Elliptikus görbén alapuló titkosírás

Csirmaz László, CEU

A nyilvános kulcsú titkosírás hátterében egy-egy matematikai probléma áll, melyről azt tételezzük fel, hogy nehéz megoldani. A *Ron L. Rivest*, *Adi Shamir* és *Leon Adleman* nevéhez fűződő RSA módszer azt használja, hogy míg viszonylag egyszerű eldönteni, hogy egy több száz jegyű szám összetett-e vagy sem, addig prímtényezőire bontani már reménytelenül nehéz feladat. *Whitheld Diffie* és *M. E. Hellmann* nevét viselő módszer hátterében az úgynevezett *diszkrét logaritmus* probléma áll: adott az alap és a hatványozás eredménye, keressük meg a kitevőt. A hatványozás természetesen ismételt szorzás, viszont a szorzást nem a szokásos, általános iskolában tanult módon kell elvégezni, hanem maradékosan, valamilyen előre meghatározott prímszám modulussal. A problémát általánosan is meg lehet fogalmazni, a következőképpen: definiáljunk egy szorzásnak nevezett műveletet sok (tipikusan 10^{100}) elem között. Választunk egy g alapot, egy n kitevőt, és kiszámítjuk a $y=g^n$ hatványt. A feladat y és g ismeretében megkeresni az n kitevőt. Mivel az n kitevő is 100 jegyű szám, az y hatvány kiszámítása sem lehetséges ismételt szorzással. Ezért még feltesszük, hogy a definiált művelet az adott elemeken *csoportot* alkot. Ilyenkor a hatványozást jóval gyorsabban el tudjuk végezni: a g generátor elemet ismételten négyzetre emeljük (ehhez egy-egy szorzást használva), amivel megkapjuk g-nek a 2^0, 2^1, 2^2, 2^3, stb kitevős hatványait. Az n kitevőt kettes számrendszerben írjuk fel, és a jegyeinek megfelelő hatványokat szorozzuk össze. Ezzel a szorzások számát még száz jegyű kitevő esetén is 1000 alatt tudjuk tartani.

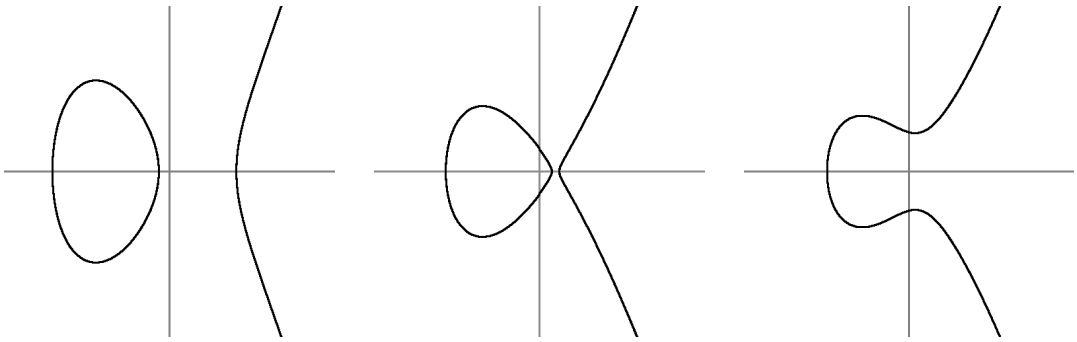
Diszkrét Logaritmus probléma

Adott egy G csoportban a g generáló valamint egy y elem. Keressük meg az y rendjét, vagyis azt n kitevőt, amire a csoportban $g^n=y$.

