Elliptikus görbe kriptorendszerek I. rész

Elliptic Curve Cryptography – elméleti háttér

Egy jó ideje egyre több helyen találkozhatunk az ECC betűhármassal, mint egy kriptorendszer megjelölésével. Egyre több előnyéről hallunk: az RSA-nál rövidebb kulcsokkal érhető el ugyanakkora biztonság, gyorsabb a működése, kisebb a memóriaigénye. Mindezek a SmartCard technológia biztonsági eszközeit is az ECC felé fordítják. De mi is ez? *Elliptic Curve Cryptosystem*, bár gondolom ettől most sokan nem lettek okosabbak, nekik (is) szól az a kétrészes cikksorozat, melynek első része következik.

Az elliptikus görbékről egyre gazdagabb irodalommal és egyre több ismerettel bírunk. A Certicom szerint az elliptikus görbéket, mint algebrai és geometriai elemeket az elmúlt 150 évben behatóan tanulmányozták és ezeken a tanulmányokon alapul a gazdag és mély ismeretek tárháza. Tekintve, hogy a Certicom Corporation [**certicom**] az ECC egyik vezéregyénisége, sajnos ez a megállapítás csak féligazság, mert az igaz ugyan, hogy az elliptikus görbék régóta ismertek, azonban kriptográfiai szempontból csak 15 éve vizsgálják őket. Az elliptikus görbék kriptográfiai alkalmazását – egymástól függetlenül – két kutató is javasolta: először *Neal Koblitz* (University of Washington) 1985-ben, majd tőle függetlenül *Victor Miller* (IBM). A jelenleg használt PKI rendszerek (szinte) mindegyike a következő három probléma valamelyikének megoldási nehézségén alapul:

- A faktorizálás problémája – IFP (~1970-től, *integer factorization problem*)
- diszkrét logaritmus – DLP (~1970-től, *discrete logarithm problem*)
- diszkrét logaritmus az elliptikus görbék felett – ECDLP (~1985-től, *elliptic curve discrete logarithm problem*)

Sajnos jelenleg senki sem tudja biztosan, hogy e három probléma elég erős-e, de ennek ellenkezőjét sem bizonyította még senki. Jelenleg csak olyan megoldó-algoritmusokról beszélhetünk, amelyek „napjaink legjobb algoritmusai", de nem biztos, hogy elvileg is a legjobbak. Mindenesetre – figyelembe véve a ma ismert támadási módokat – az IFP megoldására ma ismert legjobb algoritmus messze leköröz a ma ismert legjobb ECDLP megoldó algoritmust.
Az alábbi táblázatban három kriptorendszer általánosan vagy szabvány szerint használt kulcsméreteit láthatjuk. Az egy sorban lévő kulcsméretek közel azonos biztonságot nyújtanak (a NIST ajánlása szerint). Látható, hogy a két legismertebb nyíltkulcsos algoritmus (RSA és ECC) kulcsméretei – azonos biztonság mellett – „köszönőviszonyban" sincsenek egymással...

NIST javaslata az egyes kriptorendszerek kulcshosszának összehasonlítására:			
ECC modulus	AES	RSA modulus	RSA:ECC
112	56	512	5:1
161	80	1024	6:1
256	128	3072	12:1
384	192	7680	20:1
512	256	15630	30:1

Az a tény, hogy az ECC sokkal kisebb kulcsmérettel is biztonságos, a következőket vonja maga után:

- gyorsabb algoritmusok (Azonban az RSA-aláírás ellenőrzése és az üzenet titkosítása szinte verhetetlen, ha kis nyilvános exponenst használunk, például *e=65537*!)
- kevesebb továbbítandó és kezelendő adat
- kisebb tárigény
- kisebb tanúsítványok.

