INTRODUCTION

This experiment was developed by Saisrinivasa Likhit Kota (2018101043) and Subodh Sondkar (2018101064) as part of the requirements for Introduction to Software Systems Assignment-4(Spring 2019).
Digital Signature is an authentication mechanism that enables the creator of the message to attach a code,i.e digital signature.

Signature is generated by hashing the plaintext message and encrypting it with the creator's private key.This protects the message from being read by foreign entities other than those possessing the public key, and also ensures the source and integrity of the message.

ABOUT

This experiment teaches the student the method to verify the integrity of the message in public key setting. The student must then verify why the digital signature scheme works.

DIGITAL SIGNATURES

Key Generation Algorithms :
Digital signature are electronic signatures, which assures that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise the data can be altered or someone can also act as if he was the sender and expect a reply.
Signing Algorithms:
To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm

is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter fixed length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing.
Signature Verification Algorithms :
Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. Then the hash value and the output of the verification algorithm are compared. If they both are equal, then the digital signature is valid else it is invalid.

SIGNATURE GENERATION

message data --> hash function --> message digest --> signature generation using private key

Message digest is computed using one-way hash function, i.e. a hash function in which computation of hash value of a is easy but computation of a from hash value of a is very difficult.

SIGNATURE VERIFICATION

message data --> hash function --> message digest --> signature verification using public key

KEY GENERATION

before signing a message the signee must generate keys and announce the public key to the public

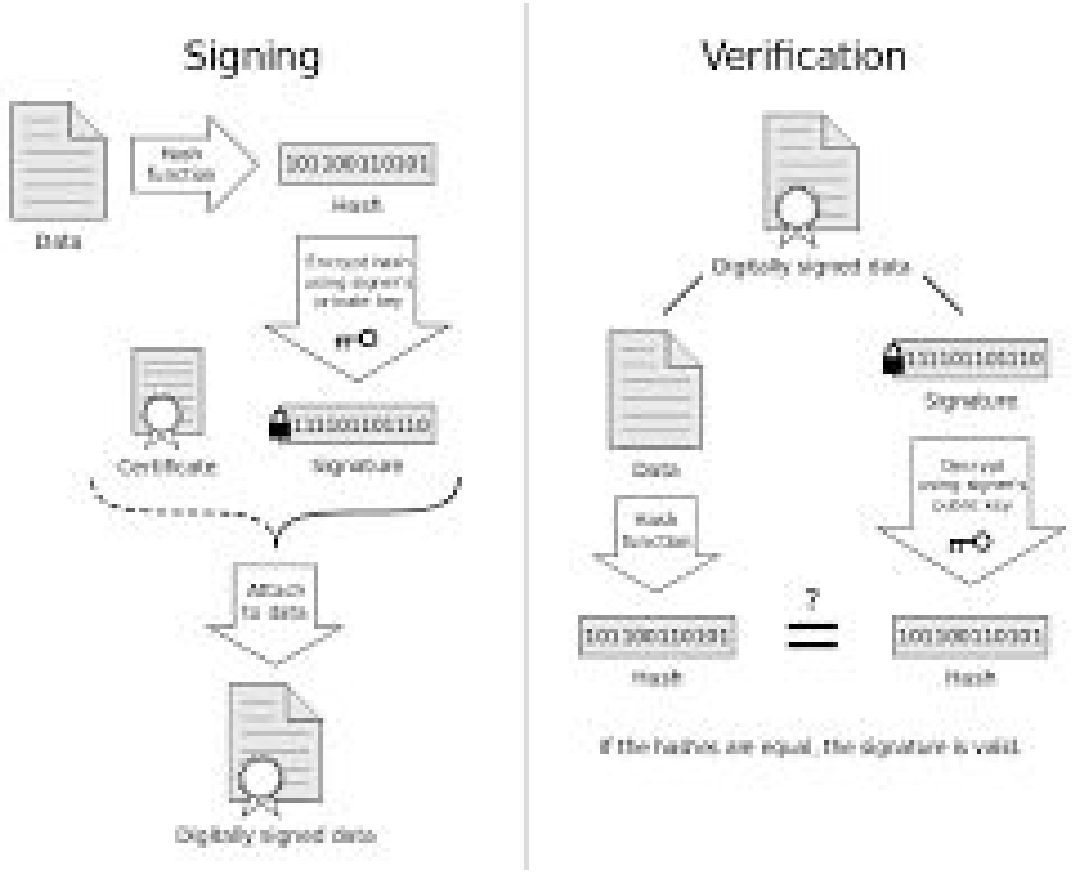
Process

- choose prime p, between 512 and 1024 bit length ; number of bits in p must be multiple of 64
- choose 160-bit prime q such that q divides (p-1)

- choose primitive root e0 in Zp
- create e₁ such that e₁ = e0(p-1)/q mod p
- choose d as private key and evaluate e2 = e1*d
- PUBLIC KEY: Pbk = (e1,e2,p,q)
- PRIVATE KEY: Prk = d

Creating Digital Signature

- Message digest is computed by applying hash function on the message and then message digest is encrypted using private key of sender to form the digital signature. (digital signature = encryption (private key of sender, message digest) and message digest = message digest algorithm(message)).
- Digital signature is then transmitted with the message.(message + digital signature is transmitted)
- Receiver decrypts the digital signature using the public key of sender.(This assures authenticity,as only sender has his private key so only sender can encrypt using his private key which can thus be decrypted by sender's public key).
- The receiver now has the message digest.
- The receiver can compute the message digest from the message (actual message is sent with the digital signature).
- The message digest computed by receiver and the message digest (got by decryption on digital signature) need to be same for ensuring integrity.



CODE DESIGN

DIRECTORY STRUCTURE and FILES

- static -- contains css, images, and fonts
- dynamic -- contains js
- templates
  - base.html -- the basic structure of all webpages in webapp
  - experiment.html -- contains the experiment module
  - further_readings.html -- further reference material for digital signatures
  - introduction.html -- mainpage for experiment
  - theory.html -- contains theory necessary for experiment
  - manual.html -- experiment manual for the experiment
  - procedure.html -- execution steps for the experiment
  - objective.html -- describe aim of experiment
  - quiz.html -- questions related to the experiment
- __init__.py -- initialize flask object and import all libraries needed for the webapp

- routes.py -- decorators used, functions for template rendering based on webpage requested
- digital_signatues.py -- main webapp
- requirements.py -- python libraries used
 - flask v1.0.2
 - flask-sqlalchemy v2.3.0