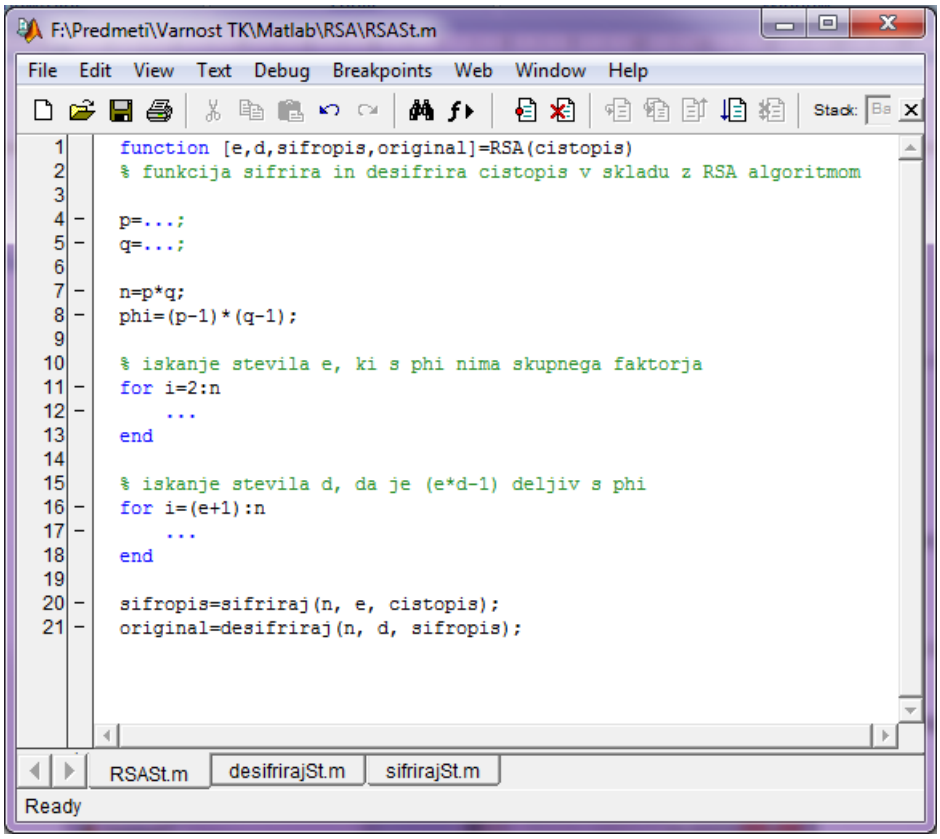
RSA šifriranje in dešifriranje

RSA algoritem:

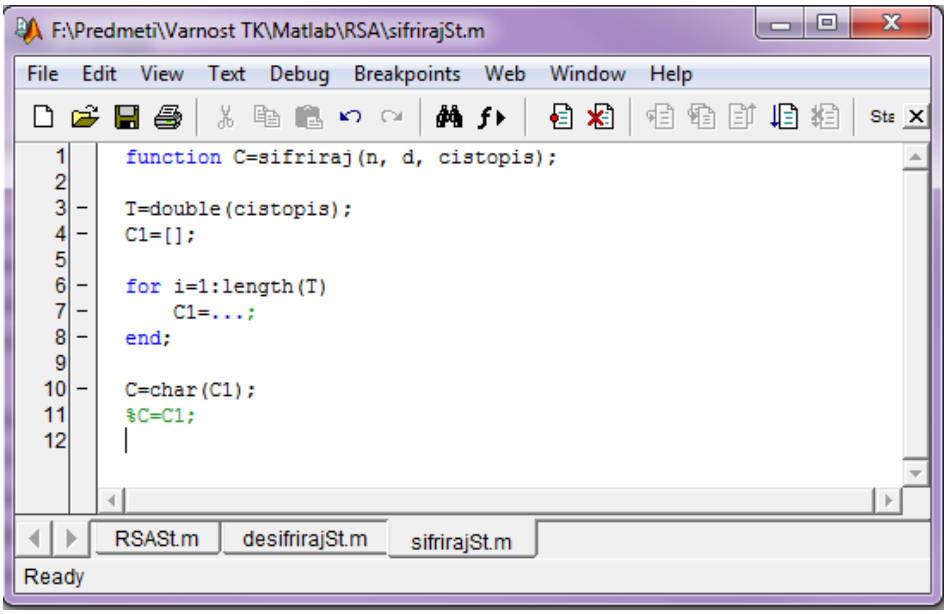
- Je asimetričen šifrirni algoritem.
- Pošiljatelj šifrira sporočilo z javnim ključem pošiljatelja (modularna eksponentna funkcija $s=c^e \bmod n$, ki se izvaja kot zaporedje modularnega množenja).
- Prejemnik dešifrira sporočilo z zasebnim ključem (modularna eksponentna funkcija $c=s^d \bmod n$, ki se izvaja kot zaporedje modularnega množenja).
- RSA izkorišča težavnost faktorizacije velikih števil.
- Na osnovi znanega javnega ključa, čistopisa in šifropisa v realnem času ni mogoče ugotoviti zasebnega ključa.
- Dolžina ključa je odvisna on zahtevane stopnje varnosti.
- RSA je počasnejši od blokovnih šifrirnih postopkov kot sta npr. 3DES ali AES. Zato je pogost naslednji šifrirni postopek med npr. Metko in Lukom:
  - Metka šifrira čistopis po AES postopku z naključno izbrani ključem.
  - Metka zahteva Lukin javni RSA ključ s katerim šifrira uporabljen AES ključ.
  - Oboje pošlje Luki.
  - Luka najprej z lastnim zasebnim RSA ključem dešifrira šifropis AES ključa.
  - S tem nato dešifrira tudi vsebino prejetega šifropisa sporočila v skladu z DES algoritmom.

