# Paramétervválasztás nyilvános kulcsú kriptográfiai rendszereknél.

Pethő Attila, Debreceni Egyetem

Budapest, 2002. május 7.

# 1. Bevezetés

A nyilvános kulcsú kriptorendszerek paraméterválasztásánál legalább három fontos követelményt kell szem előtt tartani:

- **Biztonság:**
  - Elegendően nagyok legyenek.
  - Véletlenül válasszuk őket.
  - Kerüljük a "gyenge" kulcsokat, azaz ismert algoritmusokkal ne lehessen azokat megtalálni, vagy a kódolt üzenetet megfejteni.
- **Hatékony implementálhatóság:**
  - Ne legyenek túl nagyok. A kódoló/dekódoló algoritmusok sebessége a kulcs hosszának **harmadik** hatványával arányos!
  - Legyenek speciális alakúak, pl. $2^k \pm 1$.
- **Szabványosíthatóság:**
  - Bizonyos paraméterek szabványosak legyenek. Pl. ElGamal rendszereknél a ciklikus csoport.

A követelmények egymásnak ellentmondanak!

# 2. RSA

Legyenek

- $p, q$ különböző prímszámok, $n = pq$ és $\varphi(n) = (p-1)(q-1)$,
- $0 < e, d < \varphi(n)$ egészek, amelyekre $ed \bmod \varphi(n) = 1$.

- **Nyilvános kulcsok:** $n, e$.
- **Titkos kulcsok:** $d, (p, q)$.

A $0 \leq m < n$ üzenet kódolása:

$$c \equiv m^e \bmod n.$$

A $0 \leq c < n$ titkosított üzenet dekódolása:

$$m \equiv c^d \bmod n.$$

A dekódolás egyértelmű, ha $(m, n) = 1$ vagy $m = 0$, különben faktorizálni tudjuk $n$-et.

**Ismert támadások:**

- $n$ faktorizálása $\longrightarrow$ teljes feltörés.
- Kis $d$ $\longrightarrow$ teljes feltörés.
- Kis $e$ $\longrightarrow$ bizonyos üzenetek megfejtése.

A továbbiakban $kd$ illetve $kb$ egy $k$ jegyű decimális illetve bináris számot jelent.

### 2.1 $p$ és $q$ választása

Néhány ismert és hatékony faktorizáló algoritmus:

- Fermat módszere. Ha $p-q$ nem nagy, akkor hatékony. Ergo: nemcsak $p$ és $q$, de $p-q$ is nagy legyen.
- Kvadratikus szita (C. Pomerance, 1986).
- Elliptikus görbés faktorizáció (H.W. Lenstra Jr., 1987).
- Számtest szita (J.M. Pollard, ~1990).

A prímfelbontás jelenlegi csúcsteljesítménye: RSA 155, CABAL, 1999 a számtest szita felhasználásával.

*Factorization of a 512-bits RSA key using the Number Field Sieve*

———————————————————————————-

*On August 22, 1999, we found that the 512-bits number*

RSA-155 =
109417386415705274218097073220403576120037329454492059909
138421314763499842889347847179972578912673324976257528997
818337970765372440271467435315933543338 97

*can be written as the product of two 78-digit primes:*

102639592829741105772054196573991675900716567808038066803341933521790711307779
*
106603488380168454820927220360012878679207958575989291522270608237193062808643

**Wassenaar Utasítás: A legalább 512 b kulccsal működő RSA implementációk exportjához engedély szükséges.**

Debreceni példák. (Járási István) HP Omnibook + MAPLE V, SUN 2 db SuperSPARC 40 MHz + MAGMA

| $p, q$ | n | idő | gép + szoftver |
|---|---|---|---|
| 20 d | 40 d | 35 perc | PC + MAPLE |
| 35 d | 70 d | 2.1 óra | SUN + MAGMA |
| 40 d | 80 d | 21.5 óra | SUN + MAGMA |

$$
\begin{aligned}
n &= 36181294466683518350010965391743467599820817648828956077674100513717033297298 87 \\
p &= 32062222085722974121768604305614071 \\
q &= 45580037409259811952655310075487251
\end{aligned}
$$

**A $p$ és $q$-t tehát jelenlegi ismereteink szerint a következőképpen kell megválasztani:**

- Legalább 512 bináris jegyűek legyenek. Ekkor $n$ 1024 bites. Lenstra és Verheul táblázata szerint ez nagyjából 72 bit hosszúságú szimmetrikus kulcs biztonságával ekvivalens.
- $p - q$ legalább 511 bináris jegyű legyen.
- Válasszuk őket ezen feltételek mellett véletlenül.

