$ için

$$\psi: G_1 \mapsto G_2$$

$$\psi(gh) = \psi(g)\psi(h)$$

olacak şekilde birebir ve örten bir dönüşüm var ise bu durumda G_1 *ve* G_2 *izomortur* denir. (Unutmamak gerekir ki burada gh çarpımı G_1'in elemanı ve $\psi(g)\psi(h)$, G_2'nin elemanıdır.)

2.2.3 Teorem:

Bir sonlu değişmeli grup, $i = 1, 2, \dots, s-1$ için $n_i | n_{i+1}$ olacak şekilde

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \dots \oplus \mathbb{Z}_{n_s}$$

formundaki bir gruba izomorfiktir. Burada n_i tamsayıları *G* tarafından tek olarak belirlenirler, [1].

Eğer *G*'nin her bir elemanı, $m_i \in \mathbb{Z}$ için

$$m_1 g_1 + \cdots + m_k g_k$$

formunda $\{g_1, g_2, \dots, g_k\}$ sonlu kümesi kullanılarak yazılabiliyor ise *G* değişmeli grubuna *sonlu üreteçli* denilir.

2.2.4 Teorem:

Bir sonlu üreteçli değişmeli grup, $i = 1, 2, \dots, s-1$ için $n_i | n_{i+1}$ ve $r \geq 0$ olacak şekilde,

$$\mathbb{Z}^r \oplus \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \dots \oplus \mathbb{Z}_{n_s}$$

formunda bir gruba izomorftur. Burada n_i ve *r* tamsayıları *G* tarafından tek olarak belirlenirler, [1].

*G'*nin

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \ldots \oplus \mathbb{Z}_{n_s}$$

ile izomorf olan alt grubuna *G'nin torsiyon alt grubu* denilir. *r* tamsayısına ise *G'nin rankı* denilir.

2.2.5 Teorem:

$G_1 \subseteq G_2 \subseteq G_3$ gruplarını alalım ve varsayalım ki herhangi bir *r* tamsayısı için hem G_1 hem de G_3, $\mathbb{Z}^r$'ye izomorf olsun. O halde G_2 de $\mathbb{Z}^r$'ye izomorftur, [1].

Örneğin, $G_1 = 12\mathbb{Z}$, $G_2 = 6\mathbb{Z}$ ve $G_3 = \mathbb{Z}$,yani her biri $\mathbb{Z}$'ye izomorf gruplar olsun. G_1 ve G_3 ,$\mathbb{C}$ üzerinde birer latistir. O halde G_1 ve G_3, $\mathbb{Z}^2$'ye izomorftur. Eğer $G_1 \subseteq G_2 \subseteq G_3$ ise, $G_2 \simeq \mathbb{Z}^2$ olur. Yani $G_2 = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ olacak şekilde ω_1 ve ω_2 vardır. G_1 bir latis olduğundan, $\mathbb{R}$ üzerinde lineer bağımsız iki vektörü bulunur. $G_1 \subseteq G_2$ olduğundan, ω_1 ve ω_2 , $\mathbb{R}$ üzerinde lineer bağımsız iki vektördür. Buradan G_2 bir latis elde edilir, [1].

G_1 ve G_2 iki grup olsun. Eğer her $g, h \in G_1$ için

$$\psi: G_1 \mapsto G_2$$

$$\psi(g + h) = \psi(g) + \psi(h)$$

olacak şekilde bir fonksiyon bulunuyorsa buna G_1 den G_2 ye bir *homomorfizma* adı verilir. Diğer bir değişle, ψ, G_1'deki bir toplamı G_2'deki toplam karşılığına götüren bir fonksiyondur. ψ'nin *çekirdeği*

$$Ker\ \psi = \{g \in G_1 \mid \psi(g) = 0\}$$

olur.

2.2.6 Teorem:

G_1 , bir sonlu grup ve $\psi: G_1 \mapsto G_2$ bir homomorfizma olsun. O halde

$$\#G_1 = (\#Ker\ \psi)(\#\psi(G_1))$$

olur, [1].

2.3 Cisim Teori

Bu alt bölümde verilen bilgiler [1] nolu kaynaktan derlenerek verilmiştir. *K* bir cisim olsun. $1 \in \mathbb{Z}$ yi $1 \in K$ ya götüren bir

$$\psi: \mathbb{Z} \mapsto K$$

halka homomorfizması vardır. Eğer ψ birebir ise, o halde *K* 'nın *karakteristiği 0*'dır denilir. Aksi halde, $\psi(p) = 0$ olacak şekilde bir en küçük *p* pozitif tamsayısı vardır. Bu durumda da, *K* nın *karakteristiği p* dir denilir. Eğer *p,* $1 < a \leq b < p$ olacak şekilde *ab* ye eşit ise, buradan

$$\psi(a)\psi(b) = \psi(p) = 0$$

olacağından ya $\psi(a) = 0$ ya da $\psi(b) = 0$ dır. *p* nin en küçük olması istendiğinden *p* bir asal sayı bulunur.

K nın karakteristiği *0* olduğunda, bu rasyonel sayılar cismi $\mathbb{Q}$ nun *K* da olduğunu, *K* nın karakteristiği *p* olduğunda ise $\mathbb{F}_p$nin *K* da olduğunu gösterir.

$K \subseteq L$ olacak şekilde *K* ve *L* cisimlerini alalım. Eğer $a_0, a_1, \ldots, a_{n-1} \in K$ için

$$f(X) = X^n + a_{n-1}X^{n-1} + \cdots + a_0$$

sabitten farklı bir polinom varsa ve $\alpha \in L$ için $f(\alpha) = 0$ oluyor ise α , *K üzerinde cebirseldir* denilir. Eğer *L*'nin her bir elemanı *K* üzerinde cebirsel ise bu duruma *L, K üzerinde cebirseldir* veya *L, K'nın cebirsel genişlemesidir* denilir. *K* cisminin *cebirsel kapanışı* $\overline{K}$ cismidir ve şunları sağlar;

- $\overline{K}$, *K* üzerinde cebirseldir.
- Her katsayıları $\overline{K}$'de sabitten farklı $g(X)$ polinomu yine $\overline{K}$'de köklere sahiptir.(Bu da demek oluyor ki $\overline{K}$ cebirsel olarak kapalıdır.)

Eğer $g(X)$'in derecesi *n* ve $\alpha \in \overline{K}$ onun bir kökü ise, $n-1$ dereceli $g_1(X)$ polinomu için

$$g(X) = (X - \alpha)g_1(X)$$

yazılabilir. Tümevarım kullanılarak, $g(X)$'in $\overline{K}$'de tam olarak *n* tane kökü bulunur.

Her *K* cismi için bir cebirsel kapanış olduğu ve *K*'nın herhangi iki cebirsel kapanışının birbirine izomorf olduğu açıktır.

$K = \mathbb{Q}$ olduğunda, kompleks sayıların kümesi olan $\overline{\mathbb{Q}}$ cebirsel kapanışı, $\mathbb{Q}$ üzerinde cebirseldir. $K = \mathbb{C}$ olduğunda, $\overline{\mathbb{C}}$ cebirsel kapanışı yine kendisidir. Yani $\mathbb{C}$ cebirsel olarak kapalıdır.

2.3.1 Sonlu Cisimler

p bir asal sayı olsun. $\mathbb{F}_p$, *p* elemanlı bir *mod p* tamsayılar cismini belirtmektedir. Bir sonlu cismin eleman sayısı, bir asal sayının kuvvetidir ve *p* 'nin her bir p^n kuvveti için p^n elemanlı tek bir cisim vardır. (*Not:* $n \geq 2$ *için p'nin çarpımsal tersi olmadığından, ve hatta* $p.p^{n-1} \equiv 0 \ (mod\ p^n)$ *yani p bir sıfır bölen olduğundan* $\mathbb{Z}_{p^n}$ *bir cisim belirtmez.*) Bu tezde p^n elemanlı bir cisim $\mathbb{F}_{p^n}$ ile gösterilmektedir. Bunun için literatürde geçen başka bir gösterim şekli ise $GF(p^n)$'dir.

$$\mathbb{F}_{p^m} \subseteq \mathbb{F}_{p^n} \iff m|n$$

olur ve $\mathbb{F}_p$'nin cebirsel kapanışı

$$\overline{\mathbb{F}}_{\mathrm{p}} = \bigcup_{n \geq 1} \mathbb{F}_{p^n}$$

eşitliğiyle verilir.

2.3.2 Teorem:

$\mathbb{F}_p$'nin cebirsel kapanışı, $\overline{\mathbb{F}}_{\mathrm{p}}$ve $n \geq 1$ pozitif tamsayısı için $q = p^n$ olsun. O halde

$$\mathbb{F}_q = \{\alpha \in \overline{\mathbb{F}}_{\mathrm{p}} \,|\alpha^q = \alpha\}$$

dur, [1].

İspat: $\mathbb{F}_q{}^{\times}$, $\mathbb{F}_q$'nin $q-1$ elemanlı, sıfırdan farlı elemanlarının bir grubunu göstersin. Bu durumda $0 \neq \alpha \in \mathbb{F}_q$ için $\alpha^{q-1} = 1$'dir. Buradan her $\alpha \in \mathbb{F}_q$ için $\alpha^q = \alpha$ olur.

$$\frac{d}{dX}(X^q - X) = qX^{q-1} - 1 = -1$$

olduğundan, $X^q - X$ polinomunun katlı kökü bulunmaz. Bu nedenle, $\alpha^q = \alpha$ olacak şekilde q 'dan farklı $\alpha \in \overline{\mathbb{F}}_{\mathrm{p}}$ vardır.

Teoremde geçen iki küme de q elemana sahip ve biri diğerini içerdiğinden bu kümeler eşittir. Bu da ispatı tamamlar.

Her $x \in \overline{\mathbb{F}}_{\mathrm{q}}$ için $\overline{\mathbb{F}}_{\mathrm{q}}$'nun *q-uncu Frobenius otomorfizması* ϕ_q'yı

$$\phi_q(x) = x^q$$

şeklinde tanımlayalım.

2.3.3 Önerme:

q, bir p asal sayısının kuvveti olsun.

1. $\overline{\mathbb{F}}_q = \overline{\mathbb{F}}_p$.
2. ϕ_q, $\overline{\mathbb{F}}_q$ nun bir otomorfizmasıdır. Özel olarak, her $x, y \in \overline{\mathbb{F}}_q$ için

$$\phi_q(x+y) = \phi_q(x) + \phi_q(y)$$

ve

$$\phi_q(xy) = \phi_q(x)\phi_q(y)$$

sağlanır.

3. $\alpha \in \overline{\mathbb{F}}_q$ olsun. O halde

$$\alpha \in \mathbb{F}_{q^n} \iff {\phi_q}^n(\alpha) = \alpha$$

olur, [1].

Daha önce, Bölüm 2.1'de değindiğimiz gibi ${\mathbb{F}_p}^\times = {\mathbb{Z}_p}^\times$, bir ilkel kökle üretilmiş devirli bir gruptur. Daha da genelleştirecek olursak, ${\mathbb{F}_q}^\times$ bir devirli gruptur. Şimdi çok kullanışlı bir sonuç ve ispatına bakalım.

2.3.4 Önerme:

$p \nmid m$ olacak şekilde bir *m* pozitif tamsayısı alalım. μ_m, birimsellerinin *m*-inci köklerinin bir grubu olsun. O halde

$$\mu_m \subseteq {\mathbb{F}_q}^\times \Leftrightarrow m | q-1$$

dir, [1].

