SZTAKI, „Kriptográfia és alkalmazásai” szeminárium
2002. május 21.

Az RSA és az ECC gyakorlati összehasonlítása

Endrődi Csilla
BME MIT Ph.D. hallgató
csilla@mit.bme.hu

Előadásvázlat

- **Nyilvános kulcsú algoritmusok**
  - Létező algoritmusok
  - RSA
  - ECC
- **Az elemzés bemutatása**
  - A mérés bemutatása
  - Az összehasonlítás bemutatása
  - Szoftver implementáció
  - A mérési környezet bemutatása
- **Összehasonlító elemzés**
  - Közös paraméterek felállítása és kulcsgenerálás
  - Kódolás és dekódolás
  - Aláírás és aláírás ellenőrzés
- **Összefoglalás, értékelés**
- **Érdekesség**
- **Irodalmak**

Nyilvános kulcsú algoritmusok

Létezik több nyilvános kulcsú algoritmus.

Alapul szolgáló nehéz problémák:

- Egész számok faktorizációja (IFP)
- Diszkrét logaritmus probléma (DLP)
- Elliptikus görbéken alapuló diszkrét log. pr. (ECDLP)
- Rács redukció
- egyebek…

Funkciók:

- Kulcsegyeztetés
- Rejtjel protokollok
- Digitális aláírás
- Anonimitási protokollok
- stb.

Nyilvános kulcsú kriptográfiai algoritmusok

DLP-n alapuló algoritmusok:

- *Diffie-Hellmann Key Agreement (DH)*
- *Diffie-Hellmann Key Agreement (DH2)*
- *El-Gamal Encryption Scheme*
- *Menezes-Qu-Vanstone (MQV)*
- *Efficient Compact Subgroup Trace Representation (XTR)*
- *Digital Signature Algorithm (DSA)*
- *BlumGoldwasser*
- *Zheng-Seberry*
- *Ballare-Rogaway*
- *Nyberg-Rueppel Signature Scheme*
- *Schnorr*

Nyilvános kulcsú kriptográfiai algoritmusok

IFP-n alapuló algoritmusok:

- RSA
- Rabin-Williams

ECDLP-n alapuló algoritmusok:

- **Elliptic Curve Diffie-Hellmann Key Agreement (ECDH)**
- **Elliptic Curve Menezes-Qu-Vanstone (ECMQV)**
- **Elliptic Curve Integrated Encryption Scheme (ECIEC)**
- **Elliptic Curve Digital Signature Algorithm (ECDSA)**
- **Elliptic Curve Nyberg-Rueppel (ECNR)**

Egyéb:

- LUCDIF
- LUCELG
- LUCRSA
- Merkle-Hellmann
- Chor-Rivest
- NTRU
- McEliece

Az alkalmazások döntő többsége mégis az RSA-t használja.

Nyilvános kulcsú kriptográfiai algoritmusok

RSA algoritmus (1978)

Kulcsgenerálás: p, q prímek; m = p*q
$\Phi(m) = (p-1) * (q-1)$
e tetszőleges: $1 \le e \le \Phi(m)$; $(e, \Phi(m)) = 1$

Nyilvános kulcs: (e, m)

Titkos kulcs: (d, m)

Kódoló függvény: $E(x) = x^e \bmod m$; ! $x < m$;

Dekódoló függvény: $D(y) = y^d \bmod m$

- Negyed évszázada nem sikerült megtörni
- Biztonsága nem bizonyított
- Egyszerű, szép matematikai háttér: modulo aritmetika
- Az RSA Security Inc. szabadalma alatt állt 2000. szept. 20-ig
- A legelterjedtebb használt nyilvános kulcsú algoritmus
- Tipikus paraméterek: modulus: 512-4096 bites; ajánlott: 1024 bites
- Alkalmazott gyorsítás: e kicsire választása (e = 65537 = 10001h)
- A maximális kódolható méretet a modulus határozza meg

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

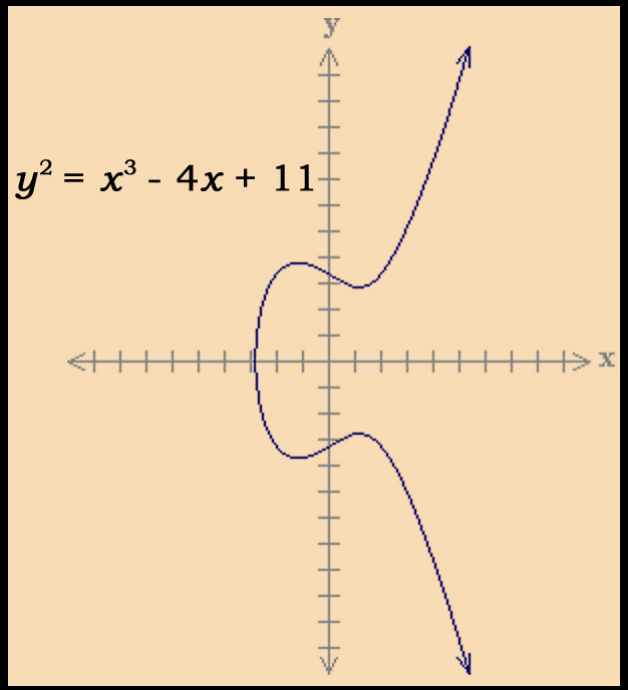
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

