Autor: Ruxandra F. Olimid
Departamentul de Informatică, Universitatea din București

- Laboratorul10-

Sisteme de criptare asimetrice

Disclaimer: Pe parcursul acestui curs/laborator vi se vor prezenta diverse noțiuni de securitate informatică, cu scopul de a învăța cum să securizați sistemele. Toate noțiunile și exercițiile sunt prezentate în scop didactic, chiar dacă uneori se presupune să gândiți ca un adversar. Nu folosiți aceste tehnici în scopuri malițioase! Acestea pot avea consecințe legale în cazul comiterii unor infracțiuni, pentru care **deveniți pe deplin răspunzători**!

1. *Factorizarea modulului RSA*

Reamintiți-vă cum funcționează *RSA*. Țineți minte că, pentru a obține aceeași securitate, dimensiunea cheii *RSA* trebuie să fie considerabil mai mare decât dimensiunea cheii *AES*.

Se consideră cheia publică RSA cu modulul pe 128 biți:

```
N=234841136411758273000763594354834942653
e=65537
```

Factorizați modulul, i.e. determinați valorile p și q [1]. Calculați apoi coeficientul de decriptare d [2].

2. *Generarea cheilor RSA folosind OpenSSL*

Răspundeți la următoarele cerințe:

a) Folosind *OpenSSL* [3], generați pentru *Alice* o cheie *RSA* pe 2048 biți, stocată într-un fișier *alice_sk.pem*.
b) Care este valoarea exponentului de criptare?
c) Decodați această cheie. Aflați valoarea modulului N și a celor două numere prime p și q.
d) Cheia lui *Alice* nu este protejată în niciun fel, deci este vulnerabilă. Alegeți o parolă puternică și generați o nouă cheie protejată folosind această parolă și *AES256*.
e) Ce diferențe observați? Decodați această cheie folosind parola folosită la creare.

f) Care este valoarea exponentului de criptare? Ce observați? Impactează această alegere securitatea?
g) Exportați cheia publică a lui *Alice* în fișierul *alice_pk.pem*. Decodați această cheie pentru a vedea valorile modulului și exponentului.

3. *Criptarea RSA și criptarea hibridă*

Răspundeți la următoarele cerințe:

a) Jucați rolul lui *Bob*. Criptați fișierul *bob_message.txt* folosind *RSA* [4] și cheia generată anterior. Încercați să criptați fișierul *bob_message.rtf* folosind *RSA* [4] și cheia generată anterior. Ce observați? De ce se întâmplă aceasta?
b) Folosiți criptarea hibridă pentru a cripta mesajul lui *Bob* către *Alice*. Pentru aceasta, generați o cheie simetrică pe 256 biți (32 bytes) și folosiți această cheie pentru criptarea fișierului *bob_message.rtf* cu *AES-CTR* [5]. Criptați noua cheie asimetric, folosind *RSA*.
c) Jucați rolul lui *Alice*. Folosiți fișierele criptate primite (criptarea cheii *AES* folosind *RSA* și criptarea mesajului folosind *AES-CTR*), decriptați și obțineți mesajul inițial.

Notă: Se poate folosi direct *smime* [6] pentru criptarea hibridă, care combină în mod direct criptarea asimetrică și criptarea simetrică.

Referințe bibliografice

1. D.Alpern. *Integer factorization calculator*. Accesibil la: https://www.alpertron.com.ar/ECM.HTM.
2. Wolfram Alpha Widget – Modulo. Accesibil la: https://www.wolframalpha.com/widgets/view.jsp?id=570e7445d8bdb334c7128de82b81fc13.
3. OpenSSL – *rsa*. Accesibil la: https://www.openssl.org/docs/man1.1.1/man1/openssl-rsa.html
4. OpenSSL – *pkeyutl*. Accesibil la: https://www.openssl.org/docs/man1.1.1/man1/openssl-pkeyutl.html
5. OpenSSL – *openssl-enc*. Accesibil la: https://www.openssl.org/docs/man3.0/man1/openssl-enc.html
6. OpenSSL – *smime*. Accesibil la: https://www.openssl.org/docs/man1.1.1/man1/openssl-smime.html