İspat: Bölüm 2.2'de verdiğimiz Lagrange teoreminden, eğer $\mu_m \subseteq \mathbb{F}_q{}^{\times}$ ise, $m|q-1$ 'dir. Tersine, $m|q-1$ olsun. $\mathbb{F}_q{}^{\times}$, mertebesi $q-1$ olan bir devirli grup olduğundan mertebesi m olan bir alt grubu vardır. Lagrange teoreminden, bu alt grubun elemanları $x^m = 1$ eşitliğini sağlar. Bu μ_m'in m elemanı olmalıdır.

2.4 Eliptik Eğriler

Tarihte ilk olarak Yunanlı matematikçiler tarafından ilgilenildiği bilinen, eliptik eğriler, aslında birer kübik denklemlerdir. Bu denklemlere eliptik eğri denilmesinin tek sebebi ise eski zamanlarda elipslerin çevrelerini ve gezegen yörüngelerinin uzunluğunu hesaplamakta kullanılmış olmasıdır. Diophantus, Fermat, Gauss gibi birçok ünlü matematikçiye de ilginç denklem özellikleri nedeniyle çalışmaya elverişli, yeni bir konu olmuştur. Tezin bu kısmında sonlu cisimler üzerindeki kübik eşitliklere bakacağız.

2.4.1 Tanım:

$\mathbb{F}_p$, tamsayılar cismi olmak üzere, $x, y \in \mathbb{F}_p$ olacak şekilde

$$C : F(x,y) = \mathcal{O}_C$$

polinom eşitliklerini sağlayan *(x,y)* ikililerini ele alalım. Genelleştirecek olursak, $\mathbb{F}_q$, $\mathbb{F}_p$ nin $q = p^e$ elemanını bulunduran genişletilmiş cismi için

$$C : F(x,y) = \mathcal{O}_C$$

polinom eşitliklerini sağlayan $x, y \in \mathbb{F}_q$ elemanlarını ele alalım. *(x,y)* sonucu *C* eğrisi üzerinde bir noktayı temsil eder. Eğer sonucun *x* koordinatı ve *y* koordinatı $\mathbb{F}_p$ içinde ise bu sonuca *rasyonel nokta* denir, [2].

2.4.2 Tanım:

E, cinsi 1 olan, singüler olmayan(kökleri ayrık olan) bir eğri ve $\mathcal{O} \in E$ olmak üzere $(E, \mathcal{O})$ ikilisine bir *eliptik eğri* denir. (Genellikle bir eliptik eğri *E* ile gösterilir. Buradan $\mathcal{O}$ elemanının her zaman var olduğu kabul edilir.) Bir *E* eliptik eğrisi, *K* cismi üzerinde tanımlandıysa bu E/K şeklinde yazılır, [3].

2.4.3 Tanım:

Bu tezde, *E* eliptik eğrisi, *A* ve *B* değişkenler olmak üzere,

$$y^2 = x^3 + Ax + B$$

formundaki bir eşitliğin grafiğidir. Bu forma *Weierstrass Eşitliği* denilir. *A,B,x* ve *y*'nin ait olacağı kümeyi belirlemek gerekmektedir. Genellikle bunlar bir *K* cisminin elemanları olarak alınır. Eğer *K, A* ve *B* sabitlerini ihtiva eden bir cisim ise *E eliptik eğrisi, K üzerinde tanımlanmıştır* denir, [1].

L ve *K* bir cisim ve $L \supseteq K$ olsun. Katsayıları *L* cisminde bulunan *E* eliptik eğrisinin noktalarının kümesi $E(L)$ ile gösterilir ve

$$E(L) = \{\mathcal{O}\} \cup \{(x, y) \in L \times L \mid y^2 = x^3 + Ax + B\}$$

Şeklindedir. Ayrıca $E(L)$ kümesi $\mathcal{O}$ noktasını her zaman ihtiva eder, [1].

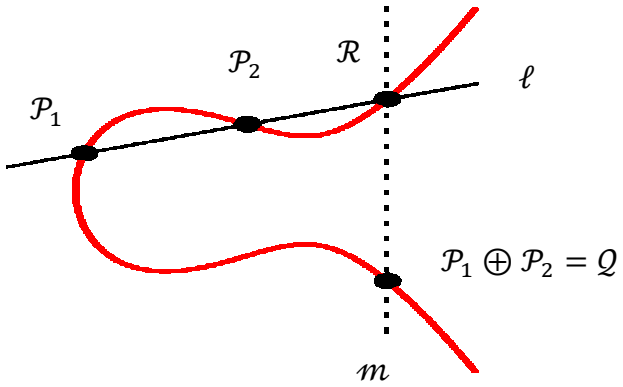
2.4.4 Grup Yapısı

Şimdi *E* üzerinde tanımlanmış toplama işleminden biraz bahsedelim.

$$\oplus : E \times E \longrightarrow E$$

E 'deki her bir nokta çifti $\mathcal{P}_1,\ \mathcal{P}_2$ için $\mathcal{P}_1 \oplus \mathcal{P}_2$:

- ℓ , $\mathcal{P}_1$ ve $\mathcal{P}_2$ 'yi birbirine bağlayan bir doğru olsun.(Eğer $\mathcal{P}_1 = \mathcal{P}_2$ ise bu doğru bir teğet doğrusudur.) O halde ℓ, E 'yi üçüncü bir $\mathcal{R}$ noktasında keser.
- $m, \mathcal{R}$ ve $\mathcal{O}$ 'yı birbirine bağlayan doğru olsun.(Eğer $\mathcal{R} = \mathcal{O}$ ise bu doğru bir teğet doğrusudur.) O halde m, E 'yi üçüncü bir $\mathcal{Q}$ noktasında keser.
- O halde $\mathcal{P}_1 \oplus \mathcal{P}_2 = \mathcal{Q}$ 'dur.



Şekil 1: Toplama İşlemi

2.4.4.1 Teorem:

$(E,\oplus)$, $\mathcal{O}$ elemanıyla bir değişmeli gruptur, [1].

2.4.4.2 Önerme:

- Eğer bir ℓ doğrusu, E eliptik eğrisi ile P,Q ve R noktalarında kesişiyorsa, o halde

$$(P \oplus Q) \oplus R = 0.$$

- Her $P \in E$ noktası için $P \oplus 0 = P$.
- Her $P, Q \in E$ noktaları için $P \oplus Q = Q \oplus P$.
- $P \in E$ olsun.

$$P \oplus (\ominus P) = 0$$

olacak şekilde E'nin bir $\ominus P$ noktası vardır.

- $P, Q, R \in E$ olsun. O halde

$$(P \oplus Q) \oplus R = P \oplus (Q \oplus R) \quad [3].$$

2.4.4.3 Lemma:

$P = (x_1, y_1)$ noktası için $-P = (x_1, -y_1)$ olur, [3].

2.4.4.4 Tanım (Toplama kuralı):

$P_1 = (x_1, y_1)$ ve $P_2 = (x_2, y_2)$, E üzerinde iki nokta olsun.

- $x_1 \neq x_2$ için $P_1 \oplus P_2 = (x_3, y_3)$ olsun.

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$

$$y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3).$$

- $x_1 = x_2$ için (dublication formula)

  i) $y_1 = -y_2$ ise $P_1 \oplus P_2 = \mathcal{O}$,

  ii) $y_1 \neq -y_2$ ise $2P_1 = (x_3, y_3)$ olsun.

$$x_3 = \left(\frac{3{x_1}^2 + A}{2y_1}\right)^2 - 2x_1$$

$$y_3 = -y_1 + \left(\frac{3{x_1}^2 + A}{2y_1}\right)(x_1 - x_3).$$

Eğer $P_1, P_2 \in E(L)$ ise unutmayın ki $P_1 \oplus P_2 \in E(L)$ dır, [1].

Eğer tanımladığımız bu toplama işlemi için $P = (x_1, y_1)$ ve $Q = (x_2, y_2)$ noktalarının toplamı $P \oplus Q$'yu hesaplamak için MAGMA hesaplama programında yazılmış fonksiyon Fonksiyon 1'dir

Fonksiyon 1: FindQP(x1,x2,y1,y2) fonksiyonu

```
FindQP:=function(x1,x2,y1,y2);
        P:=(x1,y1);
        Q:=(x2,y2);
        if x1 ne x2 then
                x3:=[(y2-y1)/(x2-x1)]^2-x1-x2;
                y3:=-y1+[(y2-y1)/(x2-x1)]*(x1-x3);
                PplusQ:=(x3,y3);
                return PplusQ;
        else if y1 eq -y2 then return "infinity";
                else
                        x3:=[(3*x1^2+A)/(2*y1)]^2-2*x1;
                        y3:=-y1+[(3*x1^2+A)/(2*y1)]*(x1-x3);
                        DoubleP:=(x3,y3);
                        return DoubleP;
                end if;
        end if;
end function;
```

2.4.4.5 Örnek:

$y^2 + y = x^3 - x^2$ Eliptik eğrisini ve üzerindeki $P = (0,0)$ noktasını alalım. $2P(= P + P)$ ve $3P(= P + 2P)$ bulunuz.

Çözüm: İlk olarak eliptik eğri üzerinde $y \to y - \frac{1}{2}$ ve $x \to x + \frac{1}{3}$ dönüşümlerini yapalım.

$$y^2 = x^3 - \frac{1}{3}x + \frac{19}{27}$$

Şeklinde Weierstrass formunda bir eliptik eğri elde etmiş olduk. Ve bu eğri üzerinde bize verilmiş olan $P = (0,0)$ noktası $Q = (-\frac{1}{3}, \frac{1}{2})$ 'na dönüşmüş oldu. Şimdi az önce

tanımladığımız toplama işlemini kullanırsak $2Q = (\frac{2}{3}, -\frac{1}{2})$ buluruz. Eğer işlemlere devam edecek olursak ve $3Q = 2Q + Q = (\frac{2}{3}, \frac{1}{2})$ elde ederiz.

Fark edeceğiniz gibi $3Q = -(2Q)$. O halde Q noktası mertebesi 5 olan bir noktadır. Yani $5Q = \mathcal{O}$'dur.

Bu işlemleri, tanımlanan Fonksiyon 1'i kullanarak da MAGMA hesaplama programında, uygulamalar yapabiliriz. Ama eğer bize sadece iki *P* noktası verilip *aP* 'yi bulmamız istenirse, çok büyük *a* değerleri için bile kullanabileceğimiz bir algoritma bulunmaktadır.

2.4.4.6 Tanım:

E/K bir eliptik eğri ve $P \in E(K)$ olsun. $a \geq 2$ tamsayısı ve $(1b_1b_2b_3 \ldots b_n)$ *a*'nın ikilik sistemdeki yazılışı olsun. aP'yi hesaplayan *Katla-ve-Ekle Algoritması;*

1. *Q:=P*
2. *j=1,2,...,n için*
   a. *eğer* $b_j = 0$ *ise Q:=2Q,*
   b. *eğer* $b_j = 1$ *ise Q:=2Q⊕P.*

Adımlarıyla *Q=aP* 'yi yaklaşık $\log_2 a$ ikiye katlama ve $(\log_2 a)/2$ toplama işlemiyle bulmamızı sağlar.

Bu algoritma için yazılmış Fonksiyon 3'ü vermeden önce elimizdeki *a* tamsayısını ikilik sistemde yazdıracak Fonksiyon 2'yi verelim.

Fonksiyon 2: BinaryExpansion(a) fonksiyonu

```
BinaryExpansion:=function(a);
        b:=a;
        table:=[];
        repeat
                if b mod 2 eq 0 then table cat:=[0];
                else table cat:=[1];
                end if;
                b:=b div 2;
                until b eq 0;
        return Reverse(table);
end function;
```

Artık verilen bir tamsayıyı ikilik sistemde yazabildiğimize ve eliptik eğriler üzerinde nokta toplamı tanımını kullanmamızı sağlayan Fonksiyon 1'i de daha önce verdiğimize göre şimdi asıl fonksiyonumuz olan Fonksiyon 3'ü verebiliriz.