Naloge:

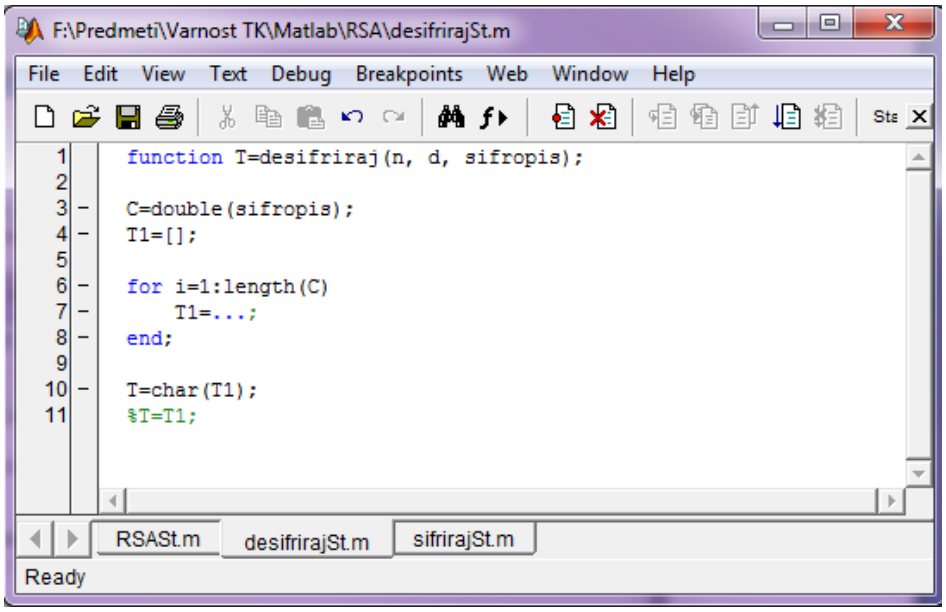
1. V programskem okolju Matlab napišite funkcijo za šifriranje in dešifriranje v skladu z RSA algoritmom *[e,d,s,c]=RSA(c)*
   a. Generirajte ključe
      i. Izberite dve praštevili *p* in *q* manjši od 100.
      ii. Izračunajte produkt *n=p*q.*
      iii. Določite število *e*, takšno da *e* in *(p-1)*(q-1)* nimata skupnega faktorja razen 1.
      iv. Določite število *d*, takšno da je *(e*d-1)* deljiv s *(p-1)*(q-1).*
   b. Čistopis šifrirajte, tako da je $s=c^e \bmod n$. Za izračun slednjega lahko uporabite funkcijo *modulo(a,x,n).*
   c. Šifropis dešifrirajte, tako da je $c= s^d \bmod n$. Za izračun slednjega lahko uporabite funkcijo *modulo(a,x,n).*



Slika 1: Zgradba funkcije *RSA*: generiranje ključev, šifriranje in dešifriranje



Slika 2: Zgradba funkcije *sifriraj* za RSA šifriranje s ključem e.



Slika 3: Zgradba funkcije *desifriraj* za RSA dešifriranje s ključem d.

2. Preizkusite delovanje algoritma:
   a. Postavite vrednosti praštevil p=61 in q=53.
   b. Kolikšni sta vrednosti ključev e in d?

      e=_____

      d=_____

   c. Šifrirajte sporočilo 'Preizkus delovanja'.
   d. Dešifrirajte šifropis iz točke 2c:

