

Kriptográfia jegyzőköny 1.

Prajczér Péter

Első feladat:

Használt függvények:

- RSA kulcs generálása: `Crypto.PublicKey.RSA.generate(%bitszám)`, ami esetünkben 2048
- Publikus részét a `publickey()` hívással kapom meg
- PEM formátumra az `export_key('PEM')` alakítja át
- A kulcsot `RSA.import_key`-yel lehet beolvasni, melynek argumentumában van a beolvasott kulcs fájl

RSA kulcsaim:

- Privát PEM formátumban:

```
-----BEGIN                RSA                PRIVATE                KEY-----
MIIEpAIBAAKCAQEA rTG09wA8+G6KtQQ+GBtFWDUPuW4YjINyS9qF0SI3wKKA6z3a
3jiGGednrCIMb8DFqvKMHb1L0Ps68/BKawAsG1CZ+10nHVwXSRfLWlpjNQYkIM37
ZN1LI9NC5iHB03TZ20KjipDHPA2e+n64WbxybYMDrbGUtUj93RAvnXIKRIKE7LZ
obHgnmak65NYSKW0upihrq8FYHzIx1pL+Go+b8P3+z7B15Kf4UipXrOFjjFquKKh
Am9v+aZ+nwHWtiQJWdjVHLAsgkKOEURQnLnReM2jnuF0horNq42fryf70mfNj61
yZykgb4yMUKTUqPX/IEac5V0H5ag0m5xq+I3jwIDAQABAoIBAWhul9ku4tGEv6Y
YXvGzxLwBJNJ8aZGEmc8HguG3hORpaW8xiFddudG8lgjD8P0OORp666u8n9HrzAk
hidLR8unWwyPJJeJemnegQbo2tCROsyZ+ceb1fqyaa8vlqW64YgTzJ17ZbVtOM536
/7wpcHujU4zkTgLCCKxDacBC99Zd3iY44E7kJOa7chT6BPAA2pInIYeHQMx6kutB
EeZmNbVVRj+h8Dr/fsjj+S+qKRr/JJEvY7alB2hxEQi90jQ+5kFHBjCqrgyAzSCy
1S5t8P/nOc7soo8DLxVDIxDxVohRHQF+AEQrZrQHHiMvsbrmxFxDvNyL7kmo3OkE
hWqAdAECgYEA vYDUyrW4PKpao12vLqspDaNOxfHiW8V4hK3RqQdUWV9fdwwFdT29
eZ3MqyTvpFLNvGhCwBXGYaNIwl+a+/ONr9pVhphHehS+YgSsZWjeQXUnz9EF3Ijv
LEG2A51AxXetFXHeXhujiHtfThbNxTGvsXqmhuKoesKm+y+r0ornhYECgYEA5/qU
h69tDI6u8bGARefQrQ/nxVuASM13BKX9FZ/1nocYY33S67aRKkHX7V4WEnblwIMp
GqiCzE2287BosqR8oeinCkCqOMitJvvdTnqbk2fwPACnY5dGOtrPfJ0lLaMYgHfr
9lTyxyUNNEbkW+zg3BDbz2ewg96oDLvi53SRJQ8CgYAYomeRqjXzw3U0b59lLVgJ
nHxMKugo4/lrLN7ng+SX5CeLep5IIGQh/OVXqMpNrfEhbWj1YAOVjzrE0ilBw1zd
9Qv694xHOGJacg34TJn0K9y9kZnvvhDU2aQCYSXHf07+TsF4i9zrNjjypa2IACnD
A4+V7SUHtFX6FyoZl6WGgQKBgQC2RI+ReMe1108jviigRH0UjWu5CKTqDu4V6+nH
pPBTB6ik+4bPB5pVjnh/G67SARzJDgUs2fwzVFIMbi5FNrJqcKdWiep1RU+vUrQn
VqBomIEMVCMxP/nBYo0mRz6f9gbZYcA2NoOe70DJs23XwG4I2wlqzmj40+hMYQ/u
dgtZBwKBgQCm8WpHLr/oFcgiTDfmj7WHIIF6SjddKKaf2ed7lwEiEeUHLc6EUOD
CaYT9Fk0cZgvjvT0WQwrWaSM9ZT5DSqKRXzo7zHutC2qQUUQCRRRTKKcaB2B0ML
dlwFG4Meh4Tl+cWvqRGiFcvAk3eeyEvVE8J8nWwYzrRM/xDFXsQSWA==
-----END RSA PRIVATE KEY-----
```