Diffie és Hellmann ismerték fel, hogy véges csoportokon a diszkrét logaritmus nehéz probléma, vagyis a kitevőt az alap és a hatvány értékéből nem lehet gyorsan visszakeresni. (Természetesen nem ez a helyzet a valós számok esetében, ekkor a kitevőt – vagyis y-nak g alapú logaritmusát – gyorsan ki tudjuk számítani.) Különböző speciális csoportok esetén más és más, igen bonyolult és mély matematikai eszközöket használó algoritmusokat fejlesztettek ki. Tetszőleges csoportra működő leggyorsabb ismert algoritmus futásideje a csoport elemszámának négyzetgyökével arányos. Vagyis 10^{100} elemű csoportnál az algoritmus futása 10^{50} körüli műveletet igényel, amit összevetve a világegyetem korára becsült becsült 10^{22} évvel, elég meggyőző érv a probléma nehézségére. Speciális csoportokon – így például egy p prímszámmal való maradékos szorzás esetén is – gyorsabb algoritmusok is ismeretesek. Abban az esetben, mikor a csoport éppen a modulo p *összeadás*, g pedig az 1 szám, igazán nincs is feladat, hiszen az eredmény és a kitevő (jelen esetben a szorzótényező, hiszen most a művelet összeadás) egy és ugyanaz. Ez is mutatja, hogy kriptorendszer biztonsága szempontjából igen fontos a felhasznált csoport megfelelő megválasztása.

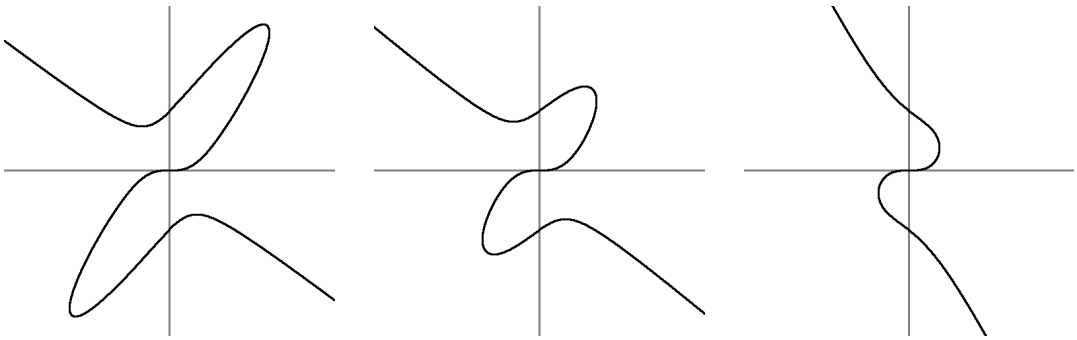
Kriptográfiai rendszerekben két típusú csoportot szokás használni. Az egyik a az 1, 2, …, p-1 számokon a modulo p maradékos szorzás alkotta csoport. Ezekre a csoportokra ismeretes a fent említett $\sqrt{p}$ idejű általános algoritmusnál gyorsabb eljárás is. Ha a p prímszám ezen kívül még speciális alakú is, például p-1-nek (vagy p+1-nek) mindegyik prímosztója legfeljebb 10 jegyű, akkor a diszkrét logaritmus problémát emberi idő alatt (néhány hónap vagy év) is meg lehet oldani. A csoportoknak egy másik gazdag, és egyre többet használt forrása az *elliptikus görbék*. Általános elliptikus görbékből adódó csoportok esetén a diszkrét logaritmus problémára nem ismeretes a csoport méretének gyökénél kevesebb műveletet igénylő algoritmus. Természetesen vannak kifejezetten rossz, illetve "nem ajánlott" görbék, viszont elliptikus görbékből jóval több van, mint természetes számokból, nagyobb a választási lehetőség, kisebb esélyünk van arra, hogy egy rossz csoportba botlunk.

Az elliptikus görbék a középiskolából jól ismert kör, ellipszis, parabola, hiperbola általánosításai. Míg például az egységkör a síknak azokból az (x,y) koordinátájú pontjaiból áll, amikre $x^2+y^2=1$, addig például az $y^2=x^3+ax+b$ egyenletet kielégít• pontok egy harmadrend• algebrai görbét határoznak meg.