Prímszámok keresésére **valószínűségi** és **determinisztikus** tesztek állnak a rendelkezésünkre. A valószínűségi tesztek, pl. Miller-Rabin teszt igen gyorsak, eredményeik kriptográfiai szempontból megfelelőek, de nem döntik el teljes biztonsággal, hogy az input prím-e.

A legjobb ismert determinisztikus módszer, az Atkin-Morain teszt. Ennek egy újabb implementációjával F. Morain bizonyította, hogy $x^y + y^x$ prím $x = 1148, y = 321$, 2878d és $x = 1040, y = 553$, 2763d mellett.

## 2.2. Kis $d$.

**1. Tétel.** *[M. Wiener, 1990] Legyenek $p, q, n, e, d$ RSA paraméterek és tegyük fel, hogy $d < (n/18)^{1/4}$. Akkor van olyan $k$ egész, hogy $k/d$ az $e/n$ egy közelítő törtje.*

Tekintettel arra, hogy $e$ és $n$ nyilvánosak, könnyű kiszámítani az $e/n$ lánctörtelőállítását, így annak közelítő törtjeit és a jelölteket $k/d$-re.

## 2.2. Kis $e$.

Ez a kérdés sokkal izgalmasabb, mint az előző, különösen a Smart kártyák alkalmazása miatt. Azok általában kódoló feladatokat látnak el és kicsi a számítási/tároló kapacitásuk. Ezért szívesen alkalmazzák az $e = 3, 17, 65537 = 2^{16} + 1$ nyilvános kulcsokat. Az alábbi tétel miatt 3 és 17-et ne használjuk!

**2. Tétel.** *[D. Coppersmith, 1997] Legyen $n$ összetett szám és $p(x) \in \mathbb{Z}[x]$ egy $k$-ad fokú polinom. Ha a*

$$p(x) \equiv 0 \pmod{n}$$

*kongruenciának van olyan $x_0$ megoldása, amelyre $|x_0| < n^{1/k}$, akkor $x_0$-at $k$ és $\log n$-ben polinom időben meg lehet határozni.*

Példa: Tegyük fel, hogy A az $e = 3$ és $n = 10111 * 22367 = 226152737$-et használja és B-nek az $m = 23467$, C-nek az $m + t = 23467 + 37 = 23467$ üzenetet küldi. A titkosított üzenetek: $c_B = m^3 \bmod n = 6985435$ és $c_C = (m + t)^3 \bmod n = 169885946$. Számítsuk ki az

$$\begin{aligned} R(y) &= Res_x(x^3 - c_B, (x + y)^3 - c_C) \\ &= y^9 + 189756678y^6 + 22712249y^3 + 20484351 \end{aligned}$$

rezultánst.

# 3. Elliptikus görbék

Legyen $\mathbb{F}_q$ egy véges test, ahol $q = p^k$ és $p$ prímszám. Legyenek $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$. Az

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 \ : \ y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6\}\cup\{O\}$$

halmazt elliptikus görbének nevezzük.

Ha $p > 3$, akkor $E(\mathbb{F}_q)$ definiálható

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 \ : \ y^2 = x^3 + Ax + B\} \cup \{O\}$$

u.n. rövid normálalakkal is, ahol $A, B \in \mathbb{F}_q$.

$E(\mathbb{F}_q)$-t megfelelő összeadással ellátva véges Abel csoportot kapunk. Erről tudjuk, hogy legfeljebb két ciklikus csoport direktösszege.

**3. Tétel.** *[Hasse, 1933] $E(\mathbb{F}_q)$ elemeinek a száma $q+1-t$, ahol $|t| \leq 2\sqrt{q}$.*

Az $E(\mathbb{F}_q)$-t **szuperszinguláris**nak nevezzük, ha $p$ osztja $t$-t.

Legyenek $P \in E(\mathbb{F}_q)$ és $m \in \mathbb{Z}$. Akkor

$$mP = \pm(\underbrace{P + \cdots + P}_{|m|})$$

.

### 3.1 Titkosítás elliptikus görbékkel:

(1) Előkészítés:
  (a) A és B választ egy $E(\mathbb{F}_q)$ elliptikus görbét.
  (b) A választ egy $P$ pontot $E(\mathbb{F}_q)$-n, amelynek a rendje $n$.
  (c) A választ egy $1 \leq e \leq n-1$ egészet, amelyre $(e, n) = 1$. Kiszámítja azt az $1 \leq d \leq n-1$-et, amelyre $ed \bmod n = 1$.
  (d) A nyilvánosságra hozza $n, P$ és $Q = eP$-t.