- Publikus PEM formátumban

```
-----BEGIN                PUBLIC                KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArTG09wA8+G6KtQQ+GBtF
WDUPuW4YjINyS9qF0SI3wKKA6z3a3jiGGednrCIMb8DFqvKMHb1L0Ps68/BKawAs
G1CZ+10nHVwXSRfLWlpjNQYkIM37ZN1LI9NC5iHB03TZ20KjipDHPA2e+n64Wbxy
ybYMDrbGUtUj93RAvnXIKRIKE7LZobHgnmak65NYSKW0upihrq8FYHzIx1pL+Go+
b8P3+z7B15Kf4UipXrOFjjFquKKhAm9v+aZ+nwHWtiQJWdjVHLAsgkKOEURQnLnR
eM2jnuF0horNq42fryf70mfNj61yZykgb4yMUKTUqPX/IEac5V0H5ag0m5xq+I3
jwIDAQAB
-----END PUBLIC KEY-----
```

Tárolásuk:

PEM- RFC1421 szabványnak megfelelően, ezen kívül lehet még DER, mely binárisan tárolja az adatot. Illetve OpenSSH, melynél csak a publik kulcsokat lehet konvertálni, ez is szöveg jellegű. Mindegyik tárolás titkosítást takar, melyből visszafordítható az eredeti kulcs.

Második feladat:

Használt függvények:

- A hash objektumot az SHA256.new()-vel hozom létre, amellyel így a hashelés típusa SHA256 lesz
- A hash objektum értéket az update(%data)-val tudom frissíteni, mely egy bináris bemenetet vár, amit a json.dump, majd encodedal kapjuk meg a kívánt binaryt
- Az aláírás objektumot a pkcs1_15.new(key) hozza létre, melynek argumentuma a privát kulcsunk
- A kívánt üzenetből először hash-t csinállok az előbb használt függvénnyel, majd ezt a .sign-nal aláírom
- Ezt követően az aláírást jobb olvashatóság miatt b64 formátumra alakítom
- Az ellenőrzésnél az aláírás objektum a publikus kulcsból áll elő, a kész aláírást dekódolni kell a b64 formátumból, majd a verify függvénnyel levalidálni hogy az adott publikus kulcshoz jó-e
- A verify-nak nincs visszatérési értéke, csak ValueError hibát dob ha nem jó, ezt try-except-tel el lehet kapni

Json.dumps:

Bármilyen objektumot képes stringgé alakítani. Legyen az objektum, vagy bármi. Ezzel el tudjuk érni, hogy más enkódolású bemenetet is elfogadjon, ne legyen érzékeny az üzenet tartalmára. Ezért is szükséges, hogy minden üzenetet át tudjunk alakítani. Probléma vele hogy nem teljes kétirányú, és lehet több fajta bemenetre is ugyanolyan kimenetet kapni.

Lenyomat szerepe:

Az aláírás során saját privát kulcsunk segítségével készítünk lenyomatot az üzenetről. Ezt az aláírást mindenki ellenőrizheti a publikus kulccsal, viszont az enkriptálást (lenyomatot) csak a privát kulcs tulajdonosa hajthatja végre. Az ellenőrzésnél az üzenetet ismerve előállítják az ő lenyomatukat az üzenetről, melyet összevetnek a kapottal.