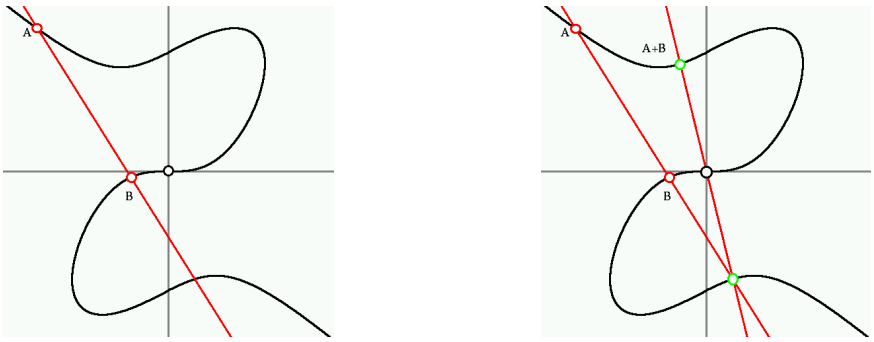
$y^2=x^3+ax+b$ egyenletű elliptikus görbék különböző paraméterekkel

Az ábrán különböző a és b paraméterek mellett mutatjuk be magát a görbét. A görbe maga egy vagy két részből áll, a jobb oldali ágai egyre meredekebben tartanak a végtelenbe. Azokat, melyek elmetszik saját magukat, vagy csúcsban végződnek, mint az a=b=0 esetben, *szinguláris görbéknek* nevezzük. A függőleges egyeneseket kivéve minden más egyenes vagy egy vagy három pontban metszi a görbét, ezért a függőleges egyenesek ideális (végtelen távoli) pontját is hozzá szokás venni a görbéhez, hogy azok se legyenek kivételek. A görbének ezt a pontját egy projektív transzformációval az origóba vive a görbe formája is megváltozik:



$x^3=y+axy^2+by^3$ egyenletű duális görbék, az inflexiós pont az origó

Egyetlen olyan pont van, ahol az érintő harmadrendben simul a görbéhez, ezt a görbe *inflexiós pontjának* nevezzük. Az $y^2=x^3+ax+b$ egyenlet• görbéknél ez a függ•leges egyenesek ideális pontja, az $x^3=y+axy^2+by^3$ görbéknél viszont az origó. Az ilyen harmadrendű kifejezések által meghatározott görbékkel, és általában egy kétváltozós polinomot kielégítő (x, y) pontok tulajdonságaival foglalkozik a matematika egyik legszebb, de ugyanakkor legnehezebb ága, az algebrai geometria. *Elliptikus görbék* a harmadrendű kifejezés által definiált nem szinguláris alakzatok. Ezek számtalan érdekes tulajdonságai közül azt használjuk fel, hogy pontjain definiálható egy művelet, amivel az elliptikus görbe pontjai csoportot alkotnak. Ezt a csoportot hívjuk *elliptikus csoportnak*. A műveletet szokás szerint összeadásként értelmezzük, vagyis a görbe pontjait *összeadhatjuk*. Nullelemnek, vagyis amihez a görbe bármely más elemét adva saját magát kapjuk vissza, a formulák egyszerűsítése érdekében a görbe inflexiós pontját választjuk.