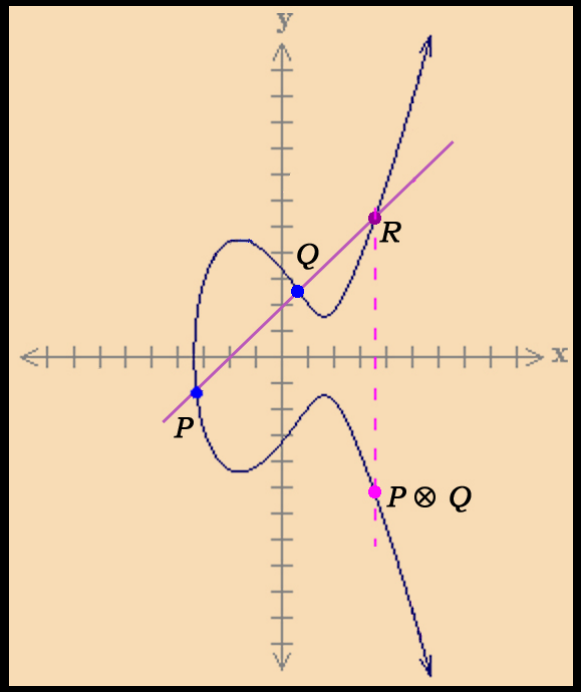
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

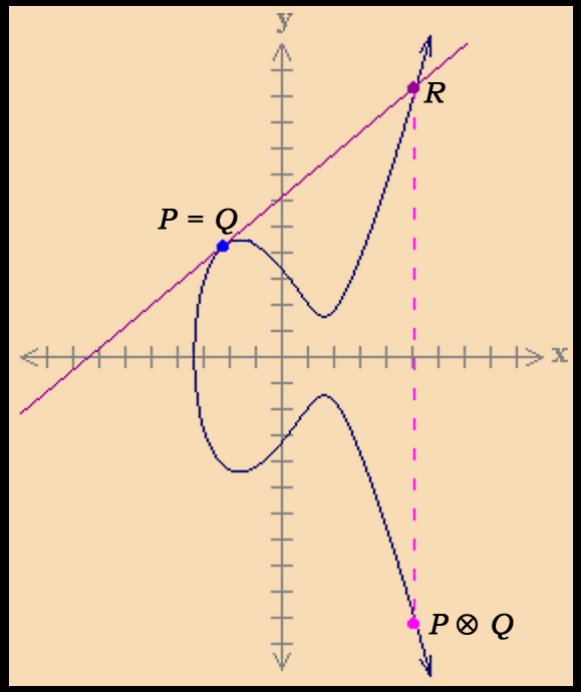
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

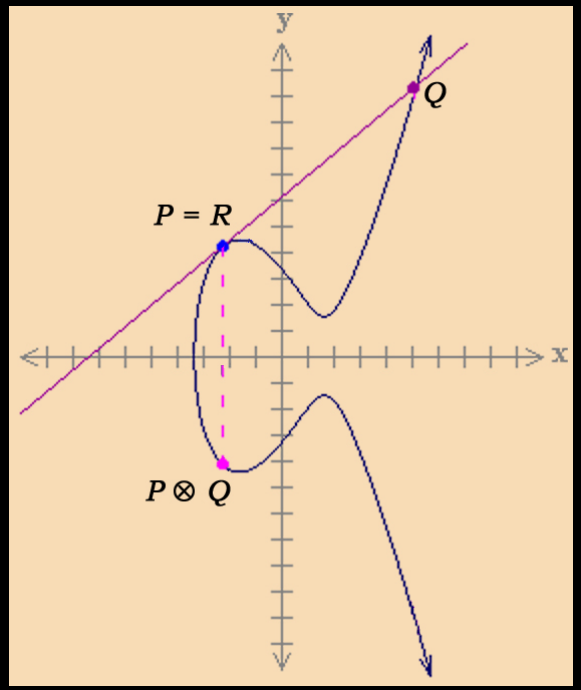
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

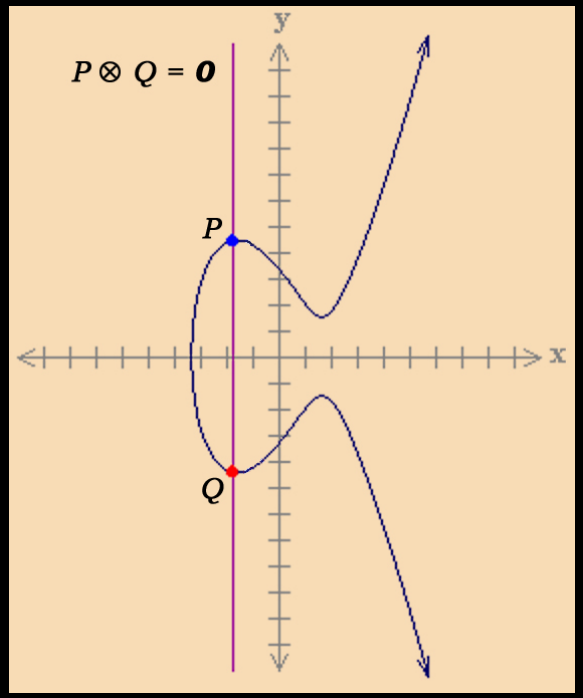
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

Az elliptikus görbék

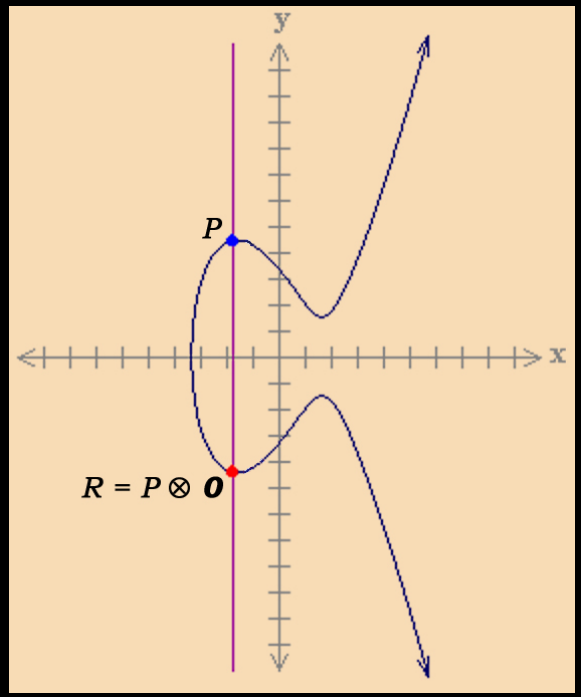
Elliptikus görbe általános alakja: $y^2 + axy + by = x^3 + cx^2 + dx + e$

$x, y, a, b, c, d, e \in \mathbf{F}$

F = valós esetben: $y^2 = x^3 + ax + b$

$x, y, a, b \in \mathbf{R}$

Az elliptikus görbe pontjai az egyenletet kielégítő (x, y) pontok: **E(F)**.