Fonksiyon 3: FindaP(a,x1,y1) fonksiyonu

```
FindaP:=function(a,x1,y1);
        b:=BinaryExpansion(a);
        k:=#b;
        for i in [1..k] do
                if b[i+1] eq 0 then Q:=FindQP(x1,x1,y1,y1);
                        else
                        Q:=FindQP(x1,x1,y1,y1);
                        x3:=Q[1];
                        y3:=Q[2];
                        Q:=FindQP(x3,x1,y3,y1);
                        end if;
                end for;
        return Q;
end function;
```

3. BÜKÜM(TORSİYON) ALT GRUPLARI

3.1 Büküm(Torsiyon) Noktaları

3.1.1 Tanım:

P herhangi bir grubun bir elemanı olsun. Eğer

$$mP = \underbrace{P + P + \cdots + P}_{m \text{ tane}} = \mathcal{O}$$

olacak şekilde bir m var ise P, *m sonlu mertebeye sahiptir* değilse P, *m sonsuz mertebeye sahiptir* denir. Burada sonsuzdaki nokta $\mathcal{O}$, grubun sıfır elemanı olarak alınmıştır, [2].

3.1.2 Tanım:

E, K cismi üzerinde tanımlanmış bir eliptik eğri ve n bir pozitif tamsayı olsun.

$$E[n] = \{P \in E(\overline{K}) \mid nP = \mathcal{O}\}$$

kümesine E 'nin *n-inci mertebeden noktalarının kümesi(n-inci büküm alt grubu)* ve bu kümenin her bir P elemanına da *büküm(torsion) noktası* denir. Burada dikkat edilirse $E[n]$, koordinatları sadece K da olan noktaları değil $\overline{K}$ da olan noktaları da içinde bulunduruyordur, [1].

3.1.3 Teorem:

E, $\mathbb{F}$ cismi üzerinde bir eliptik eğri ve $n \in \mathbb{Z}^+$ olsun. Eğer $\mathbb{F}$ nin karakteristiği *n* yi bölmezse veya sıfırsa

$$E[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n$$

dir. Eğer $\mathbb{F}$ 'nin karakteristiği $p > 0$ ve $p|n$ ise $p \nmid \acute{n}$ olacak şekilde $n = p^r \acute{n}$ 'dir. O halde

$$E[n] \cong \mathbb{Z}_{\acute{n}} \times \mathbb{Z}_{\acute{n}} \text{ veya } \mathbb{Z}_n \times \mathbb{Z}_{\acute{n}}$$

olur, [1].

3.1.4 Tanım:

E bir eliptik eğri ve $n \geq 1$ olacak şekilde bir pozitif tamsayı olsun. Sonlu mertebeli noktaların kümesi,

$$E_{tors} = \bigcup_{m=1}^{\infty} E[n]$$

E'nin torsiyon alt grubunu belirtir ve kısaca bu grup E_{tors} ile gösterilir, [3].

3.2 Bölme Polinomları

Şimdi, daha önce tanımladığımız toplama formüllerine daha detaylı bakalım. $m \in \mathbb{Z}_{\geq 1}$ olacak şekilde $(x_3, y_3) = m(x_1, y_1)$ için

$$x_3 = \left(\frac{\mathcal{X}_m(x_1)}{\phi_m(x_1)}\right) \in K(x_1)$$

formunda olan formüller fark edileceği gibi $\phi_m(x_1) = 0$ için çözümsüzdür.

Burada ϕ_m polinomunun squarefree kısmı olan $\psi_m \in \mathbb{Z}[x_1, A, B]$ polinomuna *m-inci bölüm polinomu* denir. Şimdi bu polinomların nasıl tanımlandıklarını görelim.

Büküm alt gruplarını çalışabilmek için eliptik eğriler üzerinde, tamsayıyla çarpımla verilen bir bağıntı tanımlamamız gerekmektedir. Bu eliptik eğrinin bir endomorfizmasıdır ve rasyonel fonksiyonlarla tanımlanabilir. Şimdi bu fonksiyonlar için formülleri verelim. *A* ve *B* değişkenleriyle *bölüm polinomları* $\psi_m \in \mathbb{Z}[x, y, A, B]$;

$$\psi_{-1} = -1$$
$$\psi_0 = 0$$
$$\psi_1 = 1$$
$$\psi_2 = 2y$$
$$\psi_3 = 3x^4 + 6Ax^2 + 12Bx - A^2$$
$$\psi_4 = 4y(x^6 + 5Ax^4 + 20Bx^3 - 5A^2x^2 - 4ABx - 8B^2 - A^3)$$
$$\vdots$$
$$\psi_{2m+1} = \psi_{m+2}{\psi_m}^3 - \psi_{m-1}{\psi_{m+1}}^3 \qquad ,m \geq 2 \; için$$
$$\psi_{2m} = (2y)^{-1}(\psi_m)\left(\psi_{m+2}{\psi_{m-1}}^2 - \psi_{m-2}{\psi_{m+1}}^2\right) \quad ,m \geq 3 \; için$$

şeklinde tanımlanır, [1].

3.2.1 Lemma:

n tek iken ψ_n, $\mathbb{Z}[x, y^2, A, B]$ içinde tanımlı bir polinom ve *n* çift iken ψ_n, $2y\mathbb{Z}[x, y^2, A, B]$ içinde tanımlı bir polinom belirtir, [1].

Aşağıdaki polinomları tanımlayalım.

$$\phi_m = x{\psi_m}^2 - \psi_{m+1}\psi_{m-1}$$
$$\omega_m = (4y)^{-1}\left(\psi_{m+2}{\psi_{m-1}}^2 - \psi_{m-2}{\psi_{m+1}}^2\right)$$

Yukarıdaki tanımlamayla bölüm polinomlarını hesaplamak için yazılan Fonksiyon 4'ü verelim.

Fonksiyon 4: DivPol(m,q,A,B); fonksiyonu

```
function DivPol(m,q,A,B);
        R<y,x>:=PolynomialRing(FiniteField(q),2);
        if m eq -1 then
                return R!-1;
        else if m eq 0 then
                 return R!0;
         else if m eq  1 then
                 return R!1;
         else if m eq  2 then
                 return R!(2*y);
        else if m eq 3 then
                 return R!(3*x^4+6*A*x^2+12*B*x-A^2);
        else if m eq 4 then
                 return R!(4*y*(x^6+5*A*x^4+20*B*x^3-5*A^2*x^2-4*A*B*x-
8*B^2-A^3));
        else if m mod 2 eq 0 then
                k:=R!DivPol(m div 2,q,A,B);
                l:=R!DivPol((m div 2) + 2,q,A,B);
                t:=R!DivPol((m div 2) - 1,q,A,B);
                s:=R!DivPol((m div 2) - 2,q,A,B);
                n:=R!DivPol((m div 2) + 1,q,A,B);
                return R!(k *(l*t^2-s*n^2) div (2*y));
        else
                p:=R!DivPol((m+3) div 2,q,A,B);
                 r:=R!DivPol((m-1) div 2,q,A,B);
                 u:=R!DivPol((m-3) div 2,q,A,B);
                w:=R!DivPol((m+1) div 2,q,A,B);
                return R!(p*r^3-u*w^3);
         end if;
        end if;
       end if;
      end if;
     end if;
    end if;
end function;
```

3.2.2 Lemma:

Her n için $\phi_n \in \mathbb{Z}[x, y^2, A, B]$ dir. Eğer n tek ise, $\omega_n \in y\mathbb{Z}[x, y^2, A, B]$ ve eğer n çift ise, $\omega_n \in \mathbb{Z}[x, y^2, A, B]$ dir, [1].

Şimdi A ve B değişkenlerini içinde bulundukları cisim veya halkayı belirtmeden kullanmaya devam edelim ve bir

$$E: \quad y^2 = x^3 + Ax + B, \quad 4A^3 + 27B^2 \neq 0$$

eliptik eğrisini ele alalım. $\mathbb{Z}[x, y^2, A, B]$ içindeki polinomları, $y^2 = x^3 + Ax + B$ eşitliğinden yararlanarak $\mathbb{Z}[x, A, B]$ içindeki polinomlar olarak belirtebiliriz.

3.2.3 Teorem:

$P = (x, y), \quad y^2 = x^3 + Ax + B$ eliptik eğrisi üzerinde bir nokta ve n bir pozitif tamsayı olsun. O halde

$$nP = \left(\frac{\phi_n(x)}{{\psi_n}^2(x)}, \frac{\omega_n(x, y)}{{\psi_n}^3(x, y)} \right) \text{ [1].}$$

3.2.4 Sonuç:

E bir eliptik eğri olsun. n ile çarpma olarak tanımlanan E'nin endomorfizması n^2 derecelidir [1].

Bölme polinomlarını, sadece x'e bağlı birer polinom olarak, daha sonra Schoof Algoritması içerisinde kullanacağız. Şimdi bu fonksiyonu tanımlayalım.

$m \in \mathbb{Z}_{\geq 1}$ için $f_m(x)$ fonksiyonu;

m çift ise;

$$f_m(x) = \frac{\psi_m(x,y)}{y} \ ,$$

m tek ise;

$$f_m(x) = \psi_m(x,y)$$

şeklindedir.

Bu fonksiyon için yazılmış Fonksiyon 5'i verelim.

Fonksiyon 5: fdivpol(m,q,A,B) fonksiyonu

```
function fdivpol(m,q,A,B);
        Psi := DivPol(m,q,A,B);
        RR<y,x> := Parent(Psi);
        S<Y,X>:=quo<RR|[y^2-x^3-A*x-B]>;
        Psi_elim := RR!(S! Psi);
        R<x> := PolynomialRing(FiniteField(q));
        h := hom<RR->R | 0,x>;
        if m mod 2 eq 1 then
                return h(Psi_elim);
        else
                return h(Psi_elim div y);
        end if;
end function;
```

4. SONLU CİSİMLER ÜZERİNDEKİ ELİPTİK EĞRİLER

$\mathbb{F}$ karakteristiği 2 ve 3'ten farklı bir sonlu cisim ve E bu sonlu cisim üzerinde tanımlanmış bir eliptik eğri olsun. $x, y \in \mathbb{F}$ için sadece sonlu sayıda (x, y) ikilileri bulunduğundan $E(\mathbb{F})$ grubu sonludur. Çalışmalarımızda p asal iken $\mathbb{F}_p$ sonlu cisim ve $q = p^k,\ k \geq 1$ iken $\mathbb{F}_q$ sembolü sonlu cisim genişlemesini temsil edecektir. Bu grubun mertebesinin kriptografi için ne kadar önemli olduğunu ileriki bölümlerde göreceğiz. Şimdilik sonlu cisimler üzerindeki eliptik eğriler için temel teoremleri verelim.

Sırada, E. Artin'in tezinde varsayımda bulunduğu ve daha sonra Hasse tarafından 1930'larda ispatlanmış önemli bir teorem var.

4.1 Hasse Teorem

E, $\mathbb{F}_q$ sonlu cismi üzerinde tanımlanmış bir eliptik eğri olsun. O halde $E(\mathbb{F}_q)$ nun eleman sayısı $\#E(\mathbb{F}_q)$,

$$\left|q + 1 - \#E(\mathbb{F}_q)\right| \leq 2\sqrt{q}$$

eşitsizliğini sağlar.

Hasse'nin teoremi , $E(\mathbb{F}_q)$ 'daki noktaların sayısı için bir sınır belirler. Fakat bu, q asal sayısı çok büyük olduğunda $\#E(\mathbb{F}_q)$ 'yı bulmamız için pratik bir algoritma sağlamaz. Biz burada teoremin sadece ifadesini veriyoruz. İspatı için bakınız [3].