A görbe *A* és *B* pontjainak összege

Az ábrán mutatjuk be, hogyan számítjuk ki a görbe *A* és *B* pontjainak az összegét. Legyen *O* a görbe inflexiós pontja, ez az ábrán a koordinátarendszer középpontja. Tetszőleges egyenes a görbét egy vagy három pontban metszi, ez következik abból, hogy egy harmadfokú egyenletnek vagy egy, vagy három valós gyöke van. Ezért ha egy egyenesnek és a görbének van két közös pontja, akkor van egy harmadik is. (Speciálisan ez egybeeshet valamelyik ponttal is, ha éppen érintőről van szó.) Legyen tehát *A* és *B* a görbe két pontja. Kössük össze *A*-t és *B*-t egy egyenessel. Ez az egyenes a görbét még egy pontban metszi, ezt a metszéspontot kössük össze az *O* ponttal. Ez utóbbi egyenesnek és a görbének harmadik közös pontja lesz az *A*+*B* pont. Ha az *A*+*A* összegre vagyunk kíváncsiak, akkor természetesen az *A*-beli érintőt kell használni, az érintő és a görbe metszéspontját kell *O*-val összekötni, és a harmadik metszéspontot választani.

Mivel az *A*-t és *B*-t összekötő egyenes ugyanaz, mint a *B*-t az *A*-val összekötő egyenes, azért ez a művelet *kommutatív*, vagyis $A+B=B+A$. Az is látszik, hogy az inflexiós pontot bármely más ponthoz hozzáadva az nem változik: $A+O=O+A=A$. Ez az összefüggés még az *O* inflexiós pontra is igaz, ugyanis az *O*-beli érintő a görbét harmadszor is *O*-ban metszi (az *O*-beli érintő harmadrendben érint). Az *A* ellentettjét, vagyis a $(-A)$-t úgy kaphatjuk meg, hogy *A*-t összekötjük *O*-val, ez az egyenes metszi ki $(-A)$-t a görbéből; erre a pontra persze $A+(-A)=O$.

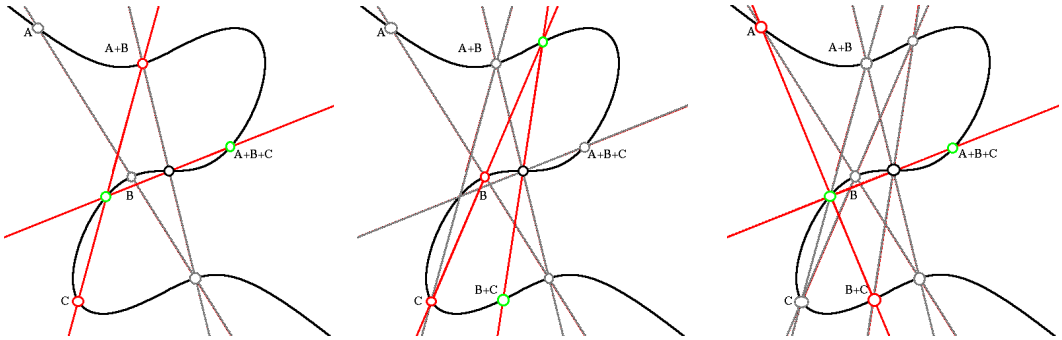
A görbénk szerencsés paraméterezése és *O* megválasztása miatt $(-A)$ nem más, mint *A*-nak *O*-ra való tükörképe, ezt tehát *A* koordinátáiból könnyen számíthatjuk: mindkét koordinátának kell venni a (-1)-szeresét. Nem ilyen egyszerű a helyzet az $(A+B)$ koordinátáinak számításával. Ha *A* az koordinátái (x_1, y_1), a *B* koordinátái pedig $(x_2,$

y_2) , akkor (A+B)-nek x és y koordinátáit az alábbi, egyszerűnek semmiképpen nem mondható képletek alapján számíthatjuk (a formulákat a Maple program állította elő):

$$x := \frac{a x_1^2 y_2^2 + 3 b x_1 y_1 y_2^2 - a y_1^2 x_2^2 - 3 b y_1^2 x_2 y_2 + x_1 y_1 + 2 x_1 y_2 - 2 y_1 x_2 - x_2 y_2}{3 x_1^2 x_2 - 2 a x_1 y_1 y_2 - 3 x_1 x_2^2 + a x_1 y_2^2 - a y_1^2 x_2 - 3 b y_1^2 y_2 + 2 a y_1 x_2 y_2 + 3 b y_1 y_2^2 - y_1 + y_2}$$