A görbe pontjai csoportot alkotnak egy megfelelően választott művelettel.

A **0** is tagja a csoportnak.

A **0** a neutrális elem.

Kriptgráfiai jelentősége a **véges testek** feletti elliptikus görbéknek van.

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

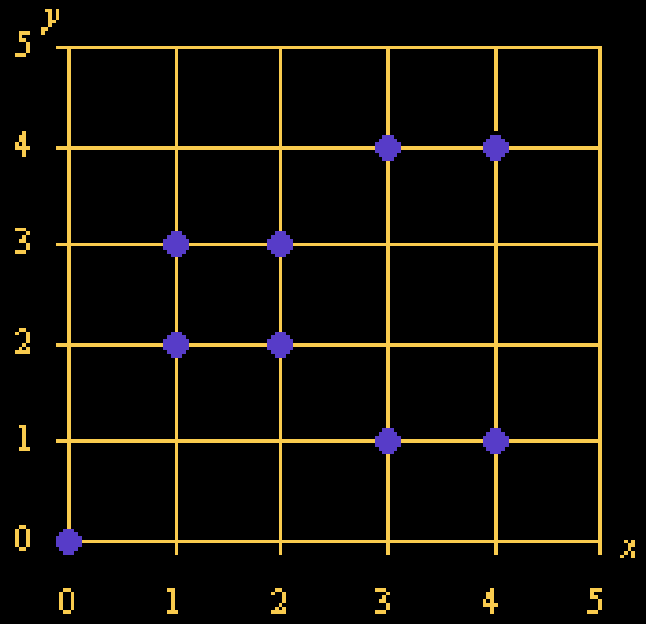
Véges test feletti elliptikus görbe:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

$x, y, a, b, c, d, e \in$ **GF(q)**

Gyakorlati jelentősége a **GF(2n)** és a **GF(p)** feletti ellipt. görbéknek van.

Példa egy véges test feletti elliptikus görbére: $y^2 = x^3 - 2x^2 \pmod 5$



A modulo 5 síknak összesen 5*5 = 25 pontja van.

A görbének 10 pontja van:
(0,0), (1,2), (1,3), (2,2), (2,3),
(3,1), (3,4), (4,1), (4,4), **0**

Nyilvános kulcsú kriptográfiai algoritmusok

ECC

ECC algoritmus (1985-)

Közös paraméterek: E(K) elliptikus görbe, **B** bázispont

Titkos kulcs: k; véletlen szám; k < csoport rendje

Nyilvános kulcs: **P** (= **B**⊗**B**⊗ … ⊗**B** = k⊙**B**)

Kódoló függvény: E(**M**) = (**Y**$_1$, **Y**$_2$);
Y$_1$ = r⊙**B**; **Y**$_2$ = **M** ⊗ r⊙(k⊙**B**); r véletlen egész

Dekódoló függvény: D(**Y**$_1$, **Y**$_2$) = **Y**$_2$ ⊗ (-) k⊙ **Y**$_1$

- Viszonylag fiatal
- Bonyolultabb matematikai háttér: ell. görbék pontjain ért. csop.
- Az ellene használható törő algoritmusok is lassabban működnek

**Adott biztonsági szint eléréséhez sokkal kisebb kulcsméretre van szükség, mint az RSA-nál,
vagyis az „egy kulcsbitre jutó biztonság” értéke nagyobb.**

- Paraméterek:
  - Kulcsméret: 110 - 570 bites; ajánlott: 160 bites.
  - Közös paraméterek: ajánlások (ANSI, Certicom)
- Alkalmazott gyorsítás: Előszámított táblák **B**-re és **P**-re.

Nyilvános kulcsú kriptográfiai algoritmusok

Különbségek

A különböző nyilvános kulcsú rendszerek ugyanazokat a funkciókat valósítják meg, de nem teljesen csereszabatosak.

Különbség:

- Inkompatibilis paraméterek (nyilv.-titk. kulcspár, közös param.)
- Ismert legjobb általános törő algoritmus költsége

Emiatt:

- Különböző működési jellemzők
- Eltérő korlátok

Összehasonlítás:

Nem csak a kis kulcsméret számít. Lényegesek lehet az algoritmusok használata során tapasztalható hatékonysági jellemzők és működési korlátok is.

Az algoritmusok inkompatíbilitása az összehasonlításnál nehézséget jelent.

Az elemzés bemutatása

A mért adatok: az ECC-nél és az RSA-nál a

- kulcsgenerálás
- közös paramétereinek felállítása
- kódolás
- dekódolás
- aláírás
- aláírás ellenőrzés

során

a futási idők valamint a műveletek során létrejövő adatok mérete, vagyis a:

- környezeti paraméterfile-ok mérete,
- nyilvános és titkos kulcsfile-ok mérete,
- kódolt adatfile-ok mérete,
- aláírásfile-ok mérete.

