Cryptography

1 USED LIBRARIES

RSA (installation required): provides a robust implementation of RSA encryption and digital signatures in Python. It's widely used and has been around for quite some time, so it's well-tested and trusted by the Python community;

PyAESCrypt (installation required): offers a straightforward interface for AES encryption and decryption in Python. AES is a well-established encryption standard recommended by government agencies and widely adopted in various industries for its security and efficiency. PyAesCrypt simplifies AES usage in Python applications;

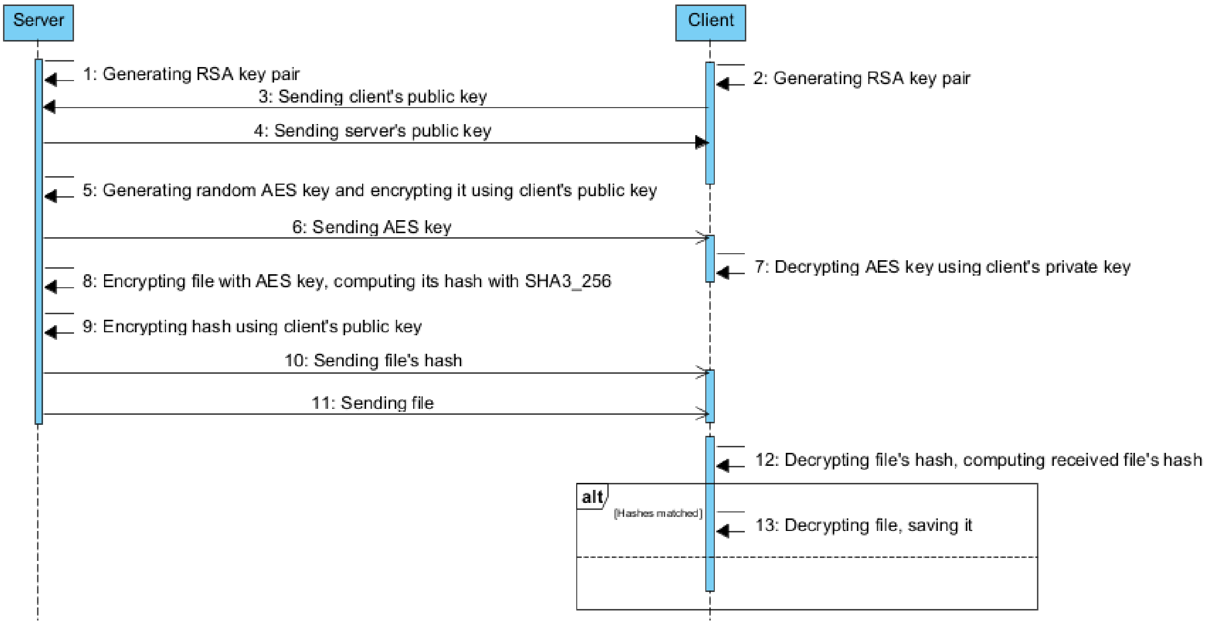
HashLib: provides a consistent interface for working with hash functions in Python. It includes implementations of various cryptographic hash algorithms, including the SHA family (which must be used). These algorithms are widely used and trusted in cryptographic applications for generating hash values and ensuring data integrity;

OS: a default library that has many useful functions;

Socket: library needed to emulate server-client architecture;

Time: useful library to measure execution time;

2 SEQUENCE DIAGRAM


Server
Client
1: Generating RSA key pair
2: Generating RSA key pair
3: Sending client's public key
4: Sending server's public key
5: Generating random AES key and encrypting it using client's public key
6: Sending AES key
7: Decrypting AES key using client's private key
8: Encrypting file with AES key, computing its hash with SHA3_256
9: Encrypting hash using client's public key
10: Sending file's hash
11: Sending file
12: Decrypting file's hash, computing received file's hash
alt
[Hashes matched]
13: Decrypting file, saving it

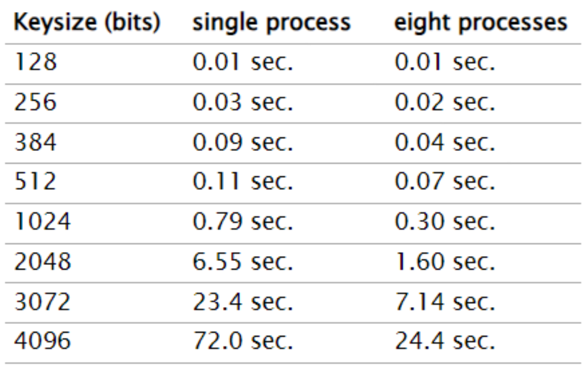
3 CHOICES EXPLAINATION

3.1 RSA KEY PAIR

RSA (Rivest-Shamir-Adleman) is one of the most widely used asymmetric cryptographic algorithms. It's based on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem."

A 1024-bit key size was chosen as it seems to be a good compromise between security and efficiency.

(https://stuvel.eu/python-rsa-doc/usage.html)



Keysize (bits)	single process	eight processes
128	0.01 sec.	0.01 sec.
256	0.03 sec.	0.02 sec.
384	0.09 sec.	0.04 sec.
512	0.11 sec.	0.07 sec.
1024	0.79 sec.	0.30 sec.
2048	6.55 sec.	1.60 sec.
3072	23.4 sec.	7.14 sec.
4096	72.0 sec.	24.4 sec.

However, 1024-bit size is a bit depreciated, a key of at least 2048-bit size is recommended though it'll be overkill in this exercise and would dime execution speed. (which usually goes around 3.2s: time library).

3.2 AES KEY

AES (Advanced Encryption Standard) is a symmetric encryption algorithm used for securing sensitive data. It's widely adopted due to its security, efficiency, and versatility.

AES is a widely trusted encryption standard that provides strong security and efficient performance, making it a cornerstone of modern cryptography. It's a crucial component of secure communication systems and data protection mechanisms in various industries and applications.

The test file used in the exercise is not big, so a key size of 256-bit, which is the highest security using AES, where a 128-bit key size is usually recommended, can be used.

3.3 HASH (SHA-3)

A hash is a fixed-size string of bytes generated by applying a hash function to an input data. Hash functions are mathematical algorithms that take an arbitrary amount of input data and produce a fixed-size string of bytes, typically representing a unique "digest" of the input data.

Hash functions are used in cryptographic applications for generating digital signatures, creating message authentication codes, and securely storing passwords. Cryptographic hash functions are designed to meet specific security requirements, such as collision resistance.

There are several hash functions, though in this exercise SHA-3 utilisation is asked.

> SHA-3, which stands for Secure Hash Algorithm 3, is a cryptographic hash function family standardized by the National Institute of Standards and Technology (NIST).

In SHA-3 vulnerabilities of SHA-2 have been fixed, but none of these were major vulnerabilities, which means SHA-2 could have also been used and is still used in 2024. (https://stackoverflow.com/questions/14356526/whats-the-difference-between-the-hash-algorithms-sha-2-and-sha-3)

SHA-3 256 is used, SHA-3 256 is a widely adopted cryptographic hash function that offers strong security properties and efficient performance.