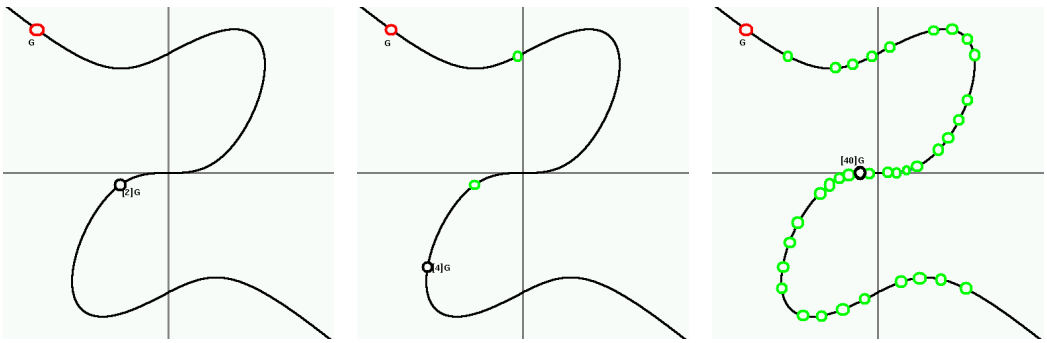
$$y := \frac{3 x_1^2 x_2 y_2 - 3 x_1 y_1 x_2^2 - a x_1 y_1 y_2^2 + a y_1^2 x_2 y_2 + y_1^2 - y_2^2}{3 x_1^2 x_2 - 2 a x_1 y_1 y_2 - 3 x_1 x_2^2 + a x_1 y_2^2 - a y_1^2 x_2 - 3 b y_1^2 y_2 + 2 a y_1 x_2 y_2 + 3 b y_1 y_2^2 - y_1 + y_2}$$

Ráadásul ezek nem használhatók abban az esetben, ha A és B egybeesik, mert ilyenkor mind a számláló mind a nevező nulla. Helyette másik képletet kell alkalmaznunk (azt nem mutatjuk itt be).



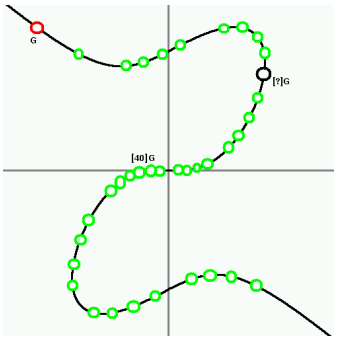
Az összeadás asszociatív: $(A + B) + C = A + (B + C)$

Ahhoz, hogy íly módon tényleg csoportot kapjuk, még meg kell mutatni, hogy a művelet *asszociatív*, vagyis tetszőleges A, B és C pontokra $(A + B) + C = A + (B + C)$. Az állítást az ábra illusztrálja. Először hozzáadjuk C-t az (A+B) összeghez (bal oldali ábra). Azután kiszámítjuk (B+C)-t (középső ábra), és ezt adjuk A-hoz. Hogy az eredmény mind a kétszer ugyanaz, következik például abból, hogy az A+B+C koordinátáit kétféleképpen kiszámítva ugyanazokat a formulákat kapjuk (nagyszerű feladat egy szimbolikus algebrai program számára). Vagy következik abból a tételből is, amelyet a harmadik ábráról olvashatunk le, és ami a projektív geometriából ismert Papposz tétel általánosítása. Felejtkezzünk el az A+B+C pontról és a rajta átmenő egyeneséről. Az ábrán marad hat egyenes, melyek kilenc pontban metszik egymást. Ha e kilenc pont közül nyolc rajta van a görbén, akkor a kilencedik is rajta van. Ez a *kilenc asszociált pont tétele* néven ismert állítás (pontosabban annak is csak egy speciális esete).