4.2 Frobenius Endomorfizmi

$\mathbb{F}_q$, cebirsel kapanışı $\overline{\mathbb{F}}_{\mathrm{q}}$ olan bir sonlu cisim olsun.

$$\phi_q : \overline{\mathbb{F}}_{\mathrm{q}} \longrightarrow \overline{\mathbb{F}}_{\mathrm{q}} \ ,$$

$$x \mapsto x^q$$

$\mathbb{F}_q$ için bir *Frobenius haritasıdır. E,* $\mathbb{F}_q$ sonlu cismi üzerinde tanımlanmış bir eliptik eğri olsun. Yani ϕ_q , $E(\overline{\mathbb{F}}_{\mathrm{q}})$ içindeki noktaların koordinatlarına

$$\phi_q(x, y) = (x^q, y^q), \qquad \phi_q(\mathcal{O}) = \mathcal{O}$$

şeklinde etki eder.

ϕ_q , *E* eliptik eğrisi üzerinde bir endomorfizm olduğundan ${\phi_q}^2 = \phi_q \circ \phi_q$ şeklinde ve her $n \geq 1$ için ${\phi_q}^n = \phi_q \circ \phi_q \circ \ldots \circ \phi_q$ biçiminde bir endomorfizm olur [1].

4.2.1 Teorem:

E, $\mathbb{F}_q$ sonlu cismi üzerinde tanımlanmış bir eliptik eğri olsun.

$$t = q + 1 - \#E(\mathbb{F}_q)$$

olacak şekilde bir *t* tamsayısı alalım. *E* 'nin endomorfizmaları ile

$${\phi_q}^2 - t\phi_q + q = 0$$

eşitliğini sağlayan *t* tamsayısı tektir.

Diğer bir değişle, eğer $(x, y) \in E(\overline{\mathbb{F}}_{\mathrm{q}})$ ise

$$\left(x^{q^2}, y^{q^2}\right) - t(x^q, y^q) + q(x, y) = \mathcal{O}$$

eşitliğinde *t,* her $(x, y) \in E(\overline{\mathbb{F}}_q)$ için tektir. Dahası, *t*, $\gcd(m, q) = 1$ olacak şekilde her bir *m* için

$$t \equiv Trace\left(\left(\phi_q\right)_m\right) \bmod m$$

denkliğini sağlayan tek tamsayıdır (Burada *t* tamsayısına *Frobenius izi* denilir)[1].

5. ELİPTİK EĞRİLER ÜZERİNDE RASYONEL NOKTA SAYIMI

Eliptik eğrilerin bu kadar çok önemli hale gelmesinin nedeni sahip oldukları rasyonel noktaların sayısıdır. Özellikle kriptografi için bir eliptik eğri üzerindeki rasyonel nokta sayısını düzgün bir şekilde hesaplamak önemlidir.

5.1 Legendre Sembolleri

Sonlu cisim üzerindeki $E: y^2 = x^3 + Ax + B$ eliptik eğrisinin noktalarının listesini yapmak için öncelikle *x*'in olası değerlerinin her birine bakılıp sonra eğer varsa $x^3 + Ax + B$ nin yani *y* nin karekökleri bulunmalıdır. Bu prosedür basit nokta sayım algoritması için bir temel oluşturmaktadır.

Bir *p* tek asal sayısı için *Legendre Sembolü* $\left(\frac{x}{p}\right)$

$$\left(\frac{x}{p}\right) = \begin{cases} +1 & eğer\ \ t^2 \equiv x\ (mod\ p) \quad t \not\equiv 0\ (mod\ p)\ için\ bir\ sonucu\ varsa\ , \\ -1 & eğer\ \ t^2 \equiv x\ (mod\ p) \quad için\ bir\ \ t\ sonucu\ yoksa\ , \\ 0 & eğer\ \ \ x \equiv 0\ (mod\ p) \end{cases}$$

olarak tanımlanır. Bu tanım q tek ve $\mathbb{F}_q$ herhangi bir sonlu cisim olmak üzere $x \in \mathbb{F}_q$ için

$$\left(\frac{x}{\mathbb{F}_q}\right) = \begin{cases} +1 & eğer\ \ \ t^2 = x \quad için\ bir\ t\ sonucu\ varsa\ \ \ t \in \mathbb{F}_q{}^{\times}, \\ -1 & eğer\ \ \ t^2 = x \quad için\ bir\ \ t\ sonucu\ yoksa\ t \in \mathbb{F}_q\ \ , \\ 0 & eğer\ \ \ x = 0 \end{cases}$$

şeklinde genelleştirilebilir [1].

5.1.1 Teorem:

E, $\mathbb{F}_q$ sonlu cismi üzerinde $y^2 = x^3 + Ax + B$ formunda tanımlanmış bir eliptik eğri olsun. O halde

$$\#E(\mathbb{F}_q) = q + 1 + \sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + Ax + B}{\mathbb{F}_q} \right)$$

dır, [1].

Lang-Trotter metodu olarak da bilinen bu teorem, $q < 100$ gibi küçük q değerleri için hızlı çalışır ancak büyük q değerleri için yavaştır. Belki en fazla q yaklaşık 10^{100} olabilir.

$E/\mathbb{F}_q$ sonlu bir cisim üzerinde tanımlanmış bir E eliptik eğrisi için Hasse teoremi

$$\#E(\mathbb{F}_q) = q + 1 - t, \qquad |t| \leq 2\sqrt{q}$$

Bağıntıları eliptik eğri üzerindeki rasyonel nokta sayısı için bir aralık belirler. Burada $\#E(\mathbb{F}_q)$ nı hesaplamak için kullanılan yöntemlerden biri de

$$t = \sum_{x \in \mathbb{F}_q} \left(\frac{f(x)}{q} \right)$$

toplamını hesaplamaktır. Her bir $\left(\frac{f(x)}{q} \right)$ Legendre sembolü $O(\log q)$ adımda hesaplanabilir. Yani bu aşikâr formüller $O(q \log q)$ adımda sonlanır. Bu da gösteriyor ki bu üstel zamanlı algoritma tercih edilecek kadar randımanlı çalışan bir algoritma değildir.

Bu tezde $\#E(\mathbb{F}_q)$ 'yi bir c sabiti için $O((\log q)^c)$ adımda polinom zamanlı hesaplayan Schoof Algoritması'nı detaylı bir şekilde inceleyeceğiz. Ama önce $O(\ldots)$

işaretinin ne anlama geldiğinden ve nerelerde kullanıldığından kısaca bahsedelim. Daha detaylı bilgi edinmek isteyenler [4]'den faydalanabilirler.

6. HESAPLAMALARIN KARMAŞIKLIĞI

6.1 Büyük-O İşareti (The Big-O Notation)

Herhangi bir pozitif n tamsayısı için pozitif değerler veren $f(n)$ ve $g(n)$ fonksiyonlarını ele alalım. Eğer bu fonksiyonlar arasında $f(n) \leq C.g(n)$ olacak şekilde bir C sabiti bulunabiliyorsa bunu kısaca $f(n) = O(g(n))$ yazarak anlatabiliriz. İşte bu gösterimde kullanılan işarete *Büyük-O İşareti* denir. Örneğin; $2n^2 + 3n - 3 = O(n^2)$.

Pratikte, aslında Büyük-O işaretinin kullanıldığı zamanlarda f ve g fonksiyonlarının ne olduğuyla ya da n olarak alınan küçük değerlerin ne olduklarıyla ilgilenmiyoruz.

Kriptografide kullanılan sayılar ikilik sistemde yazılı kullanıldıkları için bu sayıların uzunlukları yani sahip oldukları hane sayısı önemlidir. Bir n sayısının sahip olduğu hane sayısı

$$boy(n) = 1 + [\log_2 n] = 1 + \left[\frac{\ln n}{\ln 2}\right]$$

olarak tanımlanır ve bu değer $O(\ln n)$ olarak alınabilir.

6.2 Tahmini Süre

Kullanılan sayıların hepsi 2 tabanında yazılmış sayılar. Yani iki sayının toplamında 2 tabanında toplama işlemi yapılıyor. Bu yapılan toplama işleminin özel adı ise: *bit operasyonu(bit operation)*. Yani k-bit'lik iki sayının toplamı için k-bit operasyon gerekiyor. Yani bir bilgisayarın verilen bir işi yerine getirme zamanı ile

bit operasyon sayısı aslında orantılıdır. Tahmini zamanı aslında tahmini bit operasyonu sayısıdır.

İki sayıyı toplamak için gerekli zaman aslında iki sayıdan uzunluğu büyük olana eşittir. Yani

$$Time(k - bit + l - bit) = \max(k, l).$$

Eğer $k = boy(m) = O(\ln m)$ ve $l = boy(n) = O(\ln n)$ ise

$$Time(m + n) = O(\max(\ln m, \ln n)).$$

Şimdi yukarıda kullandığımız iki sayının çarpımı için tahmini zaman nasıl olur ona bakacak olursak. Zaman tahmininde hiçbir zaman "tek bir doğru sonuç" olmadığını görürüz. Mesela; *k*-bit'lik bir sayıyı *l*-bit'lik bir sayıyla çarptığımızda tahmini zaman sıradakilerden biri olabilir. 1.$Time = O(kl)$; 2.$Time < kl$; 3.$Time \leq k(l - 1)$; 4.Eğer ikinci sayının 0-bit'lik ve 1-bit'lik hane sayısı eşit ise $Time \leq k\,l/2$. Ama bizim için her zaman Büyük-O notasyonuyla ifade edilen hali geçerli olacaktır.

$$Time = O(kl)$$

Eğer sayıların boyutlarıyla bir zaman tahmini yapacak olursak;

$$Time(m \times n) = O(\ln m \ln n).$$

Eğer aldığımız iki sayı aynı boyutta ise;

$$Time(k - \text{bit} \times k - \text{bit}) = O(k^2)$$

tahmini zaman için bir sonuçtur. Ama çok büyük sayıların çarpımı için bu çok uzun bir süre olduğundan matematikçiler tahmini zamanı sadece $O(k \ln k \ln \ln k)$ bit operasyonu olan yani $O(k^2)$'den daha kısa sürecek yöntemler araştırıyorlar.

6.3 Algoritmalar

Genel olarak, bir şeyi yapmak için gerekli olan bit operasyonları sayısını belirlerken; ilk adım, görevi yerine getirmek için yapılacak olanları ana hatlarıyla belirlemektir. Hesaplamayı yapmak için izlenecek adım adım prosedüre *algoritma* denilir. Tabi ki de bir görevi yerine getirmek için birden fazla farklı algoritma yazılabilir.

Bu tezin ana konusu olan Schoof Algoritması da eliptik eğriler üzerindeki rasyonel noktaların sayısını hesaplamak için Rene Schoof tarafından yazılmış polinom zamanlı bir algoritmadır.

Şimdi algoritmalar için temel teşkil eden bir tanım verelim.

6.3.1 Tanım:

Eğer toplam boyutları yaklaşık k olan tamsayılar için çalışan bir algoritma için gereken bit operasyonları sayısı (tahmini zamanı) $O(k^d)$ olacak şekilde bir d tamsayısı varsa bu algoritmanın hesaplama performansına *polinom zamanlı* algoritma denir [4].

Basit aritmetik işlemler $+, -, \div, \times$ için yapılan operasyonlar polinom zaman için verilebilecek en basit örneklerdir.

Tanımı biraz açıklayacak olursak. Burada elimizdeki k değeri aslında bizim algoritmamızın girdisinin boyutudur. Algoritmanın tahmini süresi ise yine bu k değerinin bir d tamsayı üssüyle oranlıdır.

