SIL 765

UID

Harsimrat Singh
2015CS50284

Sudeep Agarwal
2015CS50295

Introduction -

The ***UID*** (Unique Identifier) is a project initiated *Government of India* for providing a unique identification number to every citizen of India. Each citizen is provided an ***Aadhar No.*** which is mapped to all the personal details like Name, Address, ,Phone, Finger prints, photograph etc. Auth API allows validation of identity claim by the *Aadhaar* holder. You can take the input from the user about any person and validate if it is "true" or "false".

Working of the system

The client request should contain the following fields -

```
{
     “data”: {
          “Aadhar No.”:
          “Field_to_be_verified_eg._dob”:
     }
}
```

The client request is encrypted using public key of the the server which ensures ***confidentiality*** as only server can decrypt the message using his private key.

At server, the cipher is first decrypted using server’s private key and the request fields are retrieved. The server then verifies the information in it’s database and then creates a certificate with the answer and the hash of the request and sends it back to client.

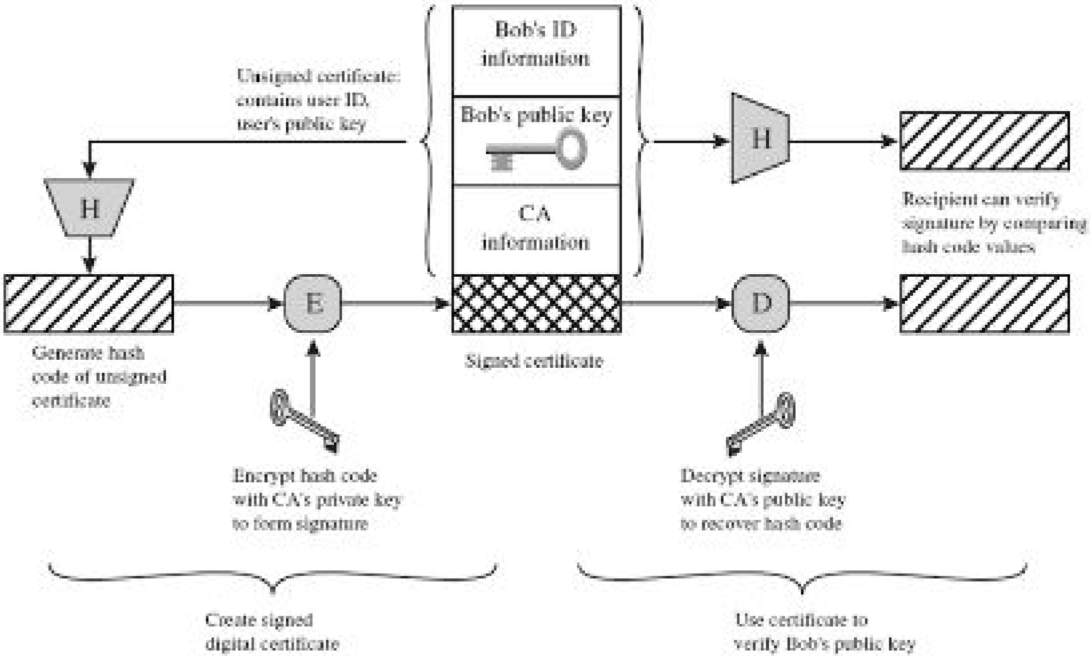
Bob's ID information
Bob's public key
CA information
Unsigned certificate: contains user ID, user's public key
H
H
Recipient can verify signature by comparing hash code values
E
D
Generate hash code of unsigned certificate
Signed certificate
Encrypt hash code with CA's private key to form signature
Decrypt signature with CA's public key to recover hash code
Create signed digital certificate
Use certificate to verify Bob's public key

Figure 1

Authentication - No authentication is required in this scheme as anyone can send the data verification request to server.

Confidentiality - Confidentiality is ensured as the request is encrypted using server's public key and only server can decrypt information using it's private key.

Message Integrity - Message integrity from server to client is ensured as server is sending certificates to client. From client to server, if an adversary alters the request or creates a new request, the certificate would contain hash of altered request and it won't match with client's request hash.

Non repudiation - Non repudiation is not required in this case since anyone can request a server and denial of client doesn't affect server.

Solution to problems in our project

1) For ensuring that information is not altered between server and client, the server can send a certificate and client can verify whether the data in the certificate has been altered or not. If the hash of certificate and decrypted signature are same then one can say that information is not altered.

2) Along with the answer as “Yes” or “No”, server will also send the encrypted hash of question asked so that client can verify the question asked.

3) Digital Signatures are relevant to ensure that certificate has been generated by *UID* server. Hash value along with the certificate is encrypted with the server’s private key. Hence, adversary cannot generate certificate without the private key of server. Client can view certificate generated by server by decrypting it with server’s public key.

4) No, access to “public-key certificate” issued by a certification authority won’t be an issue because the certificate contains only the answer to the question and the hash of the question. In any case the details of the question asked are not revealed.