A G generáló elem és többszörösei

Ezzel megkaptuk a titkosításhoz szükséges csoportot. Amit még választanunk kell, egy G generátor (a görbe tetszőleges pontja), aminek többszöröseit használjuk fel. Szokás szerint ha G-t n-szer adjuk önmagához, az eredményt $[n]G$ jelzi. Az ábrán láthatjuk ahogyan a G, $[2]G$, $[3]G$, stb. értékek egyre jobban beterítik a görbét. Esetünkben a *diszkrét logaritmus probléma* a következő: adott a görbe paramétereivel, a G generáló pont koordinátáival, továbbá a görbének egy Q pontja. Keressük meg azt az n egész számot (ami lehet negatív is), amire $Q=[n]G$. Az elliptikus görbéken alapuló nyilvános kulcsú titkosírás, szokásos angol rövidítéssel *ECC*, éppen ennek a problémának a nehézségét használja ki.



Diszkrét logaritmus

Természetesen a gyakorlati megvalósításban a harmadfokú görbét nem a valós számok fölött használjuk, hanem valamilyen *véges testet* választunk erre a célra. Mivel egy véges test elemászáma mindig valamilyen prímszámhatvány, gyakorlatilag kétféle test jöhet szóba. Az egyiknél a test elemszáma egy $2^n - 1$ alakú prímszám, a másiknál pedig kettőhatvány. A kitevőt mindkét esetben 200 körül szokás választani, így a test egy elemének felírásához ennyi bitre van szükség, a görbe egy pontját pedig két koordináta határozza meg, tehát ahhoz kétszer ennyi bitet kell tárolni.

További problémát okoz, hogy ha az alaptest elemszáma kettőhatvány, akkor az elliptikus görbe algebrai alakját nem lehet a szokásos $y^2 = x^3 + ax + b$ formára hozni, hanem csak az úgynevezett *Weierstrass* formára: $y^2 + ay = x^3 + bx^2 + cxy + dx + e$. A minél egyszerűbb alakra azért van szükség, hogy egy–egy elliptikus művelet végrehajtása minél gyorsabb legyen. Ahhoz, hogy két pont összegét meghatározzuk, jónéhány szorzást, összeadást, és ráadásul két osztást is el kell végeznünk az alaptest számain. Ráadásul egy kettőhatvány elemű testben két elem *összege* nem más, mint ezek bitenkénti mod 2 összege (vagyis a két bitsorozat XOR-ja), vagyis gyorsan számítható, addig két elem *szorzatát* már csak bonyolult módon (tipikusan polinomok szorzatát kell maradékosan osztani) lehet számítani. Ezért bár az ECC "csak" 200 bites számokat használ, a kódolás/dekódolás ideje nem kevesebb, mint az ötször több bitet használó RSA vagy Diffie-Hellmann módszereknek. A műveleti igény csökkentésére több trükköt vetettek be, például a számlálót és a nevezőt külön-külön tárolják, és csak a számítás legvégén osztják el egymással. Ha a görbe egy pontjának ismerjük az x koordinátáját, akkor ehhez két lehetséges y koordináta tartozik, melyek egymás negáltjai. A számításokat el lehet úgy is végezni, hogy csak az x koordinátákkal foglalkozunk, és soha nem mondjuk meg, hogy a két lehetőség közül melyik is az igazi.

További probléma, hogy nagyon sok algoritmusban szükség van a generált csoport *rendjére*, vagyis arra a legkisebb r pozitív egészre, amivel $[r]G = O$. Ennek értékét a modulo p szorzás esetén viszonylag egyszerű megkapni, míg elliptikus görbék esetében ennek meghatározására nem ismeretes polinomiális algoritmus. A legfrisebb kutatások viszonylag gyors algoritmusokat eredményeztek, amivel egy–egy görbe rendjét már kevesebb, mint egy óra alatt meg lehet kapni.

