Assignment3
SIL-765 Network & System Security

-By Bipul Kumar & Ayush Gupta

Problem Statement

- upload the document to server (or some version thereof) and expect to receive the same but with the current date and time stamped onto the document
- time-stamp document (in some standard format) with the current GMT data/time and a digital signature
- it should be possible to establish the fact that the **document existed at the date/time stamped**, and that the document has **not been modified**

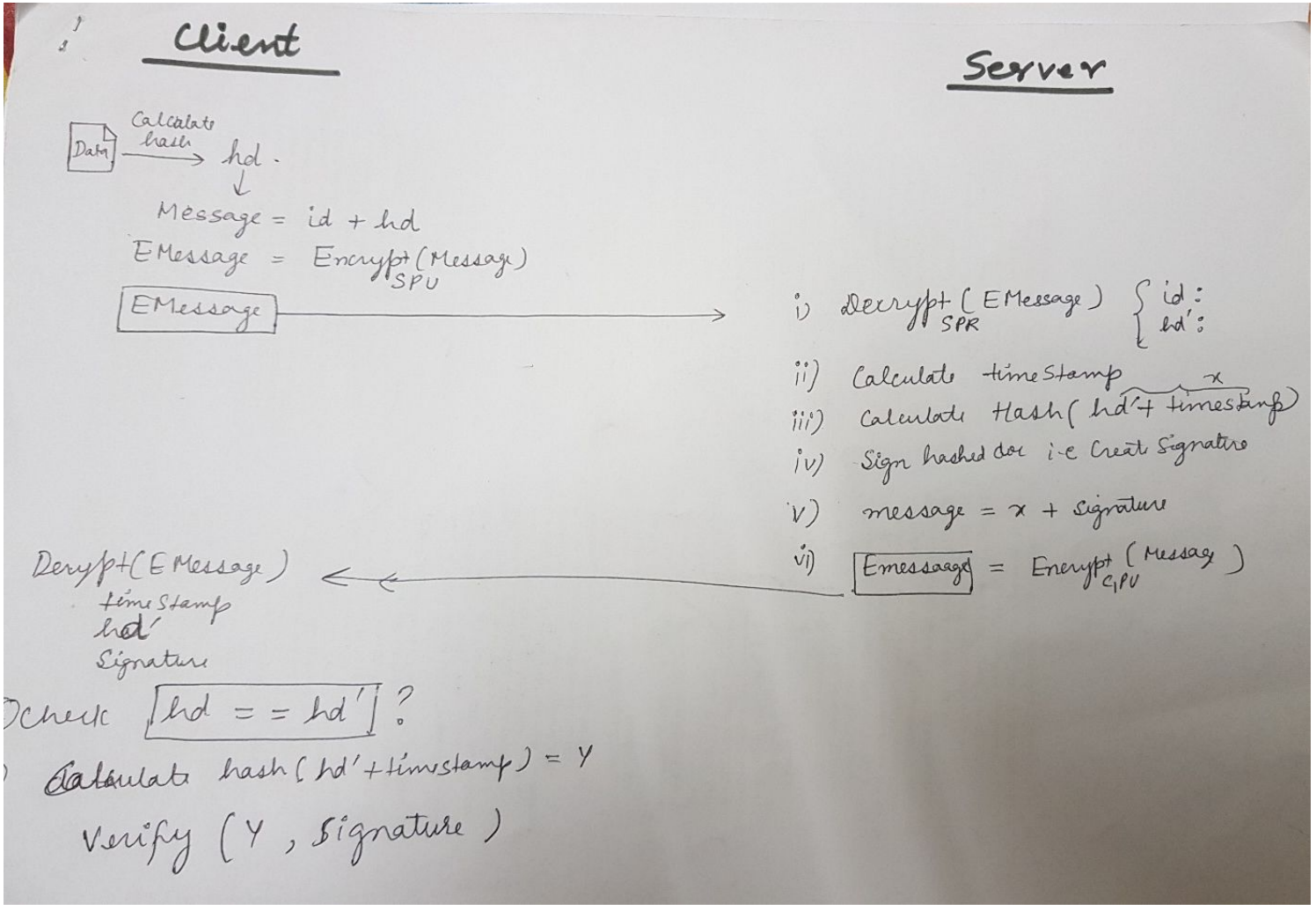
Key Generation

- Generating a private/public key pair using RSA algorithm with pycrypto
- Specify the key size as 1024bit for server key and 4096 bit for client’s keys. Larger is more secure
- Use Random module of pycrypto to specify a random number generator function

Public Key Directory

- Individual user will register their public key in public directory with its id .
- Individual user control or update their public key in secure manner.

Flow Chart


Client
Server
Data
Calculate hash
hd
Message = id + hd
EMessage = Encrypt (Message) SPU
EMessage
i) Decrypt (EMessage) SPR { id: hd':
ii) Calculate timeStamp
iii) Calculate Hash(hd' + timestamp) x
iv) Sign hashed doc i.e Creat Signature
v) message = x + signature
vi) Emessage = Encrypt (Message) CPU
Decrypt(EMessage)
timeStamp
hd'
Signature
Check hd == hd' ?
Calculate hash(hd' + timstamp) = Y
Verify (Y, Signature)

How and where do you get the correct GMT date and time? And how often?

- We use the “time.ctime” module to get the correct GMT date and time. It’s used in Server to add to the contents of the file before hashing it.

Is the source reliable and the GMT date and time obtained in a secure manner?

- Yes, the module fetches time from the OS of the device on which its run, and this fetching is done at the server. We assume that Server has not become malicious and is secure.

How do you ensure privacy, in that the server does not see/have/keep the original document?

- Encrypted hash of the message (using server public key) is sent to server. This ensures that the information is only received by server and message is not tempered by server.
- In order to ensure that message not changed, server sent original message along with signature, on receiving message it compare original hash and received hash.

How do you share the document with others in a secure manner with the date/time preserved,and integrity un-disturbed?

- Client1 send encryption of message + timestamp +signature received to client2.
- Client to verify the signature which ensure that message is not changed.

Also ensure that the user has (and uses) the correct “public-key” of the GMT date/timestamping server

- For this , we have created public key directory which contain public key of all client and server.