VT03
LMA100: Diskret Matematik
Göteborgs universitet
SK

Läsanvisningar till 17 februari

NÅGOT OM KRYPTERING

Man använder kryptering för att obehöriga inte skall komma åt någon information. Först läser vi avsnitt **4.6** i Barnett. Sedan tar vi

Rekommenderade uppgifter till 17 februari

Uppgifter ur Barnett:

Avsnitt	**Uppgift**
4.6	4.117, 4.118, 4.121

Nu läser vi följande som kommer från boken "Tillämpade Diskreta Strukturer" av Julius Brzezinski och Jan Stevens. Det är dels repetition av avsnitt **4.6** i Barnett, men den viktigaste grejen här är att läsa om den mest kända krypteringsmetoden som kallas RSA-krypteringsystem.

Behovet av att skydda information har funnits mycket länge, men först i samband med utvecklingen av datatekniken har det blivit ett allmänt problem för alla moderna samhällen. Stora datamängder måste ofta skyddas från obehörigt intrång med hjälp av en lämplig kryptering.

Krypteringsproblemet är mycket mera komplicerat matematiskt än rent tekniskt. Situationen kan beskrivas så att det finns två mängder: mängden av meddelanden X och mängden av deras krypterade motsvarigheter Y. Det finns två funktioner:

$$E : X \longrightarrow Y \quad \text{och} \quad D : Y \longrightarrow X.$$

Den första E är krypteringsfunktionen, och den andra, D är dekrypteringsfunktionen. E krypterar x till $E(x)$, och D dekrypterar $E(x)$ till x dvs $D \circ E(x) = D(E(x)) = x$ (E och D är varandras inverser). Problemet är att konstruera E och D på ett sådant sätt att E är relativt enkel att definiera, och D är mycket svår att rekonstruera av den som inte har tillgång till dess definition.

Mera formellt kan X betraktas som mängden av informationsvektorer $\mathbf{a} = a_1 a_2 \dots a_n$, där a_i tillhör en ring R (t ex $\mathbb{Z}_2$ dvs $a_i = 0$ eller 1). En krypteringsfunktion ersätter $\mathbf{a}$ med en vektor $\mathbf{a}' = a'_1 a'_2 \dots a'_m$ tillhörande mängden Y. Man kan inte konstruera $\mathbf{a}'$ helt slumpmässigt därför att det

måste finnas ett effektivt sätt att få tillbaka **a**. Samtidigt måste övergången från $\mathbf{a}'$ till **a** vara komplicerad för att göra det mycket svårt för obehöriga att komma åt **a**.

Låt oss börja med ett exempel som är drygt 2000 år gammalt:

Exempel. (Caesarkrypto[1]) Låt oss numrera alla bokstäver (för enkelhets skull i det engelska alfabetet) med 0, 1, 2, ..., 25:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Caesarkryptot är definierat som en funktion $E(x) = x + a$, där x, $a \in \mathbb{Z}_{26}$ (dvs man adderar modulo 26). Tag t ex $a = 3$. Då är $E(x) = x + 3$ så att $E(0) = 3$, $E(1) = 4$, $E(24) = 2, \ldots$, dvs A krypteras till D, B till E, Y till C osv. Ordet MATEMATIK förvandlas till PDWHPDWLN.

Dekrypteringen är mycket enkel. Man kan använda dekrypteringsfunktionen $D(x) = x + 23$ därför att D är inversen till E dvs:

$$x \longmapsto x + 3 \longmapsto (x + 23) + 3 = x.$$

Med andra ord $D(E(x)) = x$ vilket följer ur likheten $23+3 = 0$. Till exempel dekrypteras PDW, dvs 15,3,22, till $15 + 23 = 12$, $3 + 23 = 0$, $22 + 23 = 19$, dvs MAT.

Caesarkryptot är mycket enkelt och kan knappast uppfylla dagens krav på säkra krypteringssystem. Men själva krypteringsmetoden visar vikten av en algebraisk struktur vid konstruktioner av krypterings- och dekrypteringsfunktioner.

Vi skall beskriva några krypteringstekniker som bygger på restaritmetiker dvs grupper och ringar relaterade till $\mathbb{Z}_n$. För att kunna göra det behöver vi två mycket berömda satser ur talteorin.