(2) Titkosítás: B az $(x, y) \in \mathbb{F}_q^2$ üzenetet küldi.
  (a) B választ egy $1 \leq v \leq n-1$ véletlen számot.
  (b) B elküldi $A$-nek az $(u, W) = ((x, y) \oplus vP, vQ) \in \mathbb{F}_q^4$-t.

(3) Visszafejtés:
  (a) A kiszámítja $u \ominus dW = u \ominus vP = (x, y)$-t.

Az algoritmusban $\oplus, \ominus$ a koordinák-kénti összeadást, illetve kivonást jelenti.

- **Nyilvános kulcsok:** $n, P, Q$.
- **Titkos kulcsok:** $e, d, v$.

Az EC-kriptorendszer biztonsága a diszkrét elliptikus logaritmus - $Q = eP$ ismeretében $e$ kiszámítása - nehézségétől függ. **Ezt eddig még senki sem bizonyította.**

Ismert módszerek DEL számítására:

- D. Shank, baby step - giant step.
- Pohlig-Hellman algoritmus, ha $n$-nek nincs nagy prímosztója.

## 3.2 A görbe és a pont megválasztása.

A véges test választására általában két megoldást javasolnak:

(1) $p = 2$. Hatékony implementációt tesz lehetővé, de sokan kritizálják, pl. G. Frey.

(2) $q = p$. Később.

"Gyenge" paraméterek.

**4. Tétel.** *[Menezes, Okamoto, Vanstone, 1991] Ha* $E(\mathbb{F}_q)$ *szuperszinguláris, akkor az* $E(\mathbb{F}_q)$*-beli DELP-t redukálni lehet az* $\mathbb{F}_{q^k}$*-beli DLP-re, ahol* $k \in \{1, 2, 3, 4, 6\}$*. Továbbá a redukció stochasztikusan polinomiális* $\log q$*-ban.*

Ez kellemetlen, mert sok olyan görbe, amelyen egyszerű az összeadás - pl. $y^2 + y = x^3$ 2-karakterisztikájú testek felett - szuperszinguláris.

**5. Tétel.** *[Smart, Schaeffer] Ha* $E(\mathbb{F}_p)$ *elemeinek a száma* $p$ *vagy* $p + 1$*, akkor abban a DELP* $\log p$*-ben polinomiális időben megoldható.*

Ennek elméleti jelentősége van, mert ilyen görbék nagyon ritkák.

## 3.3 Egy rekord

R. Harley, D. Doligez, D. de Rauglaudre and X. Leroy of INRIA (France), 2000 április 4,

*The one just solved, called ECC2K-108, is defined as follows. Let the curve* $C$ *be* $y^2 + x * y = x^3 + x^2 + 1$ *over* $GF(2^{109})$.

*Represent* $GF(2^{109})$ *as* $GF(2)[t]/(f(t))$ *where* $f(t) = t^{109} + t^9 + t^2 + t + 1$ *and is irreducible over* $GF(2)$.

*Then the following two points:*

$$\begin{aligned} P &= (0x0478C46CC96338CED91565E17257, \\ &\quad 0x1E7965E4A3AFB73A48FC9AB790E9) \\ Q &= (0x1FF0CE5EC61893F2119C3077C59E, \\ &\quad 0x1F20E9B010AC691C9B87B438241D) \end{aligned}$$

*are on* $C$*, where the coordinates have been written as hexadecimal integers by reducing modulo* $f$ *and setting* $t = 2$.

*The problem is to find the logarithm of* $Q$ *to the base* $P$.

*The problem takes place in the sub-group of order* $g$ *where* $g$ *is the prime* 324518553658426701487448656461467*, making this by far the most difficult such problem ever solved. It is also the most difficult calculation to date in public-key cryptography, being approximately 25 times as hard as the recent factorisation of RSA-155 and roughly equivalent to the factorisation of a 600-bit RSA modulus.*

*The solution is* 47455661896223045299748316018941 *modulo* $g$*, and was arrived at after 4 months of computation on 9500 computers operated by 1300 volunteers around the Internet.*

## 3.4 Hogyan válasszuk tehát egy EC-kriptorendszer paramétereit?

Wassenaar Utasítás szerint az exportálható maximális kulcs: 112 bit.

Lenstra és Verheul táblázata szerint a kulcshossz ($n$) 135 bit $\approx$ 45 d, ha az 1024 bites RSA biztonságát akarjuk elérni.