7. SCHOOF ALGORİTMASI

1985'te Schoof, $\mathbb{F}_q$ sonlu cisimleri üzerinde tanımlanmış eliptik eğrilerin noktalarını saymak için bir algoritma yayınladı [5]. Öyle ki bu algoritma çok büyük q asalları için bile var olan algoritmalardan çok daha hızlı çalışıyordu. Özellikle, karşılaştırma yapacak olursak Baby Step- Giant Step metodunda $q^{1/4}$ bit işlerken, Schoof Algoritması için sadece $\log^8 q$ bit gerekiyordu. Daha sonraki senelerde, Atkin ve Elkies, Schoof'un metodunu da kullanarak SEA(Schoof-Elkies-Atkin) metodunu geliştirdi. Böylelikle günümüzde birkaç yüz karaktere sahip bir q için bile çok başarılı bir şekilde kullanılabiliyor. Bu tezde sadece Schoof'un metodu üzerinde duracağız. Atkin ve Elkies 'in metotlarının detayları için [6] nolu kaynakağa bakılabilir..

E, $\mathbb{F}_q$ sonlu cismi üzerinde $y^2 = x^3 + Ax + B$ formunda tanımlanmış bir eliptik eğrisi verilsin. $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ çözümlerinin sayısını ya da buna denk olarak $E(\mathbb{F}_q)$ 'da kaç tane nokta olduğunu bulmak istiyoruz. Hasse Teoremi'nden

$$\#E(\mathbb{F}_q) = q + 1 - t, \qquad |t| \leq 2\sqrt{q}$$

olduğunu biliyoruz. $S = \{2,3,5,7, \dots, L\}$ şeklinde asal sayıların bir kümesi olsun öyle ki;

$$\prod_{\ell \in S} \ell > 4\sqrt{q}\,.$$

Algoritmada kullanmak için gerekli olan ℓ asal sayı değerlerini bulan Fonksiyon 6'yı verelim.

Fonksiyon 6: Findell(q) fonksiyonu

```
Findell:=function(q);
        p:=Factorization(q)[1][1];
        bell:=1;
         y:=1;
        table := [];
         repeat
                bell:=NextPrime(bell);
                        if bell ne q then
                                table cat:=[bell];
                                 y*:=bell;
                        end if;
        until 16*q lt y^2;
         return table;
end function;
```

Eğer her bir $\ell \in S$ asalı için $t \pmod{\ell}$ 'yi hesaplayabilirsek, $t \pmod{\prod \ell}$ 'yi oradan da *t* için tek bir sonuç elde edebiliriz.

ℓ bir asal sayı olsun. Kolaylık olması için ℓ'nin $\mathbb{F}_q$ 'nun karakteristiği olan p'den farklı olduğunu ve q'nun tek sayı olduğunu varsayıyoruz. Hesaplamak istediğimiz $t \pmod{\ell}$.

Eğer $\ell = 2$ ise.

- Eğer $x^3 + Ax + B$, $e \in \mathbb{F}_q$ köküne sahipse, o halde $(e, 0) \in E[2]$ ve $(e, 0) \in E(\mathbb{F}_q)$ 'dir. Yani $E(\mathbb{F}_q)$ çift sayıda elemana sahiptir. Bu durumda $q + 1 - t \equiv 0(mod\ 2)$ yani *t* çifttir.
- Eğer $x^3 + Ax + B$, $\mathbb{F}_q$ 'de bir köke sahip değilse, o halde $E(\mathbb{F}_q)$ 'nin mertebesi 2 olan bir noktası yoktur. Yani *t* tektir.

$x^3 + Ax + B$ 'nin $\mathbb{F}_q$ 'de köklerinin olup olmadığına karar verebilmek için $\mathbb{F}_q$ 'deki tüm elemanları denememiz gerekmekteydi. Ama elimizde çok daha hızlı bir yol var. $x^q - x$'nin kökleri aynı zamanda $\mathbb{F}_q$ 'nin elemanlarıdır. Bundan dolayı, $x^3 + Ax + B$ 'nin $\mathbb{F}_q$ 'de kökü vardır ancak ve ancak $x^3 + Ax + B$, $x^q - x$ ile ortak

köke sahiptir. Öklid algoritması –polinomlar üzerinde- da bize gereken iki polinomun gcd'sini verir.

Eğer q çok büyük bir sayı ise, x^q polinomu da çok büyük bir dereceye sahiptir. Buradan

$$x_q \equiv x^q (mod\, x^3 + Ax + B)$$

ardı ardına yapılan toplama işlemleriyle hesaplanır ve sonucu

$$gcd(x_q - x, x^3 + Ax + B) = \text{gcd}\ (x^q - x, x^3 + Ax + B)$$

içinde kullanılır. Eğer gcd 1 ise, hiç ortak kök yoktur yani t tektir. Eğer gcd 1 değil ise, t çifttir. Bu da $\ell = 2$ durumunu tamamlar. (7.1)

$\ell = 2$ durumu için yazılmış Fonksiyon 7'yi verelim.

Fonksiyon 7: Schoofmod2(q,A,B) fonksiyonu

```
Schoofmod2:=function(q,A,B);
                Fq:=FiniteField(q);
                R<x>:=PolynomialRing(Fq);
                S<X>:=quo<R | [x^3+A*x+B] >;
                xqx := R ! (X^q - X);
                g:=GCD(xqx,x^3+A*x+B);
                if g eq 1 then
                        return 1;
                else
                        return 0;
                end if;
end function;
```

Birazdan x^q ve x^{q^2} gibi farklı ifadeler kullanılacak. Onlar da aynı $\ell = 2$ durumunda olduğu gibi polinomsal mod alınarak hesaplanacaktır.

ψ_n, bölüm polinomlarını daha önce tanımlamıştık. n tek olduğunda ψ_n, x'li bir polinom ve $(x,y) \in E(\overline{\mathbb{F}}_q)$ için

$$(x,y) \in E[n] \Leftrightarrow \psi_n(x) = 0$$

olur. Bu polinomlar, Schoof Algoritması'nda çok önemli bir rol oynamaktadırlar.

ϕ_q, Frobenius endomorfizması olsun. Yani

$$\phi_q(x,y) = (x^q, y^q)$$

ve Teorem 4.2.1' den

$$\phi_q{}^2 - t\phi_q + q = 0\,.$$

(x,y) mertebesi ℓ olan bir nokta olsun.

$$\left(x^{q^2}, y^{q^2}\right) + q(x,y) = t(x^q, y^q)\,.$$

$$q_\ell \equiv q \ (mod\ \ell), \qquad |q_\ell| < \ell/2$$

olarak alalım. $q_\ell(x,y) = q(x,y)$ eşitliğinden

$$\left(x^{q^2}, y^{q^2}\right) + q_\ell(x,y) = t(x^q, y^q) \tag{7.2}$$

yazabiliriz. (x^q, y^q) noktası da mertebesi ℓ olan bir nokta olduğundan bu son eşitlikten $t \pmod{\ell}$ hesaplanabilir. Buradaki düşünce t dışındaki bütün değerleri hesapladıktan sonra eşitlikten olması gereken t değerini saptamaktır. Eğer eşitlik herhangi bir $(x,y) \in E[\ell]$ noktası için sağlanırsa, $t \pmod{\ell}$ bulunmuş olur ve her $(x,y) \in E[\ell]$ için sağlanır.

İlk olarak varsayalım ki bazı $(x,y) \in E[\ell]$ için

$$\left(x^{q^2}, y^{q^2}\right) \neq \pm q_\ell(x,y)$$

olsun. Yani tanımdan

$$(\acute{x},\acute{y}) = \left(x^{q^2}, y^{q^2}\right) + q_\ell(x,y) \neq \mathcal{O}$$

ve $t \not\equiv 0 \pmod{\ell}$ 'dir. Bu durumda , $\left(x^{q^2}, y^{q^2}\right)$ ve $q_\ell(x,y)$ noktalarının x koordinatları farklıdır yani iki nokta toplamı, iki nokta üzerinden geçen doğru formülü kullanılarak bulunabilir.

$$\acute{x} = \left(\frac{y^{q^2} - y_{q_\ell}}{x^{q^2} - x_{q_\ell}}\right)^2 - x^{q^2} - x_{q_\ell}$$

j tamsayıları için

$$j(x,y) = (x_j, y_j)$$

yazalım. x_j ve y_j yi bölüm polinomlarını kullanarak hesaplayabiliriz. Dahası $x_j = r_{1,j}(x)$ ve $y_j = r_{2,j}(x)y$ şeklinde dönüşüm yapabiliriz.

$$(y^{q^2} - y_{q_\ell})^2 = y^2\left(y^{q^2-1} - r_{2,q_\ell}(x)\right)^2$$

$$= (x^3 + Ax + B)\left((x^3 + Ax + B)^{(q^2-1)/2} - r_{2,q_\ell}(x)\right)^2$$

Böylelikle elimizde tamamen x e bağlı bir $\acute{x}$ fonksiyonu kaldı.

$$(\acute{x},\acute{y}) = (x_j{}^q, y_j{}^q)$$

olacak şekilde bir j bulmak istiyoruz. İlk olarak keyfi $(x,y) \in E[\ell]$ için x-koordinatlarına baktığımızda

$$(\acute{x}, \acute{y}) = \pm\left({x_j}^q, {y_j}^q\right) \iff \acute{x} = {x_j}^q$$

sağlanıyorsa $E[\ell]$ içindeki her nokta için de sağlanır anlamına gelir. ψ_n'nin kökleri $E[\ell]$ 'deki noktaların *x*-koordinatları olduğundan, bu

$$\acute{x} - {x_j}^q \equiv 0 \ (mod\ \psi_\ell) \tag{7.3}$$

olduğunu belirtir. (Bu da demek oluyor ki; $\acute{x} - {x_j}^q$'nin payı, ψ_ℓ'in bir katıdır.) Burada ψ_ℓ'nin köklerini basit (simple) olarak kullanıyoruz.(Aksi taktirde ψ_ℓ 'nin sadece $\acute{x} - {x_j}^q$'nin kökleri olduğunu belirtilirdi.) $\mathbb{F}_q$'nun karakteristiğinin ℓ olmadığını varsayarsak elimizde mertebesi ℓ olan $\ell^2 - 1$ adet farklı kök vardır. Yani bu noktaların $(\ell^2 - 1)/2$ adet farklı *x*-koordinatları vardır ve bunların hepsi de ψ_ℓ nin $(\ell^2 - 1)/2$ dereceli kökleridir. Bu yüzden ψ_ℓ'nin kökleri basit (simple) olmalıdır.

Şimdi (7.3)'de olduğu gibi bir *j* değerini bulduğumuzu varsayalım. O zaman

$$(\acute{x}, \acute{y}) = \pm\left({x_j}^q, {y_j}^q\right) = \left({x_j}^q, \pm {y_j}^q\right).$$

Uygun işareti saptayabilmek için *y*-koordinatlarına bakmamız gerekmektedir. Hem $\acute{y}/y$ hem de ${y_j}^q/y$, *x* 'in bir fonksiyonu olarak yazılabilir. Eğer

$$(\acute{y} - {y_j}^q)/y \equiv 0 \ (mod\ \psi_\ell)$$

ise $t \equiv j \ (mod\ \ell)$ değilse $t \equiv -j \ (mod\ \ell)$. O halde $t(mod\ \ell)$ 'yi buluruz ki bu da ilk varsayımımızı tamamlamamızı sağlar. (7.4)

Şimdi varsayalım ki her $(x, y) \in E[\ell]$ için

$$\left(x^{q^2}, y^{q^2}\right) = \pm q(x, y)$$

olsun. Eğer

$$\phi_q{}^2(x,y) = \left(x^{q^2}, y^{q^2}\right) = q(x,y)$$

ise (7.2) 'den

$$t\phi_q(x,y) = \phi_q{}^2(x,y) + q(x,y) = 2q(x,y)$$

ve

$$t^2 q(x,y) = t^2 \phi_q{}^2(x,y) = (2q)^2(x,y)$$

eşiklikleri yazılabilir ki buradan

$$t^2 q \equiv 4q^2 (mod\ \ell)$$

olur. Bu da demek oluyor ki q , $mod\ \ell$ 'de bir tam karedir. q , $mod\ \ell$ 'de bir tam kare ise

$$w^2 \equiv q (mod\ \ell)$$

olsun. Her $(x,y) \in E[\ell]$ için

$$(\phi_q + w)(\phi_q - w)(x,y) = (\phi_q{}^2 - q)(x,y) = \mathcal{O}.$$

$E[\ell]$ 'nin herhangi bir P noktasını alalım. Buradan ya

$$(\phi_q - w)P = \mathcal{O} \ \rightarrow \ \phi_q P = wP$$

ya da

$$(\phi_q + w)\acute{P} = \mathcal{O}$$

olacak şekilde

$$\acute{P} = (\phi_q - w)P$$

sonlu bir noktadır. Her iki koşulda da

$$\phi_q P = \pm wP$$

olacak şekilde bir $P \in E[\ell]$ noktası vardır.