Vi vet redan att $\mathbb{Z}_n$ är en grupp med avseende på addition modulo n. Låt

$$\mathbb{Z}_n^* = \{r \in \mathbb{Z}_n : SGD(r, n) = 1\}.$$

T ex är $\mathbb{Z}_4^* = \{1, 3\}$, $\mathbb{Z}_5^* = \{1, 2, 3, 4\}$ och $\mathbb{Z}_6^* = \{1, 5\}$. Vi vet att $\mathbb{Z}_n^*$ är grupper med avseende på multiplikation modulo n. Ordningen $|\mathbb{Z}_n^*|$ betecknas med $\varphi(n)$. Funktionen $\varphi(n)$ kallas Eulers funktion. Man har alltså

[1]Detta krypto användes av Julius Caesar.

$$\varphi(n) = \text{ antalet heltal } \quad r \quad \text{ sådana att } \quad 0 \leq r < n \quad \text{ och } \quad SGD(r,n) = 1.$$

Definitionen ger $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(5) = 4$, $\varphi(6) = 2$, $\varphi(7) = 6$, $\varphi(8) = 4$, $\varphi(9) = 6$, $\varphi(10) = 4$ osv.

Eulers sats [2]**.** *Låt a och $n \geq 1$ vara heltal sådana att $SGD(a,n) = 1$. Då gäller*

$$a^{\varphi(n)} \equiv 1 \ (mod \ \ n).$$

Ett mycket viktigt specialfall av Eulers sats får man då $n = p$ är ett primtal. I sådant fall är $\varphi(p) = p-1$ ty $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ (alla element r i $\mathbb{Z}_p$ med undantag av 0 uppfyller $SGD(r,n) = 1$).

Fermats lilla sats [3]**.** *Låt a vara ett heltal och p ett primtal. Då är*

$$a^p \equiv a \ (mod \ \ p).$$

För våra tillämpningar behöver vi en generalisering av Fermats lilla sats:

Sats 1. Låt $n = p_1 p_2 \cdots p_k$ vara en produkt av olika primtal. Då är

$$a^{\varphi(n)+1} \equiv a \ (\text{mod} \ \ n).$$

Nu kan vi diskutera den mest kända av alla krypteringsmetoder som kallas RSA-krypteringssystem. RSA kommer från namnen Rivest, Shamir, Adleman. Dessa matematiker publicerade systemet 1978. Grunden för RSA-systemet är följande sats:

Sats 2. Låt $n = p_1 p_2 \cdots p_k$ där p_i är olika primtal. Låt e vara ett positivt heltal sådant att $SGD(e, \varphi(n)) = 1$ och låt d uppfylla kongruensen $ed \equiv 1$ (mod $\varphi(n)$). Då har funktionen:

$$E : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n \ ,$$

där $E(r) = r^e$, inversen

$$D : \mathbb{Z}_n \longrightarrow \mathbb{Z}_n \ ,$$

där $D(r) = r^d$.

[2]Leonard Euler 1707 - 1783.
[3]Pierre Fermat 1601-1665.

Bevis. Vi vet att förutsättningen $SGD(e, \varphi(n)) = 1$ garanterar att d existerar ty $\mathbb{Z}^*_{\varphi(n)}$ är en grupp. För att visa att D är inversen till E kontrollerar man att $D \circ E$ är identiteten på $\mathbb{Z}_n$ dvs $(D \circ E)(r) = r^{ed} = r$. Men $ed = 1 + q \cdot \varphi(n)$, $q \geq 1$ så att:

$$(D \circ E)(r) = r^{1+q\varphi(n)}.$$

Vi visar induktivt att $r^{1+q\varphi(n)} = r$. Likheten gäller då $q = 1$ enligt sats 1. Antag att

$$r^{1+(q-1)\varphi(n)} = r$$

i $\mathbb{Z}_n$. Då är

$$r^{1+q\varphi(n)} = r^{1+\varphi(n)+(q-1)\varphi(n)} = r^{1+\varphi(n)} r^{(q-1)\varphi(n)} = r \cdot r^{(q-1)\varphi(n)} = r^{1+(q-1)\varphi(n)} = r.$$

Alltså gäller likheten $r^{1+q\varphi(n)} = r$ för alla q dvs $(D \circ E)(r) = r$ då $r \in \mathbb{Z}_n$.

RSA-kryptering. Vi ger en beskrivning av den metoden enbart då $n = pq$ är en produkt av två olika primtal. Metoden fungerar på samma sätt då n är produkt av ett godtyckligt antal olika primtal.

(a) Välj två olika primtal p, q och beräkna $n = pq$ (p, q är vanligen mycket stora, säg av storleksordningen 10^{100}).

