TEMA 3: Uppgifter till gruppövningarna 5 och 6

Nedan står uppgifterna för gruppövningarna 5 och 6. Först några tips:

- I flera av uppgifterna är det lämpligt att skapa snurror (loopar) som pågår ända tills ett visst villkor är uppfyllt, tills skillnad från den vanliga "for"-snurran som pågår ett förutbestämt antal gånger. För denna typ av snurror finns kommandot "while" i MATLAB.

- I uppgift 4 behöver man en nyckel och därmed bl.a. två stora primtal p och q. Man kommer sedan att behöva räkna modulo pq och modulo $phi(pq)$. Eftersom MATLAB räknar med bara 16 siffrors noggrannhet får p och q vara högst 7-siffriga så att pq får högst 15 siffror. Dessutom kommer man i uppgift 3 (vars lösning förstås ska användas i uppgift 4) att behöva beräkna produkter av två 15-siffriga tal modulo pq och modulo $\phi(pq)$. Eftersom produkten kan få upp till 31 siffror kommer man då att tappa siffror om man använder vanlig produkt, men klarar sig om man skriver en egen produktfunktion "prodmod" som beräknar ab mod n enligt följande:

```
function y=prodmod(a,b,n)
a=mod(a,n);
b=mod(b,n);

if a>b,
c=b;
b=a;
a=c;
end

u=tenary(a);
s=length(u);
y=b*u(s);
c=b;

for i=s-1:-1:1,
c=mod(10*c,n);
y=mod(y+c*u(i),n);
end
```

Funktionen "tenary" som dyker upp här ser ut så här:

```
function y=tenary(x)

s=floor(log10(x));
y=ones(1,s);

rest=x;

for i=s+1:-1:1,
y(i)=floor(rest/10^(i-1));
```

```
rest = mod(rest,10^(i-1));
end
y=y(s+1:-1:1);
```

och gör om ett heltal till en vektor som består av siffrorna i talet. I uppgift 3 behöver man en funktion "binary" som gör om ett tal till en vektor med des siffror på binär form och det åstadkommer man genom att bara byta ut 10 mot 2 i "tenary".

- För att kolla om ett tal är ett primtal har MATLAB funktionen "isprime" som fungerar för tal med upp till ca 10 siffror. Funktionen "factor" delar upp ett heltal i dess primtalsfaktorer och fungerar också för upp till 10 siffror, så för att din RSA-nyckel inte ska kunna knäckas av andra via ett enkelt "factor"-kommando så behöver pq vara minst 11-siffrigt. Börja dock gärna med att testa ditt program med leksaksexempel.

1. Gör ett Eratostenes såll, dvs ett program "erat.m" som ger alla primtal upp till ett på förhand valt positivt heltal N. Om Eratostenes såll kan man läsa i bokens kapitel 7.1.

2. Skapa en funktion "bezout.m" som för två positiva heltal a och b svarar med dels $sgd(a, b)$, dels två heltal u och v sådana att $au + bv = sgd(a, b)$. Använd Euklides algoritm. Tips: MATLAB har funktionen "mod".

3. MATLABs funktion "mod" fungerar dåligt för beräkning av storheter av typen

$$m^n \bmod r$$

så fort m, n och r är någotsånär stora, pga att moduloräkningen inte utförs förrän m^n är uträknat. Därför blir din uppgift att skapa en funktion som beräknar m^n mod r. Tips: Eftersom $m^4 = (m^2)^2$, $m^8 = (m^4)^2$ etc är det lämpligt att utveckla n binärt.

4. Skapa ett program som läser och skickar RSA-krypterad kod. (A=1, B=2, ... , Ö=28, mellanslag=29). Skicka och ta emot meddelanden till och från andra grupper.