$\phi_q P = wP$ olacak şekilde bir $P \in E[\ell]$ noktasını alalım. Buradan

$$\mathcal{O} = ({\phi_q}^2 - t\phi_q + q)P = (q - tw + q)P$$

yani

$$tw \equiv 2q \equiv 2w^2 (mod\ \ell)$$

olur. Buradan da

$$t \equiv 2w (mod\ \ell) \tag{7.5}$$

elde edilir. Benzer şekilde , $\phi_q P = -wP$ olacak şekilde bir $P \in E[\ell]$ noktası alındığında

$$t \equiv -2w (mod\ \ell) \tag{7.6}$$

elde edilir. Bu durumda istediğimize ulaşıp ulaşmadığımızı aşağıdaki şekilde kontrol edebiliriz. Bilmemiz gereken, bazı $(x, y) \in E[\ell]$ için

$$(x^q, y^q) = \pm w(x, y) = \pm(x_w, y_w) = (x_w, \pm y_w)$$

olup olmadığıdır. Bu yüzden x 'in bir rasyonel fonksiyonu olan $x^q - x_w$ 'yi hesaplayalım. Eğer

$$\gcd\left(pay(x^q - x_w), \psi_\ell\right) \neq 1$$

ise bazı $(x, y) \in E[\ell]$ için $\phi_q(x, y) = \pm w(x, y)$ olur. Eğer bu olursa işareti saptayabilmek için *y*-koordinatlarına bakılır. Eğer

$$\gcd(pay(x^q - x_w), \psi_\ell) = 1$$

ise

$$\left(x^{q^2}, y^{q^2}\right) = q(x, y)$$

varsayımının sağlanmadığı yani aradığımız eşitliğin

$$\left(x^{q^2}, y^{q^2}\right) = -q(x, y)$$

olduğu ortaya çıkar. Bu durumda da her $P \in E[\ell]$ için

$$tP = \left({\phi_q}^2 + q\right)P = \mathcal{O}$$

$$t \equiv 0 (mod\ \ell) \tag{7.7}$$

Son olarak, her bir $\ell \in S$ için (7.1),(7.4),(7.5),(7.6) ve (7.7) 'den birinde elde ettiğimiz $t(mod\ \ell)$ değerlerini ve Çin Kalan Teoremini kullanarak $t(mod \prod \ell)$ hesaplanır. Buradan $|t| \leq 2\sqrt{q}$ olacak şekilde bir *t* seçilir ki $E(\mathbb{F}_q)$'nin noktalarının sayısı $q + 1 - t$ olarak bulunur.

Schoof'un algoritmasını özetleyecek olursak;

Yapmak istediğimiz $\mathbb{F}_q$ üzerinde tanımlanmış bir $E: y^2 = x^3 + Ax + B$ eliptik eğrisi için $\#E(\mathbb{F}_q) = q + 1 - t$ değerine ulaşmak.

1. $\prod_{\ell \in S} \ell > 4\sqrt{q}$ olacak şekilde $S = \{2,3,5,7, \dots, L\}$ kümesini şeç.

2. Eğer $\ell = 2$ ise $t \equiv 0 (mod\ 2)$ ancak ve ancak

$$\gcd(x^q - x, x^3 + Ax + B) \neq 1.$$

3. Her bir $\ell \in S$ tek asal sayısı için, sıradakileri uygulayın.
   (a) $q_\ell \equiv q\ (mod\ \ell)$ öyle ki $|q_\ell| < \ell/2$ olsun.

   (b) $\acute{x}$'nin *x*-koordinatlarını hesapla

$$(\acute{x}, \acute{y}) = \left(x^{q^2}, y^{q^2}\right) + q_\ell(x, y)\ mod\ \psi_\ell.$$

   (c) $j = 1,2, \dots, (\ell - 1)/2$ için sıradaki adımları uygulayın.
      (i) $\acute{x}$'nin *x*-koordinatlarını hesapla

$$j(x, y) = \left(x_j, y_j\right).$$

      (ii) Eğer $\acute{x} - {x_j}^q \equiv 0\ (mod\ \psi_\ell)$ ise adım (iii)'ye git. Değilse, sıradaki *j* değerini denemeye başla (adım(c)). Eğer $1 \leq j \leq (\ell - 1)/2$ aralığındaki tüm *j* değerleri denendiyse git adım(d).

      (iii) Hesapla $\acute{y}$ ve y_j . Eğer

$$\left(\acute{y} - {y_j}^q\right)/y \equiv 0\ (mod\ \psi_\ell)$$

      ise buradan $t \equiv j\ (mod\ \ell)$, değilse $t \equiv -j\ (mod\ \ell)$.

   (d) Eğer $1 \leq j \leq (\ell - 1)/2$ aralığındaki tüm *j* değerleri denendiyse ve başarı elde edilemediyse, $w^2 \equiv q (mod\ \ell)$ olsun. Eğer *w* yok ise o halde $t \equiv 0 (mod\ \ell)$.

(e) Eğer $\gcd(pay(x^q - x_w), \psi_\ell) = 1$ ise o halde $t \equiv 0(mod\ \ell)$, değilse hesapla

$$\gcd\big(pay\big((\acute{y} - {y_j}^q)/y\big), \psi_\ell\big).$$

Eğer bu gcd 1 değil ise o halde $t \equiv 2w(mod\ \ell)$. Değilse $t \equiv -2w(mod\ \ell)$.

4. Her bir $\ell \in S$ için $t(mod\ \ell)$ değerlerini $t(mod \prod \ell)$ 'yi hesaplamak için kullan. $|t| \leq 2\sqrt{q}$ olacak şekilde uygun bir t seç. $E(\mathbb{F}_q)$'nin noktalarının sayısı $q + 1 - t$.

Sırada Schoof algoritmasının ilk hali için MAGMA hesaplama programında yazılmış SANS(E); fonksiyonu var.

BAKINIZ **EK 10.1**…

Şimdi Schoof Algoritması için MAGMA hesaplama programında yazdığımız fonksiyonları da kullanarak, sayısal bir örnekle algoritmanın çalışma şeklini anlamaya çalışalım.

7.1 Sayısal Uygulama

$E: y^2 = x^3 + 2x + 1$ eliptik eğrisini *(mod 19)*'da alalım [1].

```
q:=19;
Fq:=FiniteField(q);
E:=EllipticCurve([2,1]);
E;
```

```
Elliptic Curve defined by y^2 = x^3 + 2*x + 1 over Rational Field
```

Buradan eliptik eğri üzerindeki nokta sayısı:

$$\#E(\mathbb{F}_{19}) = 19 + 1 - t$$

olacaktır. Burada $\#E(\mathbb{F}_{19})$ 'yi hesaplamak demek *t* değerini hesaplamak demektir. Yani amacımız *t* değerini hesaplamak. Algoritmanın adımlarını takip edersek. İlk olarak $S = \{2,3,5,7, \dots, L\}$ kümesi bulunmalı daha sonra her bir $\ell \in S$ elemanı için $t(mod\ \ell)$ değerleri hesaplanmalı son olarak da Çinlilerin Kalan Teoremi yardımıyla *t* değeri hesaplanmalıdır.

İlk olarak, bize gerekli olan ℓ asallarını, daha önce tanımladığımız Fonksiyon 6 ile hesaplayalım.

```
Findell(19);
```

```
[ 2, 3, 5 ]
```

Şimdi $S = \{2,3,5\}$ kümesi için bulmamız gereken

$$t \equiv \begin{cases} 1 & (mod\ 2) \\ 2 & (mod\ 3) \\ 3 & (mod\ 5) \end{cases}$$

değerlerini tek tek bulmaya çalışalım.

- $\ell = 2$ olsun. Yapmamız gereken eliptik eğrinin $\mathbb{F}_{19}$'da kökü olup olmadığına bakmak. Bunun için de ilk olarak

$$x^{19} \equiv x^3 + 13x + 14(mod\ x^3 + 2x + 1\ )$$

hesaplanır.

$$\gcd(x^{19} - x, x^3 + 2x + 1\ ) = \gcd\ (x^3 + 13x + 14, x^3 + 2x + 1\ ) = 1$$

Olduğundan $x^3 + 2x + 1$ eliptik eğrisinin $\mathbb{F}_{19}$'da kökü yoktur bulunur. Buradan $E(\mathbb{F}_{19})$'da hiç 2-torsiyon(büküm) noktası yoktur. Yani

$$t \equiv 1(mod\ 2) \tag{7.8}$$

Aynı sonucu bir de daha önce tanımladığımız Fonksiyon 7'yi kullanarak da görelim.

```
Schoofmod2(19,2,1);
```

```
1
```

- $\ell = 3$ olsun. $j = (\ell - 1)/2$ 'den $j = (3 - 1)/2 = 1$ bulunur.

$q \equiv 1(mod\ 3)$'den $q_\ell \equiv 1$ elde ederiz ki bu da demek oluyor ki her $(x, y) \in E[3]$ için

$$(x^{361}, y^{361}) + (x, y) = \pm(x^{19}, y^{19})$$

eşitliğinin doğruluğunu kontrol etmeliyiz. Üçüncü bölüm polinomu:

$$\psi_3 = 3x^4 + 6Ax^2 + 12Bx - A^2.$$

İlk olarak yapmamız gereken cisim üzerinde tanımlanmış toplama işlemini kullanarak $(x^{361}, y^{361}) + (x, y)$'nin x-koordinatını $y^2 = x^3 + 2x + 1$ eşitliğinden yararlanarak hesaplamak:

$$\left(\frac{y^{361} - y}{x^{361} - x}\right)^2 - x^{361} - x = (x^3 + 2x + 1)\left(\frac{(x^3 + 2x + 1)^{180} - 1}{x^{361} - x}\right)^2 - x^{361} - x$$

Şimdi yapmamız gereken $mod\ \psi_3$ ile indirgemek. Bunun için yapmamız gereken aslında ilk olarak genişletilmiş öklid algoritmasını kullanarak $x^{361} - x(mod\ \psi_3)$ 'ün tersini bulmak. Bunun yanında

$$\gcd(x^{361} - x, \psi_3) = x - 8 \neq 1$$

olduğundan çarpımsal bir tersinden söz edemeyiz. Onun yerine , $x = 8$,ψ_3'nin bir kökü olduğundan söyleyebiliriz ki $(8,4) \in E(\mathbb{F}_{19})$ noktasının mertebesi 3'tür. Bu yüzden

$$\#E(\mathbb{F}_{19}) = 19 + 1 - t \equiv 0(mod\ 3)$$

yani

$$t \equiv\ 2(mod\ 3) \tag{7.9}$$

- $\ell = 5$ olsun. . $j = (\ell - 1)/2$ 'den $j = (5 - 1)/2 = 2$ bulunur.
  $19 \equiv 4 \equiv -1(mod\ 5)$'den $q_\ell \equiv -1$ elde edilir ve her $(x, y) \in E[5]$ için

$$19(x, y) = -(x, y) = (x, -y)$$

eşitlikleri sağlanmış olur.

