Anvendelser af kongruenser (Afsnit 4.5)

- Hash-funktioner (behandles i DM507)
- Pseudo-tilfældige tal

  Simpel og hurtig metode:

  $$X_{n+1} = aX_n + c \mod m$$

  X_0 kaldes „seed".

  Ekstra hurtig, hvis m er en 2-potens (hvorfor?)

  Parametrene a, c, m og seed X_0 skal vælges med omhu.

  Java: `java.util.Random`:

  $m = 2^{48}$, $a = 25214903917$, $c = 11$

  Bruger bits 47...16

  „Mere tilfældige": Bedre alg. eller hardware-input.

- Check-cifre

  Eks: Paritets-bit

  (Detekterer et ulige # bit-fejl)

  Eks: ISBN10: $X_1 X_2 \cdots X_{10}$

  $X_{10} \in \{0, 1, \ldots, 9, X\}$ ← repr. 10

  $$X_{10} = \sum_{i=1}^{9} i X_i \mod 11$$

  så „ombytningsfejl" kan detekteres

Eks: ISBN10-nr. for lærebogen: 0-07-131501-2 (check-cifre)

$$X_{10} = 1\cdot 0 + 2\cdot 0 + 3\cdot 7 + 4\cdot 1 + 5\cdot 3 + 6\cdot 1 + 7\cdot 5 + 8\cdot 0 + 9\cdot 1 \mod 11$$
$$= 21 + 4 + 15 + 6 + 35 + 9 \mod 11 = 90 \mod 11 = 2$$

Fejl i ét ciffer: $0-07-83150\,1-2$

$$\begin{aligned} y_{10} &= x_{10} + 4\cdot(8-1) \mod 11 \\ &= 2 + 28 \mod 11 \\ &= 2 + 6 \mod 11 \\ &= 8 \neq x_{10} \end{aligned}$$

(6 ← ændringen i checkcifferet)

$y_{10} \neq x_{10}$, fordi 28 ikke er et multiplum af 11.

Generelt: Ved fejl i x_i er ændringen i checkciffer:
$i(y_i - x_i) \mod 11 \neq 0$, fordi
$11 \nmid i(y_i - x_i)$, da 11 er et primtal, og
$i < 11$ og $|y_i - x_i| < 11$.

Ombytning af to cifre: $0-07-13\,5\,1\,01-2$

$$\begin{aligned} y_{10} &= x_{10} + 6\cdot(5-1) + 7\cdot(1-5) \mod 11 \\ &= 2 + 24 - 28 \mod 11 \\ &= 2 - 4 \mod 11 \\ &= -2 \mod 11 \\ &= 9 \neq x_{10} \end{aligned}$$

Generelt: Ved ombytning af x_i og x_j er ændringen i checkciffer:

$$\begin{aligned} & i(x_j - x_i) + j(x_i - x_j) \mod 11 = \\ & i(x_j - x_i) - j(x_j - x_i) \mod 11 = \\ & (i-j)(x_j - x_i) \mod 11 \neq 0 \end{aligned}$$

Opgave 4.31 og 4.32 handler om ISBN13:

$$x_{13} = x_1 + x_3 + \ldots + x_{11} + 3(x_2 + x_4 + \ldots + x_{12}) \mod 10$$

Kryptering (Afsnit 4.6)

Symmetrisk kryptering:

Caesar-koder

- Hemmelig nøgle
- Let at knække v.h.a. frekvensanalyse
- Hurtig

AES (Advanced Encryption Standard)

- Hemmelig nøgle
- Svær at knække
- Hurtig

Asymmetrisk kryptering

RSA (Rivest, Shamir, Adleman)

- Offentlig nøgle (kryptering) + hemmelig nøgle (dekryptering)
- Frekvensanalyse duer ikke
- Langsom

Man kan bruge RSA til at udveksle en hemmelig AES-nøgle.

Offentlig nøgle: (n, e)

- $n = pq$, p, q meget store primtal
  aktuel anbefaling: ≥ 2048 bits i alt
- $\gcd(e, (p-1)(q-1)) = 1$

$P(M) = M^e \bmod n$

↑ Public

Hemmelig nøgle: (n, d)

- $n = pq$
- $de \equiv 1 \pmod{(p-1)(q-1)}$
  d eksisterer ifølge Sætning 4.4.1, da $\gcd(e, (p-1)(q-1)) = 1$.

$S(C) = C^d \bmod n$

↑ Secret

Eks 8:

$p = 43$
$q = 59$
$\Rightarrow \quad n = 43 \cdot 59 = 2537$

$(p-1)(q-1) = 42 \cdot 58 = \underbrace{(2 \cdot 3 \cdot 7) \cdot (2 \cdot 29)}_{13 \text{ optræder ikke}} = 2436$

$\gcd(13, 2436) = 1$

$e = 13$

D.v.s. offentlig nøgle: $(2537, 13)$

$937 \cdot e = 937 \cdot 13 = 5 \cdot 2436 + 1$

$d = 937$

D.v.s. hemmelig nøgle: $(2537, 937)$

Kryptering

- Omskriv hvert bogstav til et tal (som i Cæsarkode)

  Eks: $a = 00$, $b = 01$, ..., $o = 14$, ..., $s = 18$, ...

- Gruppér tallene i blokke, så hver blok repræsenterer et tal $< n$.

  Eks: stop

  $M = \underbrace{1819}_{M_1}\underbrace{1415}_{M_2}$

- Beregn $C_i = P(M_i) = M_i^e \bmod n$

  Eks: $C_1 = 1819^{13} \bmod 2537 = 2081$

  $C_2 = 1415^{13} \bmod 2537 = 2182$

  $C = 20812182$

Dekryptering:

- Beregn $M_i = S(C_i) = C_i^d \bmod n$

  Eks: $M_1 = 2081^{937} \bmod 2537 = 1819$

  $M_2 = 2182^{937} \bmod 2537 = 1415$

  $\Rightarrow M = 1819\,1415$

- Oversæt tal til bogstaver

  Eks: 18 19 14 15 $\rightsquigarrow$ stop

Der gælder altid (som i eks. ovenfor), at

$S(P(M)) = M$

pensum i DM549, men ikke i DM547

Bevises vha. Fermats Lille Sætning, Kinesiske Restklasse-Sætning, mod-regneregler og potens-regneregler.

Der gælder også, at

$P(S(M)) = M$

Kan bl.a. udnyttes til at lave digital signatur:

Underskriv tekst M:

- Anvend „sikker" hash-funktion h til at opnå en „kort version" af M (små ændringer i M giver ændring i $h(M)$)
- Anvend hemmelig nøgle på $h(M)$
- Send $(M, S(h(M)))$

Hvis $P(S(h(M))) = h(M)$, er det meget usandsynligt, at underskriften er falsk.