

UID Detail Verification

by

SHALINI PANWAR

Entry No. 2016MCS2681

and

MOHD SAOOD SHAKEEL

Entry No. 2012PH10850

MASTER OF TECHNOLOGY

in

Computer Science & Engineering

Introduction

The origin of given project lies in UID project of Government of India, where a central server can be accessed to determine whether some information on an individual is correct or not but without divulging the information itself.

This can be done in secure manner by basically taking all details as input from user. These details include UID, DOB, Father's Name and Name of the user (where UID is mandatory field). These details are then sent through an encrypted link to the server. The server then receives these details. These are then checked into the server database. The server then replies back to the client that whether the details are correct or not. This response is digitally signed by the server and further sent back to the client through the same encrypted link. The client receives response and checks if the details are sent by the assigned authority by verifying it with the help of digital signature present in the response received from the server.

The encrypted link is set up by **SSL Certification**. The server then hashes the response with **SHA256** and encrypts it with RSA algorithm using its own private key to create the digital signature. This digital signature is then sent along with the Yes/No response through the encrypted link. This response is received by client and the digital signature is verified for authentication and integrity.

SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client (e.g., Outlook).

The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash.

RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret.

SSL:Secure Socket Layer

SSL (Secure Sockets Layer) is a standard security technology for establishing an **encrypted link** between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client (e.g., Outlook).

SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely. More specifically, SSL is a security protocol. The SSL protocol determines variables of the encryption for both the link and the data being transmitted.

SSL Certificates have a key pair: a public and a private key. These keys work together to establish an encrypted connection. The certificate also contains what is called the “subject,” which is the identity of the certificate/website owner.

When a browser attempts to access a website that is secured by SSL, the browser and the web server establish an SSL connection using a process called an “**SSL Handshake**”.

Essentially, **three keys** are used to set up the SSL connection: the public, private, and session keys. Anything encrypted with the public key can only be decrypted with the private key, and vice versa.

Because encrypting and decrypting with private and public key takes a lot of processing power, they are only used during the SSL Handshake to create a symmetric session key. After the secure connection is made, the session key is used to encrypt all transmitted data.

This process is as follows:

- Browser connects to a web server (website) secured with SSL (https).
- Browser requests that the server identify itself. Server sends a copy of its SSL Certificate, including the server’s public key.
- Browser checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server’s public key.
- Server decrypts the symmetric session key using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session.
- Server and Browser now encrypt all transmitted data with the session key.

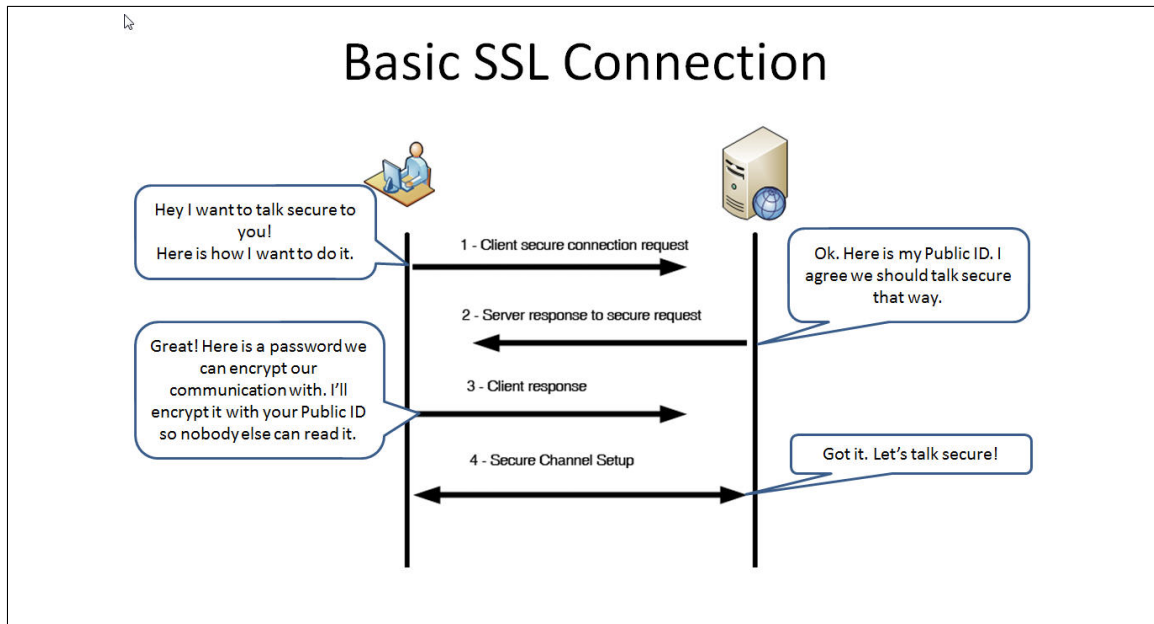


Figure 1: SSL encryption

In Our project, SSL certificate is installed at server. Whenever a client tries to connect to the site hosted at the server, the connection becomes secure through SSL and any communication between client and server becomes secure because of the encrypted link made.