Şimdi yapmamız gereken her $(x, y) \in E[5]$ için

$$(\acute{x}, \acute{y}) \stackrel{\text{def}}{=} (x^{361}, y^{361}) + (x, -y)\ ?=?\ \pm 2(x^{19}, y^{19}) \stackrel{\text{def}}{=} (\acute{\acute{x}}, \acute{\acute{y}})$$

eşitliklerinin sağlanıp sağlanmadığını kontrol etmeliyiz.

Beşinci bölüm polinomu:

$$\psi_5 = 5x^{12} + 10x^{10} + 17x^8 + 5x^7 + x^6 + 9x^5 + 12x^4 + 2x^3 + 5x^2 + 8x + 8.$$

Eşitliklerin *x*-koordinatları

$$\acute{x} = \left(\frac{y^{361} - y}{x^{361} - x}\right)^2 - x^{361} - x \ ?\equiv? \ \left(\frac{3x^{38} + 2}{2y^{19}}\right)^2 - 2y^{19} = \acute{\acute{x}} \qquad (mod\, \psi_5)$$

şeklinde hesaplanıp daha sonra $y^2 = x^3 + 2x + 1$ eşitliğinden yararlanılarak gerekli değişimler yapıldığında elde edilen *x*'li polinomlarla eşitlik doğrulanır. Buradan

$$t \equiv \pm 2 (mod\ 5)$$

bulunur.

Şimdi işareti saptamak için *y*-koordinatlarına bakalım.

$(\acute{x}, \acute{y}) \stackrel{\text{def}}{=} (x^{361}, y^{361}) + (x, -y)$'nin *y*-koordinatı $\acute{y}$:

$$y(9x^{11} + 13x^{10} + 15x^9 + 15x^7 + 18x^6 + 17x^5 + 8x^4 + 12x^3 + 8x + 6)(mod\ \ \psi_5)$$

$\left(\acute{\acute{x}}, \acute{\acute{y}}\right) = 2(x^{19}, y^{19})$'nin *y*-koordinatı $\acute{\acute{y}}$:

$$y(13x^{10} + 15x^9 + 16x^8 + 13x^7 + 8x^6 + 6x^5 + 17x^4 + 18x^3 + 8x + 18) \ \ (mod\ \ \psi_5).$$

Hesaplamalar gösterir ki

$$\left(\acute{y} + \acute{\acute{y}}^{19}\right)/y \equiv 0 \quad (mod\ \ \psi_5).$$

Bu da demek oluyor ki

$$(\acute{x}, \acute{y}) \equiv \left(\acute{\acute{x}}^{19}, -\acute{\acute{y}}^{19}\right) = -2(x^q, y^q) \qquad (mod\ \ \psi_5)$$

yani

$$t \equiv -2 (mod\ 5) \tag{7.10}$$

Bununla birlikte algoritmamızın son adımında $S = \{2,3,5\}$ kümesinin her bir elemanı için elde ettiğimiz $t(mod\ \ell)$ değerlerini ve Çin Kalan Teoremini kullanarak $t = -7$ bulunur.

O halde $E: y^2 = x^3 + 2x + 1$ eliptik eğrisinin rasyonel nokta sayısı:

$$\#E(\mathbb{F}_{19}) = 19 + 1 - (-7) = 27.$$

Eğer bu yaptığımızı, MAGMA hesaplama programında, yazdığımız SANS(E) fonksiyonuyla bir kez daha hesaplayalım.

```
q:=19;
Fq:=FiniteField(q);
E:=EllipticCurve([2,1]);
SANS(E);
```

```
Our prime number q is 19
So we have Rational Field
We have Elliptic Curve defined by y^2 = x^3 + 2*x + 1 over Rational Field
We will have to compute t mod the following primes: [ 2, 3, 5 ]
Computing trace mod 2
Computing division polynomials...
Computing trace mod 3
Computing trace mod 5
Now verifying trace: 1
Now verifying trace: 2
Now verifying trace: 3
Success.
The traces are: [ 1, 2, 3 ]
Chinese Remainder Theorem:
Trace of Elliptic Curve is -7
Number of the Rational point on E is q+1-t and t= -7
27
```

Schoof algoritmasının eliptik eğriler üzerinde nokta sayımı için kullanılan diğer algoritmalardan farklı olarak polinom zamanlı çalışan bir algoritma olduğunu daha önce söylemiştik. Şimdi bunu daha detaylı bir şekilde görelim.

Hatırlayacak olursak bütün hesaplamalar

$$R_\ell = \frac{\mathbb{F}_q[x,y]}{(\psi_\ell(x), y^2 - f(x))}$$

bölüm halkası üzerinde yapılıyordu.

Tahmini zamanı hesaplayabilmek için gereken ilk iş algoritmayı ana hatlarıyla adım adım yazmak idi.

1. Başla $A = 1$ ve $\ell = 3$.
2. Devam et $A < 4\sqrt{q}$.
3. Devam et $n = 0,1,2, \dots, \ell - 1$.
4. Çalış R_ℓ halkasında

$$\left(x^{q^2}, y^{q^2}\right) + [q](x, y) = [n](x^q, y^q)$$

İçin bir n bulana kadar.

5. Bitir n.
6. $A = \ell . A$.
7. $n_\ell = n$.
8. Bir sonraki ℓ asalına geç.
9. Bitir A.
10. Hesaplanmış bütün n_ℓ değerleriyle Çinlilerin Kalan Teoremi'ni kullanarak $t \equiv n_\ell (mod\ \ell)$ koşulunu sağlayan n_ℓ'yı bul.
11. Bitir $\#E\left(\mathbb{F}_q\right) = q + 1 - t$.

7.2 Schoof Algoritması'nın Tahmini Çalışma Süresi

$E/\mathbb{F}_q$ bir eliptik eğri olsun. Yukarıdaki algoritma $\#E\left(\mathbb{F}_q\right)$'yı hesaplayan polinom zamanlı bir algoritmadır.

7.2.1 Lemma:

$\#E\left(\mathbb{F}_q\right)$ Schoof Algoritması kullanılarak yaklaşık olarak $O((\log q)^8)$ adımda hesaplanır, [3].

İspat: Birazdan yapacağımız ispat sadece algoritmanın tahmini süresi içindir. Öncelikle üç ana iddiayı doğrulayalım.

a) Algoritma tarafından kullanılan en büyük ℓ asal sayısı $\ell \leq O(log\ q)$ karşılamaktadır.

Asal Sayı Teoremi

$$\lim_{X\to x}\frac{1}{X}\sum_{\substack{\ell\le X\\ \ell\ prime}}\log\ell=1$$

ifadesine denktir, [7]. Buradan $\prod_{\ell\le X}\ell\approx e^{X}$ olur. Çarpımı $4\sqrt{q}$ 'den büyük yapabilmek için $X\approx\frac{1}{2}\log(16q)$ almak yeterlidir.

b) R_ℓ Halkasında yapılan çarpma işlemi $O(\ell^4(\log q)^2)$ bit operasyonda yapılabilir.

R_ℓ halkasının elemanları, derecesi $O(\ell^2)$ olan polinomlardır. Böyle iki polinomun çarpımı ve daha sonra $mod\ \psi_\ell(x)$ 'ye göre indirgenmesi $\mathbb{F}_q$ cisminde $O(\ell^4)$ elemanter işlem tutar (toplama ve çarpma). Benzer şekilde, $\mathbb{F}_q$ cisminde çarpma işlemi $O((\log q)^2)$ bit operasyonda olur. Yani R_ℓ halkasında basit işlemler $O(\ell^4(\log q)^2)$ bit operasyon tutar.

c) x^q, y^q,x^{q^2} , y^{q^2} 'yi R_ℓ halkası içine indirgemek $O(log\ q)$ bit operasyonda olur.

Genel olarak, Katla-ve-Çarp algoritması, R_ℓ'de x^n ve y^n'yi $O(\log n)$ çarpımla hesaplayabilmemizi sağlamaktadır. Bu hesaplama sadece bir kere yapılır. Yani

$$\left(x^{q^2},y^{q^2}\right)+[q\ mod\ \ell](x,y)\qquad\text{ve}\qquad(x^q,y^q)$$

noktaları hesaplanır ve daha sonra Schoof Algoritmasının 4. Adımında kullanılmak üzere kaydedilir.

Şimdi a), b) ve c) 'yi kullanarak Schoof Algoritması'nın çalışma zamanını saptayalım. a) 'dan, sadece $O(log\ q)$'den küçük olan ℓ asal sayılarını kullanmamız gerekmektedir. Burada $O(\log q/\log\log q)$ asal sayılar öyle ki, Döngü-A, 2.adım-

9.adım arası kaç kere çalışır. Daha sonra, her seferinde Döngü-A içindeki Döngü-n 3.adım-5.adım arası $\ell = O(log\, q)$ kere çalışır.

Ayrıca, $\ell = O(log\, q)$ olduğundan iddia b) 'ye göre R_ℓ'de temel işlemler $O((\log q)^6)$ bit operasyonu tutuyor. Adım 4.'teki $[n](x^q, y^q)$ değeri, bir önceki $[n-1](x^q, y^q)$ değeri kullanılarak $O(1)$ operasyonda hesaplanabilir.

Buradan, Schoof Algoritma'sı için gerekli olan tüm bit operasyonu sayısı;

$$\underbrace{O(log\, q)}_{\text{Döngü-A}} . \underbrace{O(log\, q)}_{\text{Döngü-}n} . \underbrace{O((\log q)^6)}_{\substack{R_\ell\text{'deki her bir} \\ \text{Bit operasyonu}}} = O((\log q)^8).$$

Bununla birlikte Schoof Algoritması'nın $\#E(\mathbb{F}_q)$'yı hesaplayan polinom zamanlı bir algoritma olduğunu ispatlamayı bitirmiş olduk [3].

Schoof Algoritması'nın en zaman alıcı kısmı, $2\ell^2$ dereceli bir $\mathbb{F}_q$ genişlemesi olan, R_ℓ halkasında hesaplamalar yapıyor olmasıdır. Hatta ℓ'ye bağlı olması bile, tahmini süreyi $log\, q$ 'ye yakınlaştırır. Çünkü, eğer q makul ölçülerde büyük ise bu durumda, R_ℓ halkasının boyutu da ℓ'ye ve $\mathbb{F}_q$ 'ya bağlı olarak büyür.

7.2.2 Örnek:

Kriptografi için alışılmış bir değer olarak $q \approx 2^{256}$ alalım.

$$\prod_{\ell \leq 103} \ell \approx 2^{133.14} > 4\sqrt{q} = 2^{130}$$

Yani Schoof Algoritması tarafından kullanılan en büyük asal sayı $\ell = 103$. $\mathbb{F}_q[x]/(\psi_\ell(x))$ 'nin bir elemanı, $103^2 \approx 2^{13.4}$ boyutlu bir $\mathbb{F}_q$-vektör tarafından temsil edilir. Ek olarak $\mathbb{F}_q$'nun her bir elemanı 256-bit'lik bir sayıdır. Yani $\mathbb{F}_q[x]/(\psi_\ell(x))$ 'nin elemanları yaklaşık olarak 2^{22}-bit, o da 16 KB'dan fazla demek

oluyor. Günümüz bilgisayarları, bu elemanları 16 KB olan halkalarla çalışabilecek yeteneğe sahip olmalarına rağmen, bu tür hesaplamalar gereksiz yere çok fazla zaman harcamamıza neden olurlar [3].