Az elemzés bemutatása

A mérés bemutatása

Az eredmények függenek az aktuális paraméterválasztástól:
- az alkalmazott kulcs méretétől,
- a bemeneti adat méretétől és tartalmától,
- RSA-nál az exponens értékétől,
- ECC-nél az alapul szolgáló csoport típusától (GF(2n) vagy GF(p)),
- ECC-nél előszámított táblázat használatától.

- Összefüggések feltérképezése:
  minden paramétertől való függést külön-külön kell vizsgálni.

- Sokféle paraméter és ezek változatos, értelmes kombinációi:
  ECC esetében kb. 4000,
  RSA esetében több, mint 2000 mérés elvégzése.

Az elemzés bemutatása

Az összehasonlítás bemutatása

Az algoritmusok inkompatíbilisek.
A összehasonlításhoz mégis valamilyen megfeleltetés kell.

(1) Kulcsok mérete:

Kulcsméret ~ biztonsági szint.

- Azonos biztonságot nyújtó kulcsméretek mellett.
- Adott művelet milyen kulcsméret mellett ad azonos eredményt

Azonos biztonságot nyújtó kulcsméretek

Biztonsági szint ~ legjobb általános törő algoritmus gyorsasága

RSA: szubexponenciális
ECC: tisztán exponenciális

Összerendelés: Nemzetközileg elfogadott adatok alapján.

RSA (bit)	ECC (bit)
512	113
768	136
1024	160
2048	282
4096	409

Az elemzés bemutatása

Az összehasonlítás bemutatása

(2) Algoritmus típusok:

- ECC-nél és az RSA-nál is többféle típus
  - RSA: exponens értéke
  - ECC: alapul szolgáló csoport típusa
- Befolyással vannak a hatékonysági adatokra
- Ezek között nem létesíthető értelmes megfeleltetés
- Az algoritmusok egy-egy további típusaként kell őket kezelni.

Összehasonlításkor mindegyik típust meg kell vizsgálni.

RSA: Az exponens függvényében 3 releváns eset

e = 3; e = 65537; e = véletlen.

ECC: Az alapul szolgáló csoport típusától függően 3 lényegesen különböző eset.

GF(p); GF(2^n) trinomiális; GF(2^n) pentanomiális.

Az elemzés bemutatása

Szoftver implementáció

Választott szoftver implementáció: **Crypto++**

- Nyílt forráskódban elérhető
- Tartalmazza az ECC és RSA szükséges algoritmusait is
- 1995 óta folyik a fejlesztése
- Széles felhasználói kör
- Sok kriptográfiai algoritmus implementációját tartalmazza a nemzetközileg elfogadott specifikációknak megfelelően (IEEE P1363, X.509, SEC1, SEC2 stb.).

Közös implementációs forrás lényeges szempont!

Különböző programnyelvek, különböző optimalizálások, különböző programozói módszerek hatásának kiküszöbölése.

Az elemzés bemutatása

A mérési környezet bemutatása

A teszteléshez használt szoftver:

Szoftver keretrendszer:

C++ nyelven, Microsoft Visual C++ 6.0

Crypto++ 4.1-es verziója

- Parancssorból paraméterezhető (kriptográfiai művelet, bemeneti paraméterek, kimenetek megadása, ismétlési szám)
- Futtatás batch file-ok segítségével
- Mérési eredmények: text file → Microsoft Excel-el elemezve

Hardver és szoftver környezet:

PC:

- Intel Celeron 450 MHz processzor
- 128 KB synchronous write-back L2 On-board cache
- 192 MB memória
- 75 MHz memory bus speed
- 75 MHz FSB