Végül álljon itt a diszkrét logaritmus probléma egy konkrét alkalmazása. Az alábbi digitális aláírás *Claus Schnorr* 1991-ból származó módszerének egy változata. Elsőként az aláíró választ egy elliptikus görbét, azon egy G pontot. Meghatározza a G pont rendjét, vagyis azt a legkisebb pozitív r számot, amire $[r]G = O$. Választ még 0 és r–1 között egy s titkos számot, és kiszámítja a görbének a $P = [s]G$ pontját. Az aláíró

nyilvános kulcsa a következő információkat tartalmazza: a görbét (vagyis annak a paramétereit, előállítási módját), a G generátor pontot, valamint a P-t (illetve ennek koordinátáit).
Az M dokumentum aláírásához szükség van még egy mindenki által ismert *sűrítmény-készítő* algoritmusra is. Angol kifejezéssel az ilyen leképezéseket hash függvényeknek is nevezik. Léteznek minden szempontból kielégítő tulajdonságú hash függvények, egyszerűen feltesszük hogy $H(\ .\ )$ egy mindenki megelégedésére kiválasztott sűrítménykészítő eljárás, ami ráadásul mindig a generátor pont rendjénél kisebb értéket állít elő.
Az M dokumentum aláírása a következőképpen történik. Először is választunk egy v véletlen számot 0 és r–1 között, majd kiszámítjuk a $Q = [v]G$ pontot. Fűzzük hozzá a Q pont koordinátáit az M üzenethez, és számítsuk ki ennek az $a = H(M, Q)$ sűrítményét. Most használjuk ki az aláíró s titkos értékét: meghatározzuk a $v - as$ különbség maradékát r-rel osztva, legyen az eredmény (ami persze nem negatív, és kisebb r-nél) éppen b. Ekkor $as + b$ ugyanakkora maradékot ad r-rel osztva, mint v, azért az elliptikus görbének az $[as+b]G = [as]G + [b]G = [a]P + [b]G$ valamint $[v]G = Q$ pontjai ugyanazok. Az M dokumentum *aláírása* az (a, b) pár.
Az aláírás hitelességének ellenőrzéséhez az M dokumentumot, annak (a, b) aláírását, és az aláíró nyilvános kulcsát használjuk. A görbe leírása, annak G valamint P pontjai ismeretében kiszámítható a görbe $Q' = [a]P + [b]G$ pontja. Számítsuk ki ezek után az $a' = H(M, Q')$ sűrítményt. Az aláírást hitelesnek fogadjuk el, ha az így kapott a' és az aláírásban szereplő a szám megegyezik.
Könnyű látni, hogy egy helyesen aláírt dokumentumot mindig hitelesnek fogunk elfogadni. Ez azon múlik, hogy ilyenkor az ellenőrzéskor számított Q' és az aláíráskor használt Q pont ugyanaz. Ahhoz, hogy az aláírást ne lehessen hamisítani, az kell, hogy a diszkrét logaritmus probléma gyakorlatilag megoldhatatlan legyen, sajnos ezen kívül még a sűrítménykészítő H függvényről is különböző feltételeket kell tennünk. Az aláírás használatára is vannak további megszívlelendő szabályok. Így például a v számnak nem szabad olyat választani, aminek értékét valamekkora biztonsággal meg lehet jósolni, és semmiképpen nem szabad ugyanazt a v-t különböző üzenetek aláírására használni.

Irodalom

[1] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone: *Handbook of applied cryptography*, CRC Press, 1997
[2] R. Crandell, C. Pomerance: *Prime numbers, a computational perspective*, Springer, 2000
[3] *Eric Weisstein's world of mathematics*, http://mathworld.wolfram.com
[4] N. Koblitz, *Elliptic curve cryptosystems*, Math.Comp, 48, 1987