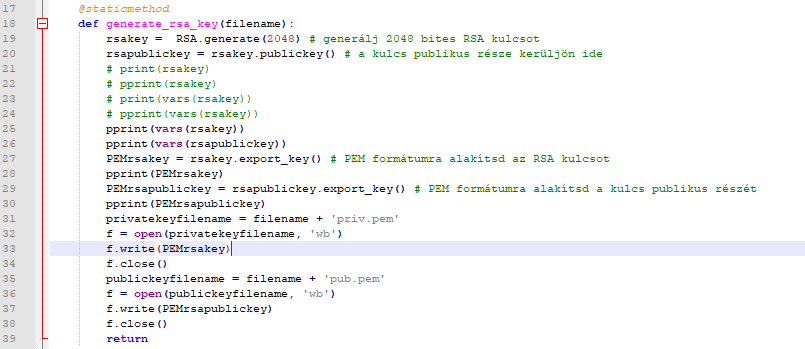
# Adatbiztonság és Kriptográfia 1. jegyzőkönyv

Készítette: Ütő Bence, V7Z3T5  
Gyakorlaton részt vett: 2020.03.05.  
Jegyzőkönyv elkészítése: 2020.03.08.

## Feladatok

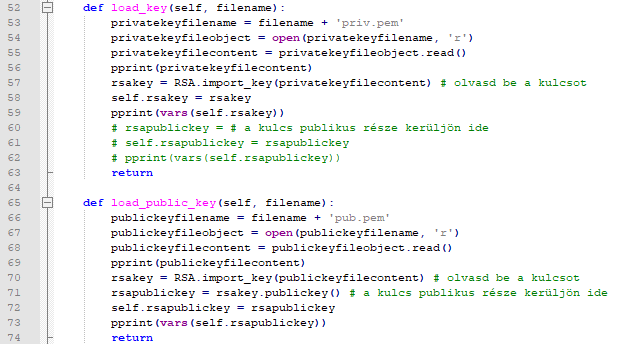
1. RSA privát és publikus kulcs generálás
2. Kulcsok betöltése és objektumhoz rendelése
3. Megadott adatok hashelésére szogáló függvény megírása
4. Digitális aláírás készítése és ellenőrzése

## RSA privát és publikus kulcs generálása



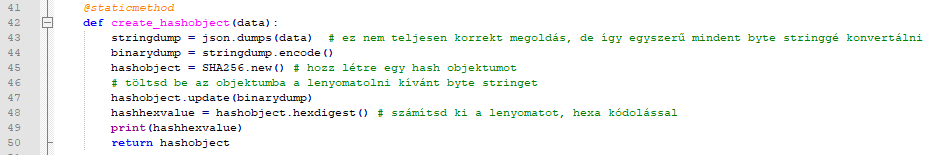
A generálás az RSA.generatekey(<2048>) függvény segítségével történt. A publikus kulcs ebből a .publickey()-el érhető el. Az export\_key() fügvénnyel lehet exportálni a kulcsokat valamilyen formátumra, alapértelmezetten ez a PEM forma. Ezután már csak ki kellett írni az objektumokat egy-egy fájlba.

## Kulcsok betöltése és objektumhoz rendelése



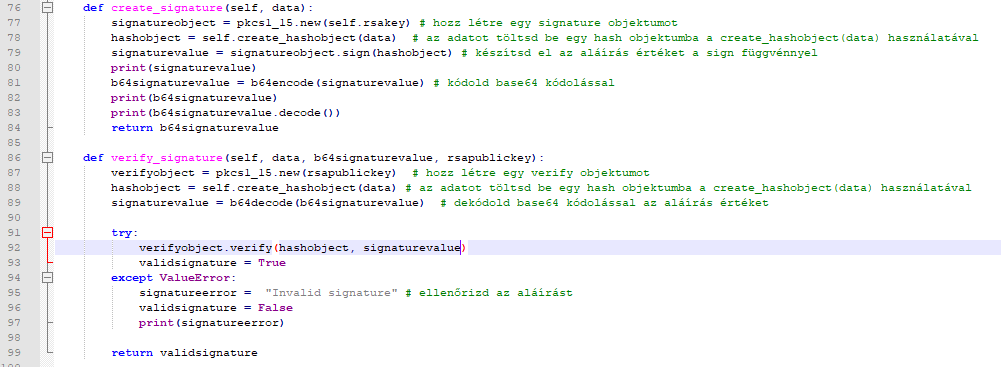
Ezek már nem statikus függvények, így a self paraméter segítségével az adott példányhoz rendelhetőek az értékek. A .import\_key(<tartalom>) függvénnyel a fájlból be lehet olvasni a privát kulcsot. A public key betöltése igen hasonló. Beolvassuk a kulcsot, majd a public\_key részét elmentjük.