Windows 98 (version 4.10.1998) operációs rendszer

Összehasonlító elemzés

I. Közös paraméterek és kulcsgenerálás

Futási idők

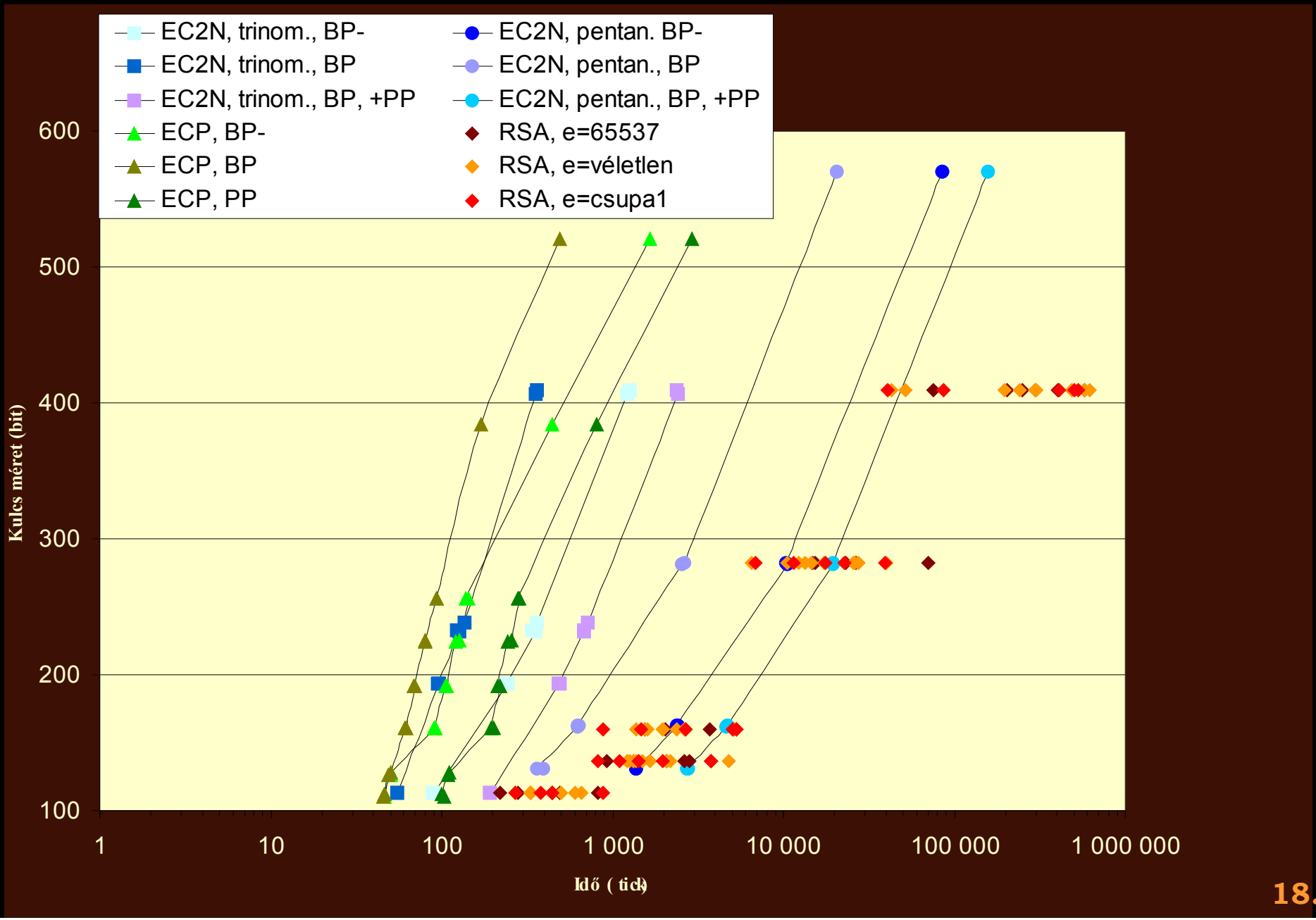
RSA

- Kritikus pont a *prím előállítása*. Módszer: Véletlen szám előáll., prímtesztelés. Valószínűségi alapon fogadunk el. Futási idő nem állandó, esetenként igen nagy lehet.
- A kulcsgenerálás idejei exponenciális eloszlást mutatnak.
- Függ a kulcsmérettől, nem függ az exponens választástól.
- Futási idők: 0,2 s - 14 perc. Jellemző érték: 2,8 s (1024 bit).

ECC

- Új közös paraméterek generálása bonyolult.
- Előszámított táblák alkalmaza: időigény, többlet tárhelyigény.
- Függ a kulcsmérettől, a típustól és az előszámított tábla használatától.
- Futási idők: 0,054 s - 1,4 perc. Jellemző érték: 0,09 s (ECP, 161 bit, előszámított tábla).

I. Közös paraméterek és kulcsgenerálás



I. Közös paraméterek és kulcsgenerálás

Méretek (nyilvános és titkos kulcsok adatfile-jai)

RSA

- Függ a választott kulcsmérettől és exponens értékétől.

ECC

- Függ a kulcs méretétől és az algoritmus típustól.

Összehasonlítás (azonos biztonságot nyújtó kulcsméretek)

Közös kulcsméret (bit)	RSA összesen (byte)	ECC összesen (byte)	ECC össz. PP-vel (byte)
ECC 113 = RSA 512	872-998	1452	3608
ECC 131 = RSA 768	1224-1422	1632	4044
ECC 160 = RSA 1024	**1584-1844**	**1890**	**4686**
ECC 283 = RSA 2048	3016-3532	3158	8006
ECC 409 = RSA 4096	5848-6870	4428	11328

II. Kódolás és dekódolás

Kódolás időigénye

RSA

- Függ a kulcsmérettől, a nyilvános exponens választástól, nem függ a kódolandó adat méretétől és típusától.
- Futási idők: 0,02 s - 6,7 s. Jellemző érték: 0,025 s (1024 bites kulcs, e=65537).