Mindezen vélt vagy valós előnyök miatt nem csak a kutatók foglalkoznak az elliptikus görbékkel, hanem az üzleti élet számára szolgáltatásokat nyújtó cégek is. Ez eggyel több ok, hogy mi is megismerjük legalább az alapvető fogalmakat és módszereket.
A következő alfejezetekben az elliptikus görbéket először a valós számok halmazán definiáljuk és megnézzük az értelmezett műveleteket, a műveletek származtatását, esetenként geometriai jelentését is. Ne ijedjünk meg a sok ábrától és képlettől, jóval egyszerűbb, mint amilyennek elsőre tűnik! (Ettől függetlenül sajnos igaz, hogy az ECC matematikailag jóval bonyolultabb, mint az RSA vagy a Diffie-Hellman, ezért sokkal nehezebb megérteni és elmagyarázni vagy bizonyítani, igazolni a felhasználók felé... Ha valaki a cikk végére ér, és úgy érzi, hogy egy kukkot sem értett meg belőle, nyugodtan lapozzon vissza és kezdje elölről!)

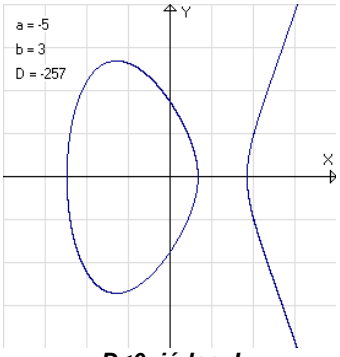
Valós számok halmazán járva – a görbe
Jelöljünk ki a koordinátasíkon egy **P**(x,y) pontot! Ha a koordináták kielégíti az alábbi egyenletet, akkor a **P**(x,y) pont az elliptikus görbe egy pontja:

$$y^2 = x^3 + ax + b$$

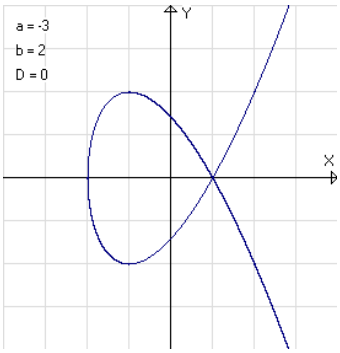
Az elliptikus görbéknek még egy – definíció szerinti – pontjuk van: a végtelenben lévő pont, melyet „**O**" betűvel jelölünk. Nem keverendő össze a valós számok végtelenjével, ami megszámlálhatatlanul sokat jelent, míg az „**O**" pont inkább mérhetetlenül messze van (nem mindig…). És ezekkel a pontokkal fogunk dolgozni! A velük végzett műveletek adják az elliptikus görbéken alapuló kriptorendszerek elemi műveleteinek nagy részét.
Az ***a*** és ***b*** paraméterek változtatásával újabb és újabb görbéket definiálhatunk, de nem mindegyik felel meg nekünk. Vannak olyan (***a,b***) paraméterpárosok, amelyekre igaz, hogy:

$$D = 4a^3 + 27b^2 = 0$$

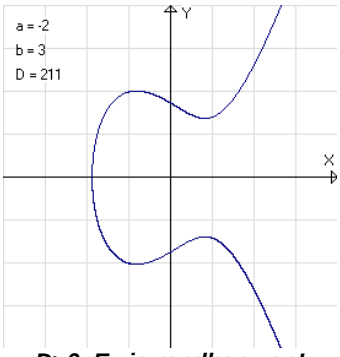
E görbék vagy elmetszik magukat vagy csúcsban végződnek. Most nem használjuk ezeket, nem tekintjük őket elliptikus görbének (szinguláris görbék). Az alábbi három ábrán három görbét láthatunk azokra az esetekre, amikor *D<0* , *D=0* és *D>0*.



D<0, jó lesz!



D=0, ez bizony nem jó!



D>0, Ez is rendben van!

Azt állítottam, hogy a fenti görbék pontjaival fogunk dolgozni, lássuk hát miként?