## Hash függvény készítése



Hash objektumot az SHA256 segítségével csinálunk. Létrehozzuk, az SHA256 objektumunkat, beletöltjük a hashelni kívánt adatot bináris formában, majd a hash eredményét hexadecimális formára változtatjuk és ezzel térünk vissza a hívó számára.

## Aláírás készítése és ellenőrzése



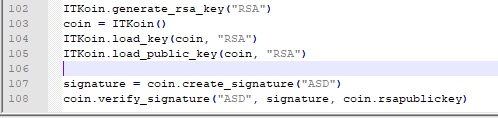
Aláírás készítése:

* pkcs1\_15 és a privát kulcs segítségével signature object készítése
* az adat lenyomatának elkészítése
* a lenyomat aláírása (privát kulcsot felhasználó signature object segítségével)
* base64 enkódolás a könnyebb olvashatóság érdekében

Aláírás ellenőrzése:

* pkcs1\_15 és a publikus kulcs segítségével signature object készítése
* az adat lenyomatának elkészítése
* az aláírás base64 dekódolása a feldolgozáshoz
* megpróbáljuk összehasonlítani (.verify függvény) a kapott aláírást és az általunk publikus kulccsal készített ellenőrző aláírást
  + ha megegyezik, az aláírás valid
  + ha nem ValueError adódik és az aláírás érvénytelen