ECC

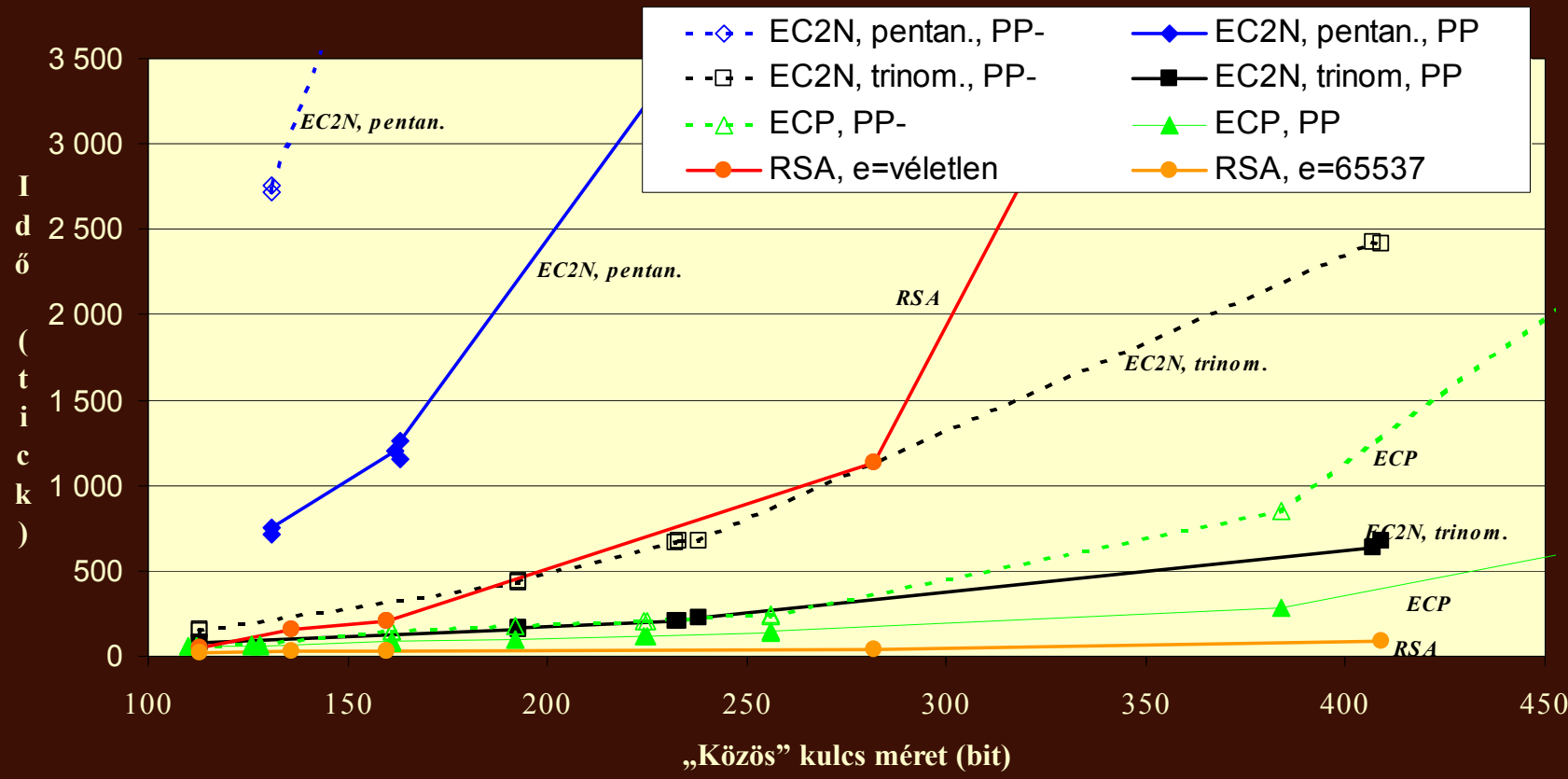
- Függ a kulcsmérettől, a típustól, a kódolandó adat méretétől és az előszámított tábla használatától.
- Előszámított táblázat használata több, mint kétszeresére gyorsítja a műveletet.
- Futási idők: 0,05 s - 2,8 perc. Jellemző érték: 0,12 perc (ECP, 163 bit, 2048 byte-os adat, előszámított tábla).

Összehasonlítás

A kicsi exponensű RSA a leggyorsabb (az e=65537 eset is ide tartozik még). Ez kb. 4-5-szörös gyorsaságot jelent. (Egyes szakirodalmak 7-szeres szorzóról is beszélnek.)

II. Kódolás és dekódolás

Kódolás időigénye



II. Kódolás és dekódolás

Dekódolás időigénye

RSA

- Függ a kulcsmérettől, nem függ a nyilvános exponens választástól és a kiindulási adat méretétől és típusától.
- Futási idők: 0,03 s - 4,45 s. Jellemző érték: 0,13 s (1024 bites kulcs).

ECC

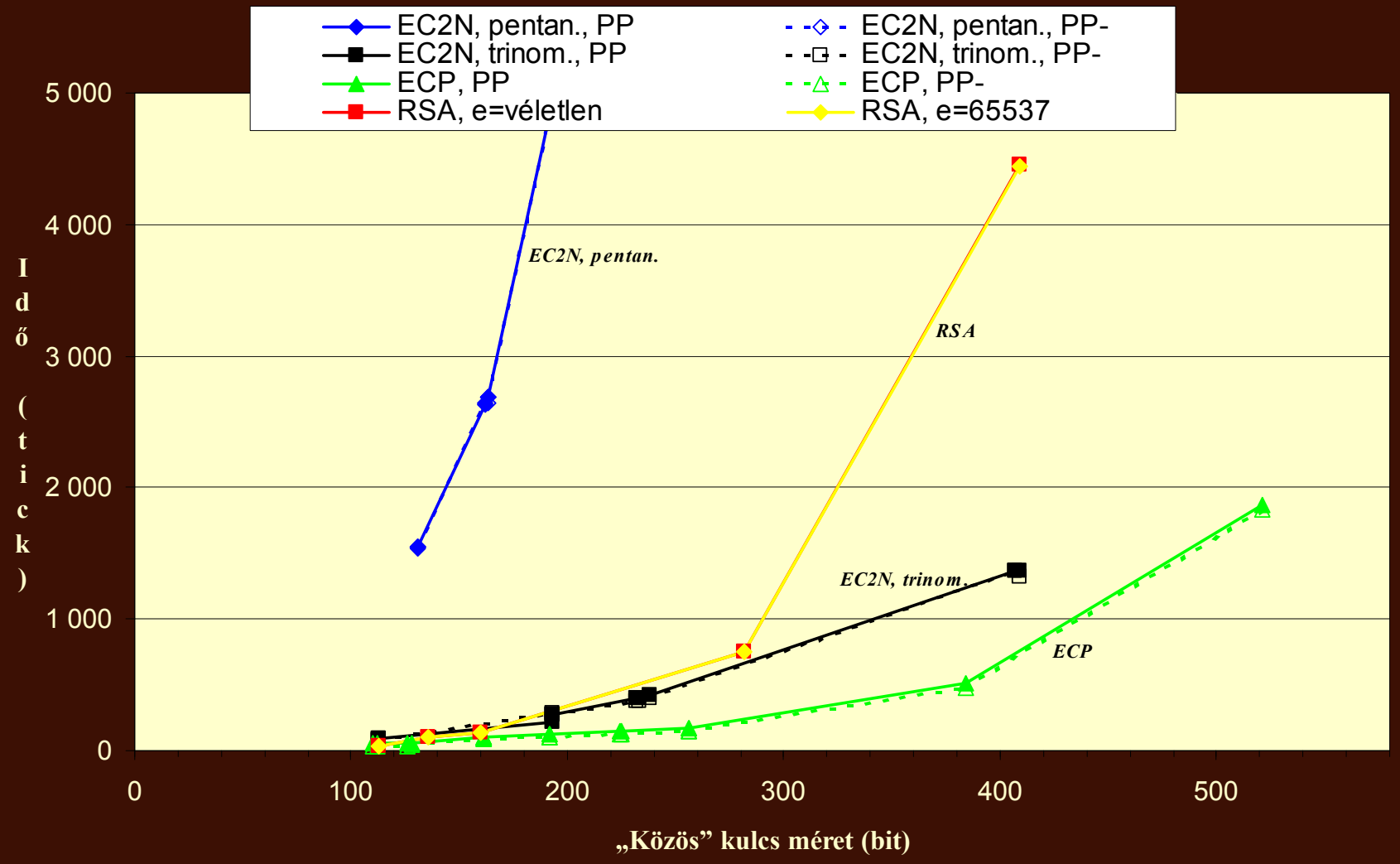
- Függ a kulcsmérettől, az algoritmus típustól, kiindulási dokumentum méretétől, nem függ az előszámított tábla használatától.
- Előszámított táblázat használata több, mint kétszeresére gyorsítja a muveletet.
- Futási idők: 0,03 s - 1,55 perc. Jellemző érték: 0,136 perc (ECP, 163 bit, 4096 byte-os adat).