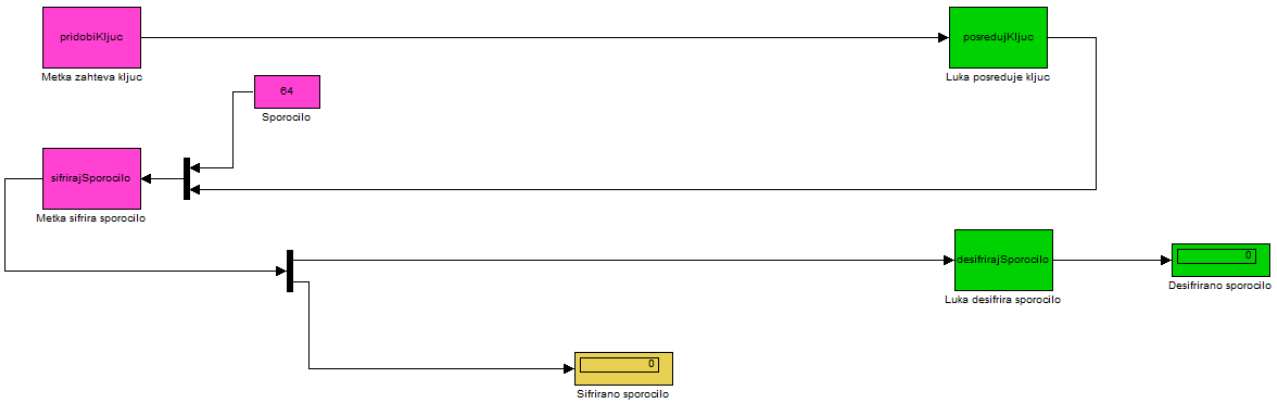
      _____

Napadi na RSA

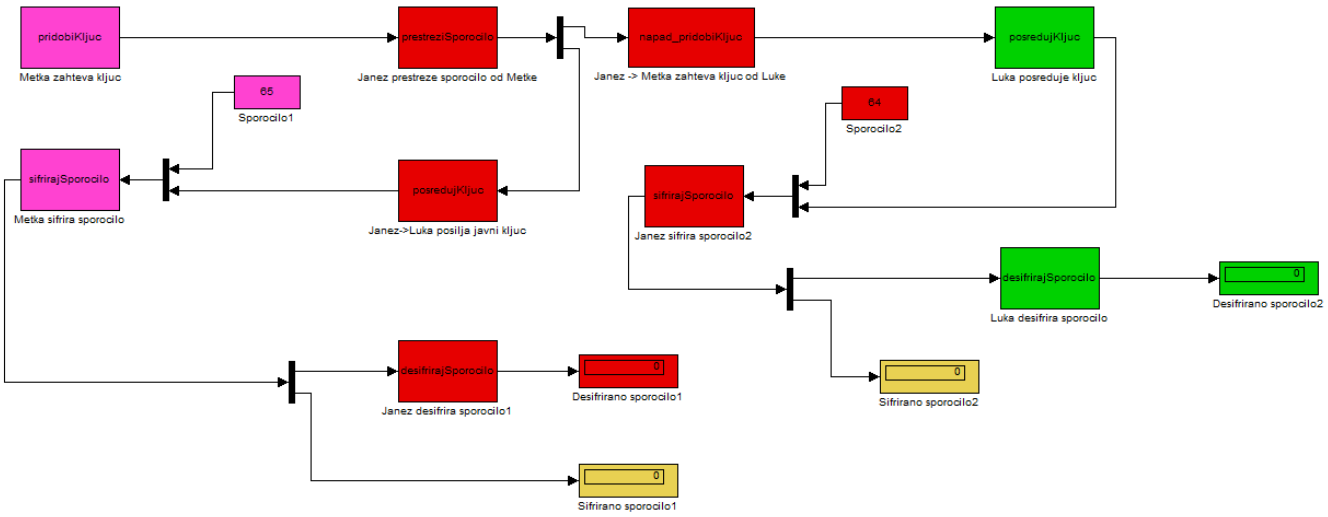
- V primeru, da sta uporabljeni praštevili *p* in *q* znani, je zasebni ključ *d* določljiv na podlagi znanega javnega ključa *e.*
- Pomanjkljivost majhnega ključa *e*:
  V takem primeru dešifriranje kratkega sporočila ($c^e<n$) je možno z iskanjem *e*-tega korena šifropisa, saj je $s=c^e$. Tak napad omogoča vpogled v vsebino šifriranega sporočila tudi brez poznavanja zasebnega ključa.
- Preprosta metoda ugibanja vsebine sporočila in preverjanja pripadajočega šifropisa, saj je RSA deterministične šifrirni postopek:
  $s\_ugib=c\_ugib^e \bmod n$
  $s=c^e \bmod n$
  Ali sta *s_ugib* in *s* enaka?
- Nevarna implementacija in hramba ključev.
- Prestrezanje sporočil ('*man in the middle*').

Naloge:

1. Izvedite napad na šifropis shranjen v datoteki sifropis.mat. Slednji je šifriran z javnim ključem e=13 (n=527).
2. V okolju Simulink opazujte potek napada s prestrezanjem sporočila:
   a. Zaženite simulacijo neposredne komunikacije med dvema osebama.



   b. Opazujte potek komunikacije za različne ključe in različna sporočila.
   c. Zaženite simulacijo komunikacije med dvema osebama, ki jo prestreže tretja oseba.



d. Opazujte potek komunikacije za različne ključe in različna sporočila.

Vprašanja:

1. Od česa je odvisna stopnja varnosti RSA šifrirnega postopka?
2. Kakšna je pomanjkljivost uporabe javnega ključa za šifriranje?
3. Kakšna je pomanjkljivost uporabe majhne vrednosti ključa *e* za kratko sporočilo *m*?
4. Število različnih praštevil manjše ali enako *k* je približno *k/ln(k)*. Koliko je različnih praštevil dolžine enake ali manjše od n_{bitov}=512?
5. Kakšna je težavnost izvedbe RSA algoritma?