3460:435/535 Algorithms, Fall 2024

Project 1 --- RSA & Digital signature

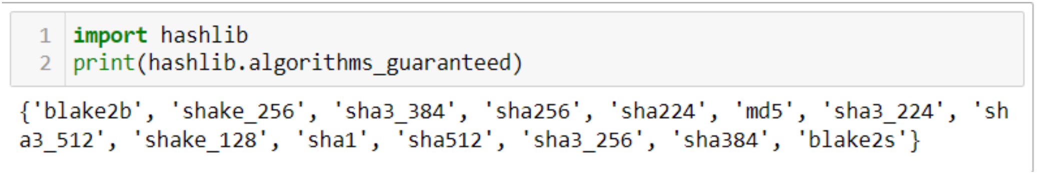
Your first programming assignment is to implement RSA Encryption and apply it to digital signature (http://en.wikipedia.org/wiki/RSA_(algorithm)). To facilitate the grading, you are to use **Python** to complete the project (see the attached template).

Part I: RSA key generation.

- Implement Fermat test;
- Use Fermat's test to generate two large prime numbers (p,q), each should have a size >= 512 bits;
- Save p and q in a file named p_q.csv, one integer per line and making sure no white space saved;
- Use the extended Euclidean algorithm to generate two pairs of keys: (e,n), (d,n), where n=p*q;
- Save the two pairs of keys in two separate files: e_n.csv and d_n.csv, one integer per line and no white space;

Part II: Generate and verify digital signatures using a SHA-256 hash.

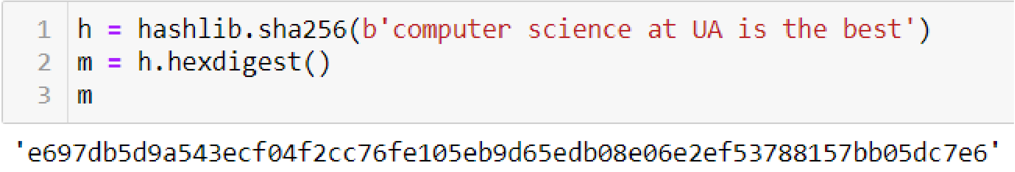
- Unless you want to implement the has function yourself, check if you have Python hashlib package.



```
import hashlib
print(hashlib.algorithms_guaranteed)
```

```
{'blake2b', 'shake_256', 'sha3_384', 'sha256', 'sha224', 'md5', 'sha3_224', 'sh
a3_512', 'shake_128', 'sha1', 'sha512', 'sha3_256', 'sha384', 'blake2s'}
```

- You will use sha256:



```
h = hashlib.sha256(b'computer science at UA is the best')
m = h.hexdigest()
m
```

```
'e697db5d9a543ecf04f2cc76fe105eb9d65edb08e06e2ef53788157bb05dc7e6'
```

- Sign a given file
  - Generate a SHA-256 hash of the content of the file to be signed (e.g., "file.txt");
  - Sign/"decrypt" this hash value using the private key stored in d_n.csv;
  - Combine the original content and the signature into one document filename.signed (e.g. "file.txt.signed"). Append the 32-byte signature (256/8=32) at the end of the original content.

- Verify the signed file
  - Separate the signature from the content of the file in the signed document (e.g. "file.txt.signed");
  - Generate a SHA-256 hash of the content of the file you have signed.
  - Check if the signature (old hashcode/m) = new SHA-256 hashcode/m.

What to submit.

1. **Submit your python program.** Make sure to test your code on more than one set of data. DO NOT submit programs that are not *reasonably correct*! To be considered *reasonably correct*, a program must be completely documented and work correctly for sample data provided with the assignment.
2. This is a classroom project. I see one file is enough. Use the template to get it started.

Grading. Your code will be graded **on correctness**, efficiency, clarity, and elegance.