Összehasonlítás

A dekódolásnál az ECP a leggyorsabb. Méréseink szerint ez kb. 2-szeres gyorsaságot jelent. (Szintén lehet olvasni ennél nagyobb, 6,4-szeres becslésről is.)

II. Kódolás és dekódolás

Dekódolás időigénye



II. Kódolás és dekódolás

Kódolt adat mérete, maximális kódolható adatméret

RSA

- A kódolt adat mérete csak a kulcsmérettől függ, mérete megegyezik a modulus méretével.
- A kódolandó adatnak kisebbnek kell lennie a modulusnál.

ECC

- A kódolt adat mérete függ a kulcsmérettől és a kódolandó adat méretétől. Adott kulcsméret esetén állandó a kódolt adatnak a kódolatlanhoz képest vett növekménye, 160 bites kulcsnál ez 61 byte.
- A maximális kódolható adatméret függ a kulcs méretétől, ez 160 bites kulcsnál 4035 byte. A maximális kódolható adatok kódoltja mindig 4096 byte.

Összehasonlítás

Az ECC-vel készített kódok kisebbek. Emellett sokkal nagyobb adatot lehet vele egy lépésben rejtjelezni, mint az RSA-nál, ahol ez a korlát viszonylag alacsony. RSA-nál az ennél nagyobb méretű adatok kódolására blokkonkénti kódolással van lehetőség.

Kódolt adat mérete, maximális kódolható adatméret

Közös kulcsméret (bit)	Kódolandó adat mérete (byte)	RSA-val kódolt adat mérete (byte)	ECC-vel kódolt adat mérete (byte)
ECC 113 = RSA 512	22	64	73
ECC 131 = RSA 768	22	96	77
	54	96	**109**
ECC 160 =RSA 1024	22	128	**83**
	54	128	**115**
	86	128	147
ECC 283 =RSA 2048	22	256	**115**
	54	256	**147**
	86	256	**179**
	214	256	307
ECC 409 =RSA 4096	22	512	**147**
	54	512	**179**
	86	512	**211**
	214	512	**339**
	470	512	595

	RSA		ECC	
Közös kulcsméret (bit)	**Max. kód. adat (byte)**	**Kódolt adat mérete (byte)**	**Max. kód. adat (byte)**	**Kódolt adat (byte)**
ECC 113 = RSA 512	22	64	4045	4096
ECC 131 = RSA 768	54	96	4041	4096
ECC 160 = RSA 1024	**86**	**128**	**4035**	**4096**
ECC 283 = RSA 2048	214	256	4003	4096
ECC 409 = RSA 4096	470	512	3971	4096

III. Aláírás és aláírás ellenőrzés

Időigény

Aláírás = lenyomatkészítés + titkos kulccsal való műveletvégzés

Aláírás ellenőrzés = lenyomatkészítés + művelet a nyilvános kulccsal + összehasonlítás

Az aláíráshoz / aláírás ellenőrzéshez szükséges idők viszonya nagyon hasonló, mint a dekódolás / kódolás esetén.

Az aláírás mérete

- Nem függ a kiindulási dokumentum méretétől, csak a kulcs méretőtől és a lenyomatkészítő függvénytől.

Összehasonlítás: ECC aláírások sokkal kisebbek.

Közös kulcs méret (bit)	RSA aláírás mérete (byte)	ECC aláírás mérete (byte)
ECC 113 = RSA 512	64	30
ECC 131 = RSA 768	96	34
ECC 160 = RSA 1024	128	42
ECC 283 = RSA 2048	256	72
ECC 409 = RSA 4096	512	102

Összefoglalás, értékelés

RSA

Kritikus pontok, negatívumok:

- A *kulcsgeneráláshoz* szükséges idő nem determinisztikus, esetenként igen nagy, akár percekben mérhető is lehet. Emiatt időkritikus rendszerekben, ahol a kulcsgenerálásra rendelkezésre álló idő felülről korlátos lehet, nem alkalmazható.
- A *kulcs mérete* (modulus mérete) megszabja a *kódolható maximális adatméretet*, ami viszonylag kicsi. Az ennél nagyobb méretű adatok kódolására blokkonkénti kódolással van lehetőség.
- Néhány *speciális paraméterválasztás* (pl. közös modulus vagy közös és kicsi nyilvános exponens választás stb.) módot ad különböző támadási lehetőségekre, ezért ezeket el kell kerülni.