## A konkrét függvényhívások



## Privát és publikus kulcsok tartalma

(b'-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEArc/GIYA30muJ92IdAu/L10QY' b'ZK0T9itd2Aj8uhDciK1yJQeF\ndfYgtgp6lXo7EYzS2+zLhJa0otnKNbTXON8CFSHhhi8mlkB' b'xEhPbvWJgIFZvKIca\nw1L1fgpiS6Sujcw1rlzcih3L0SFXvfbzMx/MvsPHGHBThDt1hRiXp5' b'gWf4CdiIdB\nu3uCfAMB/eNpMucGI0Gw0t1h5s7cAOgidW/FZYWrltBW9CTJk9R5GcrS+OAT8' b'c/7\neoYtiI9G1qnROiVBalU5i5Zv7GtvX/C0YDKutCV9YrSK9YRiyZJs0Cqr6XNT6IE8\n8qC' b'5VbKl4kNo75mgvLzO4CeWKV3qKA/FpAY0gQIDAQABAoIBAAe1ANoXb2I7NoDr\nrtGkhGtlL5' b'e3PP1bauvbailt+flth3Qw4zAZ29V2nXznUJs0nVz3IQceD6Bw+9Bp\n1mlFKjh5qiBjEtjyv' b'pDIpKO/KvgEHSW05/DlStFt8HHeOYc8GN1EbNsK+zCow9U+\nkAYiUNfcooVo5aslX86pCoB5' b'8/VvUcmJQsHvqRzlZtY9xxNcW62y8U60IBmZTkw5\nCYpLTdZCA9OZFttT0eegBq3OrHgEiRt' b'mEhBQpFxj3Xl8+C9BvZBHj9F+CuEsYbey\nzxZWucHe0M0pOzCVrpW/wrxlYf1pXwJ6M6oDPl' b'egnJaMYAfoI+FMCj0Y+mL2nN2q\n7Nqs2IkCgYEAwzVh09XUPF9fe0En+j3cvB6YOwtKSvjg3' b'nIxBLdiH2vU6egA9EZj\ngooyCBDYQuSlGNrrTFLprK0ycc4e9wldUgGaaMwVN0DXPFhpoB6c' b'ZwnYCy6ycETr\n8vWw5OsgchvWgnVq1YxgfpssafIUkPIhNijeoOtYP+Z+pdiBuCfmy3MCgYE' b'A4/CQ\ns4/QO6rz5CXcneYdOc3QIfIV7QVqNie0UWyUmJ+r/fkUlkPOjyWnhjEs20ADNOJ2\nQ' b'x42FDwRydB5yn1t1M2V+9IxYDSKm1M95GiKd4kg/N1SPRADGm1PQLOFWL38+rDS\njzLmhNOi' b'lM+A5VsCvfa3SHXZEpxJhdPjz1NzKzsCgYBQry/QLwGaMpF0kGnFtOwX\nlU22meQLOPM4/ln' b'kaSfVdOx8AbjjPnTC7/vd+ZhpjbfvngzDb0HGbY/8GgYSdXxs\n+NyplDDwtYxBFQoTdJOQwz' b'DY7mQyo0ojEgoblFPE1lzDb2TI/0yjB+BDqht2DZ5M\nlHfwa1X+k3aldeeoBlYOoQKBgQDNR' b'RhazFlfUp2eyUX7YH7TTnhwCX4bXHQA3xKw\n/RJ4u9p8tjIG/HtpSCSsYV6e9++rUYHlAbD7' b'2fi/ASKc6wXczWOTSwAyPj7QkE89\ngbOo4HI3pexolyHkS6pCfyjH+Rh4foKweMtFtP1F+/R' b'9F8/JWQRHmfFppBtGDMKs\nM9+UfwKBgFTjbN3VquSeG8ju/SOZywOPIx7SStRIHMfS193imo' b'YdTuGSsWLiJyKb\nhtUx2gmPMqN0Zfizhg94Q4nVBvMfreZU7IE72OtXGz8je46ICFXqZkx9+' b'ZPWnxQk\nc9J0iNjPD7iGktAvta6WkPTmM6ini6osMFf0zaxHAniLRjybZ1zX\n-----END RS' b'A PRIVATE KEY-----')

(b'-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr' b'c/GIYA30muJ92IdAu/L\n10QYZK0T9itd2Aj8uhDciK1yJQeFdfYgtgp6lXo7EYzS2+zLhJa0' b'otnKNbTXON8C\nFSHhhi8mlkBxEhPbvWJgIFZvKIcaw1L1fgpiS6Sujcw1rlzcih3L0SFXvfb' b'zMx/M\nvsPHGHBThDt1hRiXp5gWf4CdiIdBu3uCfAMB/eNpMucGI0Gw0t1h5s7cAOgidW/F\nZ' b'YWrltBW9CTJk9R5GcrS+OAT8c/7eoYtiI9G1qnROiVBalU5i5Zv7GtvX/C0YDKu\ntCV9YrSK' b'9YRiyZJs0Cqr6XNT6IE88qC5VbKl4kNo75mgvLzO4CeWKV3qKA/FpAY0\ngQIDAQAB\n-----E' b'ND PUBLIC KEY-----')

## JSON.dumps alkalmazása, probléma vele

Azért használjuk, hogy az objektumból byte stringet csináljunk. Nem szerencsés, mert overhead-je van az eredeti objektumhoz képest. Egyszerűen nagyobb. Esetleg lehetnek benne körkörös referálások (bár ez most minket aligha érint), amik végtelen hosszú JSON dumpot eredményeznének.

## Lenyomatok szerepe aláírás és ellenőrzés során

A lenyomatokat tudjuk összehasonlítani. Lenyomatolás nélkül bárki tudna a publikus kulcs segítségével üzenetet és aláírást kreálni. Így viszont a publikus kulccsal csak ellenőrizni lehet az aláírás hitelességét az adott üzenethez.