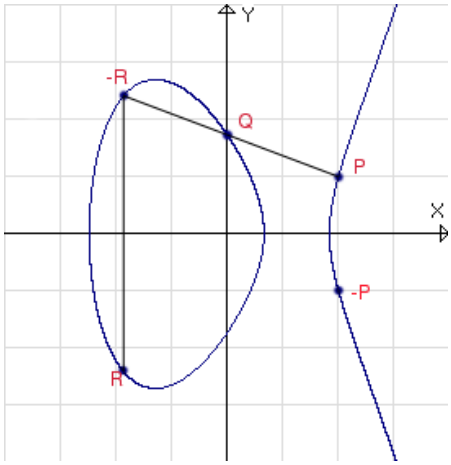
Műveletek – geometriai megközelítésben
A görbék pontjaival mindössze három elvégezhető műveletet fogunk definiálni, lássuk most őket egyesével, részletesen!

Előjelváltás – ellentett képzése
Egy **P(x,y)** pont ellentett párja **R** = **–P** nem más, mint a pont x tengelyre tükrözött képe, amely szintén rajta lesz a görbén: **R(x,-y)**

Összeadás
Legyen a görbe két különböző pontja **P** és **Q**! E két pont összegét jelölje **R**, vagyis **R=P+Q**. A műveletet a következőképpen kell elvégezni:

1. Kössük össze a P és Q pontot egy egyenessel!
2. Az egyenes egy harmadik pontban metszi a görbét, ez a pont lesz **–R**.
3. E pont x tengelyre tükrözött képe az előjelváltás szabálya szerint szintén rajta lesz a görbén és ez lesz az eredmény **R** pont.

A művelet lépései az alábbi ábrán követhetők:



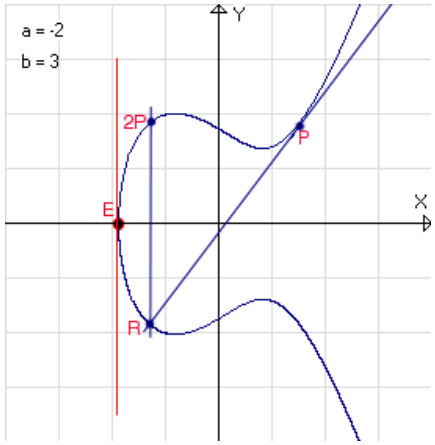
R=P+Q

P + (–P) = ?

Ha eddig egyszerűnek tűnik, akkor most próbáljuk meg **P** és **–P** pontot összeadni! Mivel **P** és **–P** egymás tükörképei az x tengelyre nézve, a rajtuk keresztül húzott egyenes párhuzamos az y tengellyel és sajnos nincs olyan harmadik pont, amiben metszené az elliptikus görbét. Az iménti ábrán látható még egy ilyen pontpáros (–**R** és **R**): az őket összekötő egyenes szintén párhuzamos az y tengellyel és hiába hosszabbítanánk meg bármelyik irányba, nem fogja harmadik pontban metszeni a görbét. Majd csak a végtelenben… Ezért **P** + (**–P**) nem számítható ki a fenti módszerrel, de szerencsére van nekünk egy **O** pontunk, amely a végtelenben van. Definíció szerint **P + (–P) = O**, amiből az is következik, hogy elliptikus görbén értelmezve **P + O = P**. Hasonló azonosság a valós számok körében is van: *x * 1 = x*.

2P = ?

Egy pontot az előzőek mintájára adhatunk össze saját magával. Ha **P** és **Q** pontokat végtelenül közel visszük egymáshoz, akkor **P**=**Q** lesz. A **P** és **Q** ponton keresztül húzott egyenes pedig a **P** pontba húzott érintővé válik. A folytatást pedig ismerjük: az érintő metszi valahol a görbét és a metszéspont tükörképe lesz a **P** pont kétszerese.