8. SONUÇ

Eliptik eğri kriptografisi, bu güne kadar kullanılan ilk nesil ortak anahtar tekniklerine(RSA ve Diffie-Helman) kıyasla çok daha güvenilirdir ve daha etkili bir performans sağlar.

Eğer güvenilirliği arttırmak için üzerinde çalışılacak, geniş bir nokta sayısına sahip bir eliptik eğri elde etmek isteniliyorsa kullanılabilecek yöntemler aşağıdaki gibidir:

- Rasgele bir eliptik eğri seçip nokta-sayım algoritmalarından yararlanmak, örneğin, Schoof algoritması veya SEA algoritması.
- Nokta sayısı kolayca hesaplanabilen bir eliptik eğri ailesinden bir eliptik eğri seçmek. Örneğin, Koblitz eğrileri.
- Nokta sayısı olarak bir değer belirleyip, bu sayıyla kompleks çarpım metoduyla bir eğri üretmek.

Bu çalışmada, sadece Schoof algoritması üzerinde durduldu. Tez boyunca bahsedildiği gibi Schoof algoritması, polinom zamanlı çalışan bir algoritma oluşu ve kullandığı farklı matematiksel yöntemlerle günümüzde bile hala kendinden söz ettiren bir algoritmadır. Daha sonraki yıllarda, Atkin ve Elkies, Schoof'un metodunu da kullanarak SEA(Schoof-Elkies-Atkin) metodunu geliştirmişlerdir. Böylelikle eliptik eğri kriptografisinde günümüzde birkaç yüz karaktere sahip bir q bile çok başarılı bir şekilde kullanılabiliyor. Bu tezde biz sadece Schoof'un algoritması üzerinde durduk. Atkin ve Elkies 'in metotlarıyla geliştirilmiş ve adını yaratıcılarının soyadlarının ilk harflerinden alan SEA algoritmasının detayları için [6] incelenilmesi gereken bir kaynaktır.

9. KAYNAKLAR

[1] Washington, L.C., *Elliptic Curves: Number Theory and Cryptography,* Chapman&Hall/CRC, United States of America, (2003).

[2] Silverman, J.H., Tate, J., *Rational Points on Elliptic Curves,* Springer-Verlag, New York, (1992).

[3] Silverman, J.H., *The Arithmetic of Elliptic Curves,* Springer-Verlag, New York, (2008).

[4] Koblitz, N., *Algebraic Aspects of Cryptography,* Springer-Verlag, New York, (1998).

[5] Schoof, R., *"Elliptic curves over finite fields and the computation of square roots mod p",* Math. Comp., 44(170), 483-494, (1985).

[6] Schoof, R., "*Counting points on elliptic curves over finite fields",* J. Théorie des Nombres de Bordeaux, 7, 219-256, (1995).

[7] Apostol, T.M., *Introduction to Analytic Number Theory,* Springer-Verlag, New York, (1976).

[8] Enge, A., *Elliptic Curves and Their Applications to Cryptography,* Kluwer Academic Publishers, Boston, (1999).

EKLER

10. EKLER

10.1 SANS(E) Fonksiyonu

```
function SANS(E);

    Fq := BaseField(E);
    q := #Fq;
    print "Our prime number q is",q;
    print "So we have",Fq;
    A := Coefficients(E)[4];
    B := Coefficients(E)[5];
    print "We have",E;


    //to find primes


    tableofprimes := Findm2(q);
    print "We will have to compute t mod the following primes:", tableofprimes;

    // schoof mod 2


    print "Computing trace mod 2";

    tableoftraces := [];
    tableoftraces cat:=[Schoofmod2(q,A,B)];
```

```
//to find division polynomial for m


print "Computing division polynomials...";

R<x> := Parent(fdivpol(-1,q,A,B));

f := [ R ! fdivpol(m,q,A,B) : m in [(-1)..(tableofprimes[#tableofprimes]+1)]];


for i in [2..(#tableofprimes)] do


  ell := tableofprimes[i];
  Fell := FiniteField(ell);
  k := q mod ell;

  print "Computing trace mod", ell;


  S<X> := quo< R | [ f[ell+2] ] >;
  F := [ S ! f[i] : i in [1..#f] ];

  if IsEven(k) then
    bigpol := (X^(q^2) - X)*F[k+2]^2*(X^3 + A*X + B) + F[k+1]*F[k+3];
    bigpol := R ! bigpol;
  else
    bigpol := (X^(q^2) - X)*F[k+2]^2 + F[k+1]*F[k+3]*(X^3 + A*X + B);
    bigpol := R ! bigpol;
  end if;
  gcd := GCD(bigpol,f[ell+2]);

// SCHOOF'S CASE 1.
```

```
if gcd ne 1 then
 if not IsSquare(Fell ! q) then tableoftraces cat:= [0];
 else
  w := Integers() ! Sqrt(Fell ! q);
  if IsEven(w) then
   bigpol := (X^q - X)*F[w+2]^2*(X^3 + A*X + B) + F[w+1]*F[w+3];
   bigpol := R ! bigpol;
  else
   bigpol := (X^q - X)*F[w+2]^2 + F[w+1]*F[w+3]*(X^3 + A*X + B);
   bigpol := R ! bigpol;
  end if;
  gcd := GCD(bigpol,f[ell+2]);
  if gcd eq 1 then tableoftraces cat:= [0];
  else
   if IsOdd(w) then
    bigpol := 4*(X^3 + A*X + B)^((q-1) div 2)*F[w+2]^3 -
F[w+4]*F[w+1]^2 + F[w]*F[w+3]^2;
    bigpol := R ! bigpol;
   else
    bigpol := 4*(X^3 + A*X + B)^((q+3) div 2)*F[w+2]^3 -
F[w+4]*F[w+1]^2 + F[w]*F[w+3]^2;
    bigpol := R ! bigpol;
   end if;
   gcd := GCD(bigpol,f[ell+2]);
   if gcd eq 1 then tableoftraces cat:= [-2*w mod ell]; else tableoftraces
cat:= [2*w mod ell]; end if;
  end if;
 end if;


// SCHOOF'S CASE 2.
 else
 if IsEven(k) then
 // k even
```

```
alphaprime := F[k+4]*F[k+1]^2 - F[k]*F[k+3]^2 - 4*(X^3 + A*X + B)^((q^2 + 3) div 2)*F[k+2]^3;
beta := ((X - X^(q^2))*(X^3 + A*X + B)*F[k+2]^2 - F[k+1]*F[k+3])*4*(X^3 + A*X + B)*F[k+2];
factor1 := (F[k+1]*F[k+3] - (X^3 + A*X + B)*F[k+2]^2*(X^(q^2) + X^q + X))*beta^2 + (X^3 + A*X + B)^2*F[k+2]^2*alphaprime^2;
factor2 := 4*(X^3 + A*X + B)^((q+1) div 2)*(alphaprime*((2*X^(q^2) + X)*beta^2*(X^3 + A*X + B)*F[k+2]^2 - F[k+1]*F[k+3]*beta^2 - (X^3 + A*X + B)^2*F[k+2]^2*alphaprime^2) - (X^3 + A*X + B)^((q^2 + 1) div 2)*beta^3*F[k+2]^2);
tau := 0; success := false;
repeat
tau +:= 1;
 // VERIFY IF tau IS THE GOOD ONE
 if IsEven(tau) then
  bigpol1 := factor1*(X^3 + A*X + B)^q*F[tau+2]^(2*q) + F[tau + 1]^q*F[tau +3]^q*beta^2*(X^3 + A*X + B)*F[k+2]^2;
  bigpol2 := factor2*(X^3 + A*X + B)^((3*q-1) div 2)*F[tau +2]^(3*q) - beta^3*(X^3 + A*X + B)^((q+1) div 2)*F[k+2]^2*(F[tau+4]*F[tau+1]^2 - F[tau]*F[tau+3]^2)^q;
 else
  bigpol1 := factor1*F[tau+2]^(2*q) + (X^3 + A*X + B)^(q+1)*F[tau+1]^q*F[tau+3]^q*beta^2*F[k+2]^2;
  bigpol2 := factor2*F[tau+2]^(3*q) - beta^3*(X^3 + A*X + B)^(q+1)*F[k+2]^2*(F[tau+4]*F[tau+1]^2 - F[tau]*F[tau+3]^2)^q;
 end if;
 if (bigpol1 eq 0) and (bigpol2 eq 0) then success := true; end if;
 print "Now verifying trace:",tau;
until success;
print "Success.";
tableoftraces cat:=[tau];
else
// k odd
```

```
alpha := (X^3 + A*X + B)*F[k+4]*F[k+1]^2 - (X^3 + A*X + B)*F[k]*F[k+3]^2 - 4*(X^3 + A*X + B)^((q^2 + 1) div 2)*F[k+2]^3;
betaprime := 4*F[k+2]*((X - X^(q^2))*F[k+2]^2 - (X^3 + A*X + B)*F[k+1]*F[k+3]);
factor1 := ((X^3 + A*X + B)*F[k+1]*F[k+3] - F[k+2]^2*(X^(q^2) + X^q + X))*(X^3 + A*X + B)*betaprime^2 + F[k+2]^2*alpha^2;
factor2 := 4*(X^3 + A*X + B)^((q-1) div 2)*(alpha*((2*X^(q^2) + X)*(X^3 + A*X + B)*betaprime^2*F[k+2]^2 - (X^3 + A*X + B)^2*F[k+1]*F[k+3]*betaprime^2 - F[k+2]^2*alpha^2) - (X^3 + A*X + B)^((q^2 + 3) div 2)*betaprime^3*F[k+2]^2);
tau := 0; success := false;
repeat
tau +:= 1;
 // VERIFY IF tau IS THE GOOD ONE
 if IsEven(tau) then
  bigpol1 := factor1*(X^3 + A*X + B)^q*F[tau+2]^(2*q) + F[tau+1]^q*F[tau+3]^q*(X^3 + A*X + B)*betaprime^2*F[k+2]^2;
  bigpol2 := factor2*(X^3 + A*X + B)^((3*q+1) div 2)*F[tau+2]^(3*q) - (X^3 + A*X + B)^((q+3) div 2)*betaprime^3*F[k+2]^2*(F[tau+4]*F[tau+1]^2 - F[tau]*F[tau+3]^2)^q;
 else
  bigpol1 := factor1*F[tau+2]^(2*q) + (X^3 + A*X + B)^(q+1)*F[tau+1]^q*F[tau+3]^q*betaprime^2*F[k+2]^2;
  bigpol2 := factor2*F[tau+2]^(3*q) - (X^3 + A*X + B)^(q+1)*betaprime^3*F[k+2]^2*(F[tau+4]*F[tau+1]^2 - F[tau]*F[tau+3]^2)^q;
 end if;
 if (bigpol1 eq 0) and (bigpol2 eq 0) then success := true; end if;
 print "Now verifying trace:",tau;
until success;
print "Success.";
tableoftraces cat:=[tau];
end if;
end if;
end for;
```

```
print "The traces are:", tableoftraces;

print "Chinese Remainder Theorem:";

N := &*tableofprimes;
tableofai:=[];

for i in [1..#tableoftraces] do
  _,_,si:=XGCD(tableofprimes[i],(N div tableofprimes[i]));
  ei:=si*(N div tableofprimes[i]);
  ai:=tableoftraces[i];
  xi:=ai*ei;
  tableofai cat:=[xi];
end for;
x:=&+tableofai;
y:=x mod N;

if y le 2*Sqrt(q) then
  t:=y;
  print "Trace of Elliptic Curve is", t;
else
  t:=y-N;
  print "Trace of Elliptic Curve is", t;
end if;
print "Number of the Rational point on E is q+1-t and t=", t;
return q+1-t;

end function;
```