(b) Beräkna $\varphi(n) = (p-1)(q-1)$ och välj e så att $SGD(e, \varphi(n)) = 1$. Beräkna även d så att $ed \equiv 1 \pmod{\varphi(n)}$($d$ räknas med hjälp av Euklides algoritm).

(c) Publicera n, e och en "ordbok" för översättning av meddelanden till $r \in \mathbb{Z}_n$ (t ex $A = 10$, $B = 11$, ..., $Z = 35$ då $n > 35$).

(d) Den som vill sända meddelanden till Dig krypterar med hjälp av (den kända) funktionen $E(r) = r^e$. Du är den ende (förhoppningsvis) som kan dekryptera med hjälp av funktionen $D(r) = r^d$(d är hemligt och $D \circ E(r) = D(r^e) = r^{ed} = r$ enligt sats 2.

Exempel. Klartext ALGEBRA kodat med $A = 10$, $B = 11$, ..., $Z = 35$ är

$$1021,\ 1614,\ 1127,\ 10.$$

Låt $n = pq = 47 \cdot 167 = 7849$. Då är $\varphi(n) = 46 \cdot 166 = 7636$. Låt oss välja $e = 29$ ($SGD(29, 46 \cdot 166) = 1$). Då är $e^{-1} = 29^{-1} = 4213$ (i $\mathbb{Z}^*_{\varphi(n)}$). Krypteringen enligt RSA-metoden ger:

$$1178,\ 1929,\ 3383,\ 4578,$$

dvs $1021^{29} \equiv 1178 \pmod{7849}$ osv. Vid dekryptering räknar man: $1178^{4213} \equiv 1021 \pmod{7849}$ osv.

Anmärkning. RSA-systemet tillhör s.k. öppen-nyckelkrypton dvs krypteringsfunktionen E är allmänt känd. Vad gör den som vill beräkna inversen $D = E^{-1}$? För att beräkna d måste man lösa ekvationen $ed \equiv 1 \pmod{\varphi(n)}$. För att göra det måste man känna $\varphi(n)$. För att beräkna $\varphi(n) = (p-1)(q-1)$ måste man (med all sannolikhet) känna p och q. Men för att få p och q ur $n = pq$ (som är känt) måste man faktorisera n. Faktoriseringsproblemet är mycket svårt att lösa. De bästa kända algoritmerna för primtalsfaktorisering av ett heltal n kräver c:a $n^{1/5}$ räkneoperationer om man vill hitta en primfaktor till n. Om $p, q \approx 10^{100}$ så är $n \approx 10^{200}$. Om en räkneoperation tar $1\mu s$ så krävs det $10^{40}\mu s \approx 3 \cdot 10^{26}$ år för att genomföra beräkningarna för n (10^6 datorer var och en kapabel att utföra en räkneoperation på $1\mu s$ skulle behöva $3 \cdot 10^{20}$ år för att klara dessa beräkningar!). Men det finns inte något bevis att faktororiseringsproblemet är så pass svårt. Det är alltså möjligt att det finns bättre algoritmer som inte är kända nu. Å andra sidan ökar datorernas beräkningskapacitet dramatiskt och redan nu är man mycket försiktig med valet av p och q [4].

Anmärkning. RSA-systemet kan även användas för äkthetskontroll. Den som känner D (se beskrivningen av RSA-kryptering) kan signera dokument med $D(r) = r^d$. Den som vill kontrollera äktheten av signaturen räknar ut $E \circ D(r) = r^{de} = r$ (E är allmänt känd).

ÖVNINGAR

I alla övningar är $A = 1$, $B = 2$, ..., $Z = 26$ och mellanrum = 27.

1) Låt $n = 3 \cdot 11 = 33$.

   (a) Låt krypteringsnyckeln vara $e = 3$. Kryptera DISKRET MATEMATIK.

   (b) Bestäm dekrypteringsnyckeln d.

   (c) Dekryptera 19, 1, 4, 12, 26.

2) För att kontrollera äktheten av dokument som skickas från MATEMATIK AB använder man krypteringsnyckeln $n = 221$, $e = 7$ (känd för alla mottagare). Kontrollera äktheten av ett dokument med signaturen:
   208, 1, 45, 112, 208, 1, 45, 76, 54.

[4] Nyligen visades att man ibland kan forcera RSA-kryptot i sådana fall som tidigare uppfattades som omöjliga att klara.