(1) Válasszuk $p$-t véletlenül a kívánt nagyságrendben.
(2) Válasszuk $0 \leq A, B < p$-t véletlenül.
(3) Határozzuk meg $E : \quad y^2 = x^3 + Ax + B$ csoport rendjét. Schoof algoritmus bonyolultsága $\log^8 p$. Ha $E$ rendjének nincs nagy prímosztója, akkor goto 2.
(4) Válasszunk addíg véletlenül $0 < x < p$-t és keressünk hozzá megfelelő $y$-t, amíg $P = (x, y) \in E$ teljesül.
(5) Határozzuk meg $P$ rendjét $E$-ben. Ez időigényes lehet, ha $E$ rendje nem prímszám.

Alternatívaként kiindulhatunk $(x, y)$-ból és ehhez kereshetünk megfelelő $A, B$ párt.

## Még egy példa

Nagy Pál, 1990 diplomamunkájában a következő algoritmus hatékonyságát vizsgálta.

(1) $q \leftarrow$ random odd prime with $q \equiv 1 \pmod 3$,
$p(x) \leftarrow x^4 + a_1x^3 + a_2x^2 + a_3x + a_4 \in \mathbb{F}_q[x]$ random polynomial,
$t \leftarrow$ discriminant of $p(x)$,
(2) if $t = 0$ or $A(4) = 12a_4 + a_2^2 - 3a_1a_3 = 0$ in $\mathbb{F}_q$ then goto 1,
(3) $N \leftarrow$ the order of $\tilde{E}_t(\mathbb{F}_q)$,
(4) if $N = q + 1$ then goto 1,
(5) factorize $N$. If failed goto 1,
(6) compute $\tilde{P}_0$ and its order $M$,
(7) output $q, \tilde{P}_0, N, N/M$.

**Remarks** *1. The algorithm was tested for 80,100,120 and 200 decimal digit primes.*

*2. To compute $N = |\tilde{E}_t(\mathbb{F}_q)|$ Cornacchia's algorithm was used, which complexity is $O(\log^2 q)$.*

*3. To factorize $N$ only trial division with the first 100 000 primes was performed. If $N$ or one of its divisors passed this test and the Miller-Rabin test then it was declared to be prime. We did not used deterministic primality tests.*

*The algorithm was implemented in SIMATH 4.3. The computation were done on a PC with 166 MHz PENTIUM-MMX processor. Our experiences are the following:*

- *For 100 decimal digit primes the algorithm was performed 4000 times. We were able to compute the order of $\tilde{P}_0$ 440 times. The largest index was 1350. Curves with a point of small index were found in 1.3 minutes.*
- *For 120 decimal digit primes the algorithm was performed 2200 times. We were able to compute the order of $\tilde{P}_0$ 186 times. The largest index was 1158. Curves with a point of small index were found in 3.2 minutes.*
- *For 200 decimal digit primes the algorithm was performed 1632 times. We were able to compute the order of $\tilde{P}_0$ 89 times. The largest index was 223. Curves with a point of small index were found in about 30 minutes.*

## 4. Záró megjegyzések

*Miért bízunk jobban az RSA-ban, mint a többi javasolt kriptorendszerben? Ez nem matematikai vagy informatikai, hanem szociológiai kérdés. Válaszlehetőségek:*

- *Mert a kódoló/dekódoló algoritmus egyszerű.(?) Minden esetre sokkal egyszerűbb, mint más kriptorendszerek.*
- *Mert az alapszituációt legalább 2000 év óta tanulmányozzák tudós elmék.* **A gyengeségek ismerete sokkal biztonságosabb, mint az ismeretlen gyengeségek.**
- *Mert 26 év óta folyamatosan próbálják az RSA gyenge pontjait megtalálni, de sikertelenül. A gyengeségek mindíg az implementáció nem eléggé körültekintő, a releváns irodalmat nem ismerő alkalmazása miatt léptek föl.*

## 5. Felhasznált irodalom:


- *G. Walsh, Efficiency vs. Security in the Implementation of Public-Key Cryptography,*
- *A.K. Lenstra and E.R. Verheul, Selecting Cryptographic Key Sizes, J. Cryptology* **14** *(2001), 255-293.*
- *Wassenaar Utasítás (Arrangement), www.wassenaar.org*
- *A. Pethő, Index form surfaces and construction of elliptic curves over large finite fields, In: Public-Key Cryptography and Computational Number Theory, Eds.: Kazimierz Alster, Jerzy Urbanowicz and Hugh C. Williams, Walter de Gruyter GmbH & Co, Berlin, 2001, pp. 239–247.*