Egy pont megduplázása

Sajnos azonban itt is van kivétel (lásd az alábbi ábrán **E** pontot): ha a pont az x tengelyen szíveskedik tartózkodni, akkor az érintő függőleges és nem metszi másik pontban a görbét. Ebben az esetben definíció szerint a pont kétszerese a végtelenben van, vagyis 2**E**=**O**. Ezek a pontok – melyeknek y koordinátája zérus – elég furcsán viselkednek, ha 2**E** után kiszámoljuk 3**E**, 4**E**, 5**E** stb. pontokat:

- 2**E** = **O**
- 3**E** = **E**+2**E** = **E**+**O** = **E**
- 4**E** = **E**+3**E** = **E**+**E** = **O**
- 5**E** = **E**+4**E** = **E**+**O** = **E** és így tovább.

Műveletek – algebrai megközelítésben
Az előzőekben az elliptikus görbén értelmezett műveleteket geometriai eszközökkel mutattuk be, de ez nem túl gyakorlatias a digitális számítógépek számára. Kicsit nehézkes érintőket és húrokat húzkodni, x tengelyre tükrözni, de szerencsére a geometriai műveletek eredményét ki is tudjuk számolni...

Előjelváltás – ellentett képzése
Ebben semmi újdonság sincs, ha a pont **P**(x,y), akkor **R = –P**:

$$x_R = x_P$$
$$y_R = -y_P$$

Összeadás
A levezetést mellőzve, csak a végeredményre hivatkozva: ha **P**(x_P,y_P) és **Q**(x_Q,y_Q) nem egymás ellentettje, akkor **R** = **P** + **Q**:

$$s = (y_P - y_Q)/(x_P - x_Q)$$
$$x_R = s^2 - x_P - x_Q$$
$$y_R = s(x_P - x_R) - y_P$$

P + (–P) = ?
Korábban láttuk, hogy ha a két összeadandó pont egymás ellentettje, akkor a rajtuk keresztül húzott egyenes egy függőleges egyenes, ami **O** pontban „metszi" a görbét. Az egyenes függőleges voltát jól jelzi az is, hogy az *s* kiszámításához használt tört nevezőjében zérus van. (Figyeljük meg, hogy *s* a PQ egyenes meredeksége!)

2P = ?
Ha y_P nem zérus, vagyis a duplázandó pont nem az x tengelyen van (lásd előző ábra **E** pontját!), akkor **R** = 2**P**:

$$s = (3x_P^2 + a)/(2y_P)$$
$$x_R = s^2 - 2x_P$$
$$y_R = s(x_P - x_R) - y_P$$

A moduláris aritmetika közbelép – a görbe
A már megismert egyenletünket egészítsük ki egy kicsit! Ne a valós számokon értelmezzük, hanem a moduláris aritmetika szabályai szerint:

$$y^2 \equiv x^3+ax+b \pmod p, \text{ ahol } p \text{ prím}$$

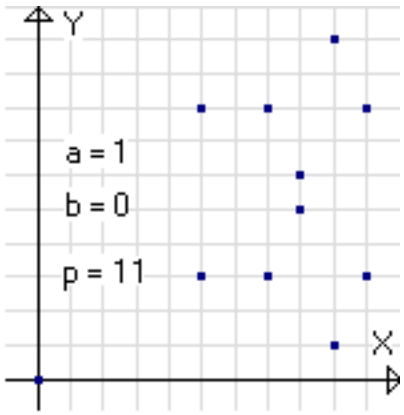
Vagyis ha az egyenlet mindkét oldala ugyanazt a maradékot adja *p*-vel történő osztás után, a **P***(x,y)* pont a görbe pontjai közé tartozik. (Igazság szerint a *p* modulus nem feltétlenül prím, bizonyos feltételek mellett akár összetett szám is lehet, de ez olyan speciális esetnek számít, amelynek megoldása sokkal egyszerűbb, mintha *p* prím lenne. Az ANSI X9.63 szabvány ezért egyenesen kizárja az összetett *p* modulussal definiált görbéket a kriptográfiai alkalmazásokból.) Két további feltétel:

1. a **P** pont mindkét koordinátája kisebb legyen, mint *p-1*.
2. valamint $4a^3 + 27b^2 \bmod p <> 0$.