Specialitások, pozitívumok:

- Létezik ismert, RSA-n alapuló *anonimitási* és *vak aláírás* protokoll.

Összefoglalás, értékelés

ECC

Kritikus pontok, negatívumok:

- Új *közös paraméterek* (elliptikus görbe és bázispont) keresése körülményes, bonyolult.
- Praktikusan szükség van *előszámított táblák* alkalmazására. Ezek létrehozása időigényes (ezt csak egyszer kell megtenni), és többlet tárhelyet igényel.

Specialitások, pozitívumok:

- *„Önigazoló” aláírások*, amelyek kiváltják a certificate-ek alkalmazását.
- Kiválóan *párhuzamosítható*, ezzel nagy gyorsítás érhető el. (Pl. az aláíró lánc ellenőrzése egyetlen lépésben.)
- Szoftveres megvalósítás esetén a *prím-rendű*, hardveres implementáció esetén a *2-hatvány-rendű* csoport alapú ECC alkalmazása lehet a praktikusabb. Utóbbi esetben megfelelő célhardver alkalmazásával sokkal jobb futási eredmények érhetők el, mint a most szoftveres implementációval mért adatok.

Összefoglalás, értékelés

Egymáshoz való viszonyok

- Kódolásnál és aláírás ellenőrzésénél a kicsi exponensű RSA a leggyorsabb, kb. *4,5-ször* gyorsabb, mint az ECP.
- A dekódolás és aláírás létrehozása az ECP esetében a leggyorsabb, *2-szer* gyorsabb, mint az RSA.
- A paraméterek file-ok mérete ECC-nél és RSA-nál nagyjából ugyanannyi, viszont előszámított táblákkal az ECC majdnem *3-szor* annyi helyet igényel.
- Az ECC-vel készített kódok és digitális aláírások akár 3-szor kisebbek, mint az RSA-val készültek.
- ECC-vel sokkal *nagyobb, akár 10-szer akkora adatot* lehet egy lépésben rejtjelezni, mint az RSA-nál, ahol ez a korlát viszonylag alacsony.

Összefoglalás, értékelés

- A maximális kódolható adatméret korlátja miatt nagy adat kódolásakor (a kulcsmérettől függően akár már 100 byte felett) az RSA-nél az adatot kisebb darabokra kell tördelni, és a kódolást több részletben kell elvégezni. Emiatt a kódolás az ECC-t alkalmazva akár *10-szer gyorsabban* elvégezhető, mint RSA-val, annak ellenére, hogy a kis exponensű RSA azonos méretű adat kódolásakor gyorsabb az ECC-nél.
- A számítási kapacitások növekedésével az elegendő biztonságot nyújtó RSA kulcshossz nagyobb mértékben fog növekedni, mint az ECC kulcshossz.

Σ:

- **Általánosságban nem egyértelmű!**
- **Adott alkalmazás szempontjából eldönthető.**

Általánosságban nem egyértelmű!
Melyik a jobb algoritmus…?

Érdekesség: Új riváliosok

XTR (Efficient Compact Subgroup Trace Representation)
http://www.ecstr.com/

- Kidolgozók: *Arjen Lenstra, Eric Verheul*
- Crypto2000 konferencián publikálták először
- Az XTR biztonság a véges testek feletti DLP-n alapszik

- *„Az XTR legalább olyan biztonságos, mint az RSA.”*
- *„Az XTR pareméter- és kulcsgenerálása 80-szor gyorsabb, mint az RSA.”*
- *„Az XTR olyan kulcsbit-hatékony, mint az ECC.”*
- *„Az XTR gyorsabb, mint az ECC.”*
- *„Az XTR egyesíti az RSA és az ECC előnyeit, jó alternatívája ezeknek az SSL-ben, WAP-ban stb.”*

Érdekesség: Új riválisok

NTRU

http://www.ntru.com/

- Kidolg.: *Jeffrey Hoffstein, Joseph Silverman, Jill Pipher, Daniel Lieman*
- 1996: NTRU Cryptosystems Inc.
- 2000. július: USA szabadalom
- Az NTRU biztonság a rács redukción alapszik
- NTRU Challange (eddigieket senki sem oldotta meg)

- *„Az NTRU akár 2000-szer gyorsabb, mint az alternatívái.”*
- *„A szükséges programkód akár 1/50 méretű.”*
  (Kódolás 20 soros, dekódolás 52 soros C program.)
- *„Az NTRU legalább 100-szor gyorsabb, mint az ECC.”*
- *„Az NTRU struktúrája egyszerűbb az RSA-énál.”*
- *„Az NTRU kulcsméretek hasonlóak a jelenleg használtakhoz.”*
  - *Titkos kulcs: max. 80 bit*
  - *Összesen: max. 2000 bit*
- *„A kódolás és dekódolás az NTRU-val 1-2 nagyságrenddel gyorsabb.”*
- *NTRU 167 ≈ RSA 512; NTRU 263 ≈ RSA 1024; NTRU 503 ≈ RSA 2048;*
- *Törés: NTRU 167: 550 év; NTRU 503: 5,4 10^3 év*

Irodalmak

- **RSA**

RFC 2437, PKCS #1: RSA Cryptography Specifications

http://www.rsa.com, http://www.ietf.org

- **ECC**

Sandards for Efficient Cryptography Group (SECG)

SEC1: *Elliptic Curve Cryptography*

SEC2: *Recommended Elliptic Curve Cryptography Domain Parameters*

- **Crypto++ "*a C++ Class Library of Cryptographic Primitives*"**

http://www.cryptopp.com

- ***XTR (Efficient Compact Subgroup Trace Representation)***

http://www.ecstr.com/

- ***NTRU***

http://www.ntru.com/

Endrődi Csilla

<csilla@mit.bme.hu>