Néhány gyakorlati indok az áttérés mellet:

- A valós számokkal való számolás lassú és pontatlan. A moduláris aritmetika gyors és pontos, csak egész számokkal dolgozik.
- A „valós" görbének végtelen sok pontja van, a modulárisnak jóval kevesebb.
- A moduláris aritmetikában behatárolható a számok értelmezési tartománya, mert a műveletek operandusa(i) és eredménye mindig $0 \le x \le p-1$ közé esik.
- A moduláris aritmetika alkalmazása megnöveli a megoldások számát.

Ha a ponthalmazt az xy-síkon ábrázoljuk, már nem lesz olyan „szép", mint eddig, de megfigyelhető például, hogy továbbra is szimmetrikus. Igaz, most már nem az x tengelyre. A következő ábra *p=11, a=1 és b=0* paraméterek által kijelölt „görbét" mutatja:



Íme az E: $y^2 \equiv x^3+1x+0 \pmod{11}$ „görbe”

Figyeljük meg, hogy:

- 11 darab pontja van a görbének,
- ebből egy darab az origóban (mert *b=0*),
- 10 darab viszonylag véletlenszerűen, de az Y=5,5-re szimmetrikusan helyezkedik el, ezért
- minden X értékhez továbbra is kettő Y tartozik.

A görbe pontjainak száma most csak véletlenül egyenlő *p*-vel. Azokat a görbéket, amelyek pontjainak száma megegyezik *p*-vel, rendhagyó görbéknek (*anomalous curve*) nevezzük és gyakorlatilag az összes szabvány tiltja használatukat, mert létezik hatékony támadási módszer az ilyen görbét használó ECC-rendszer ellen. A pontok számát kiszámolni nem egyszerű feladat, de egy Hasse nevű úriember tétele szerint a pontok száma $(p+1) \pm 2\sqrt{p}$ között van. Egy pontra jellemző még egy szám, amelyet a pont rendjének hívunk: hányszor tudjuk összeadni saját magával, mielőtt ***O***-ba érkezünk? Vagyis ha $n\boldsymbol{P}=\boldsymbol{O}$, akkor *n* a pont rendje. Ha a pont rendje megegyezik a görbe pontjainak számával, a pontot generátorpontnak hívjuk.

Műveletek a „görbe” pontjaival
Lássuk, hogyan kell értelmezni a már jól megismert művelteinket moduloaritmetikával! Szerencsére nem sok újdonságot forgunk látni!

Előjelváltás – ellentett képzése
Ha visszaemlékszünk a valós számokon értelmezett görbére, ott egy **P**(x,y) ellentettje az (x,-y) pont volt. Most sincs ez másként, csak modulárisan kell számolnunk: (x, -y mod p), ami nem más, mint: (x, p-y mod p). Ha megnézzük a fenti példa megoldásait, láthatjuk, hogy az egymással szemben lévő pontok y koordinátáinak összege mindig *p=11*.

Például (5,3) és (5,8) → 3+8 = 11

Tehát egy **R** = **–P** pont kiszámolása:

$$x_R = x_P$$
$$y_R = -y_P \bmod p.$$

Összeadás
Kicsit nehézkes most olyat értelmezni, hogy „kössünk össze” két pontot és keressük meg a harmadik metszéspontot, főleg hogyan „húzzunk érintőt”? Ezért a korábbi algebrai eredményeket vesszük át a moduláris aritmetika szabályai szerint:

$$s = (y_P - y_Q)/(x_P - x_Q) \bmod p =$$
$$\lightning\ (y_P - y_Q)(x_P - x_Q)^{-1} \bmod p$$
$$x_R = s^2 - x_P - x_Q \bmod p$$
$$y_R = s(x_P - x_R) - y_P \bmod p$$

P + (–P) = ?
Ez továbbra sem változik: **P + (–P) = O**

2P = ?
Ha y_P nem zérus, vagyis a duplázandó pont nem azx tengelyen van, akkor R = 2P:

$s = (3x_P^2 + a)/(2y_P) \bmod p = (3x_P^2 + a)(2y_P)^{-1} \bmod p$

$x_R = s^2 - 2x_P \bmod p$

$y_R = s(x_P - x_R) - y_P \bmod p$

A probléma: diszkrét logaritmus

Minden kriptorendszer alapja egy olyan probléma, amit *gyakorlatilag* lehetetlen megoldani. Az ECC-nek is ilyen probléma adja a biztonságát, mégpedig a „diszkrét logaritmus elliptikus görbék felett" kiszámolásának problémája, röviden: ECDLP. (1991-ben néhány kutató elkészítette az RSA algoritmus elliptikus görbéken alapuló változatát, de néhány évvel később többen is megmutatták, hogy az elliptikus RSA-nak (*ECC-like RSA*) nincs számottevő előnye a hagyományos RSA-val szemben. Az ECRSA problémája egyébként továbbra is a faktorizálás maradt.)

Eddig lényegében két műveletet definiáltunk a görbén: pontok összeadása és egy pont duplázása. Egy pont sorozatos összeadásával (**R**, **R**+**R**, 2**R**+**R**, 3**R**+**R**) tulajdonképpen már szorozni is tudunk. Az így képzett ***Q***=*n****R*** pontot a pont *skalár szorzatának* nevezzük, de *n* meghatározása a szorzat alapján nem egyszerű feladat főként, ha a görbét *mod p* felett értelmezzük.

Legyen egy elliptikus görbe egyenlete:

$y^2 = x^3 + 9x + 17 \bmod 23$

Mennyi az R=(16,5) alapú diszkrét logaritmusa Q = (4,5) pontnak? (Q=nR)

Első megközelítésben *n* kiszámolásához addig adogassuk össze a R pontot önmagával, amíg meg nem kapjuk Q-t:

1R=(16,5) 2R=(20,20) 3R=(14,14) 4R=(19,20)

5R=(13,10) 6R=(7,3) 7R=(8,7) 8R=(12,17)

9R=(4,5)

Mivel Q(4,5)=9R, ezért a Q pont R alapú diszkrét logaritmusa mod 23 felett: 9.

Ha most valaki azt mondja, hogy a logaritmus a hatványozás inverze és nem a szorzásé (főleg nem az összeadásé), akkor annak maximálisan igaza van. De...

1. A racionális számokon értelmezett „hagyományos" logaritmus pontosan ugyanarról szól, mint például a *Diffie-Hellman* kulcscsere megoldása. Ha egy *g* számot *x* alkalommal összeszorzunk önmagával ($a=g^x$), akkor az eredményből és *g* ismeretében: $x=\log_g a \bmod p$, ha létezik. Ez a DLP.
2. Az ECDLP esetén adott **P** és **Q** pont a *mod p* felett értelmezett elliptikus görbén. Feladat: megkeresni *x*-et úgy, hogy *x***P**=**Q** legyen, ha létezik ilyen szám.

Úgy tűnik a két probléma jelentősen eltér egymástól, hiszen eleve más műveletekről szólnak, az elsőben sorozatos *szorzás* van, a másodikban sorozatos *összeadás*. Azonban nálam hozzáértőbb matematikusok azt mondják, hogy a DLP szorzása és a ECDLP összeadása absztrakt módon ugyanaz. Általánosságban nem is szorzásnak meg összeadásnak hívják a használt műveleteket, hanem egyszerűen csoportműveleteknek (*group operations*). Egy mérnöki szemléletű embernek ez hajmeresztőnek tűnik, ezért a „békesség kedvéért" ne keressünk most logikát az elnevezésekben, hanem vegyük tudomásul: az elliptikus görbéken értelmezett sorozatos összeadások (=szorzás) inverzét logaritmusnak hívjuk. Pont.

S mindezt mire lehet használni? Erről lesz szó a második részben!

(Folyt. köv.)

Virasztó Tamás
wacher@westel900.net
http://wacher.fpn.hu