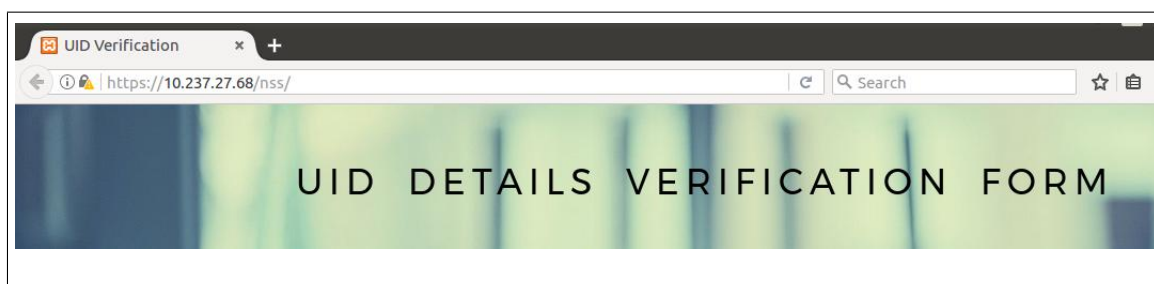
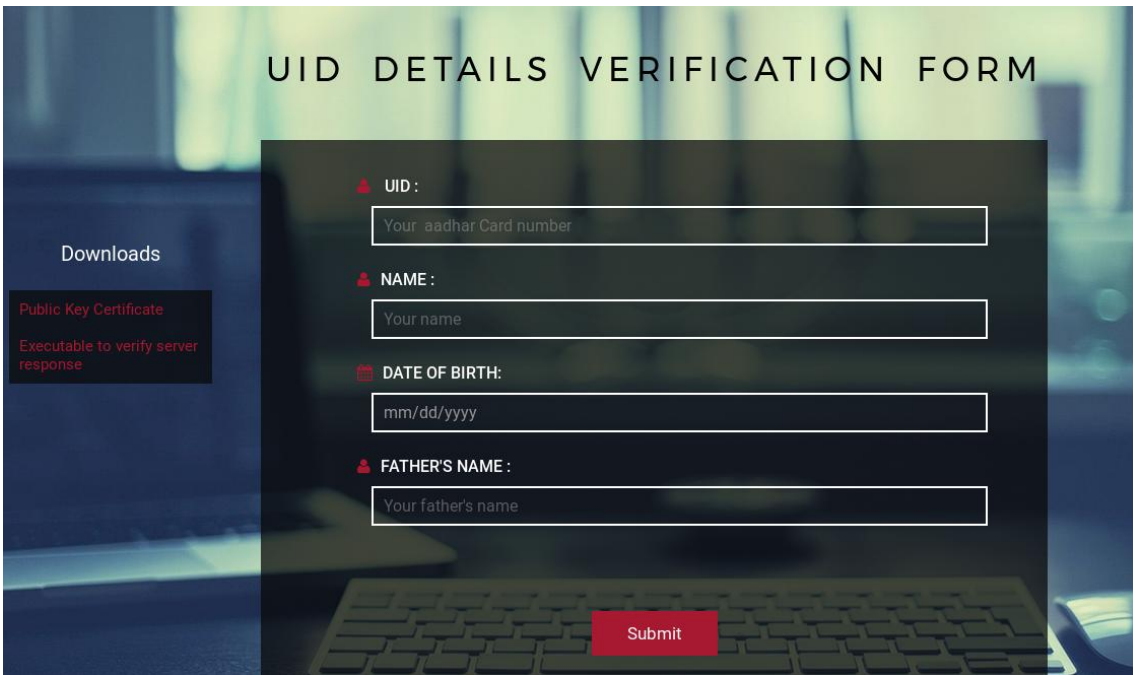


Figure 2: SSL encryption implemented in project



The image shows a web application interface for UID verification. The main heading is "UID DETAILS VERIFICATION FORM". On the left, there is a "Downloads" section with a link to "Public Key Certificate" and a note that it is an "Executable to verify server response". The main form area contains four input fields: "UID:" (with a person icon) for "Your aadhar Card number", "NAME:" (with a person icon) for "Your name", "DATE OF BIRTH:" (with a calendar icon) for "mm/dd/yyyy", and "FATHER'S NAME:" (with a person icon) for "Your father's name". A red "Submit" button is located at the bottom right of the form. The background is a blurred image of a laptop keyboard.

UID DETAILS VERIFICATION FORM

Downloads

Public Key Certificate
Executable to verify server response

UID :
Your aadhar Card number

NAME :
Your name

DATE OF BIRTH:
mm/dd/yyyy

FATHER'S NAME :
Your father's name

Submit

Figure 3: Login form for entering details at client side

So, all the details entered by user at first page are sent in encrypted form to the server which remains unaltered during transmission. Thus, this ensures that information is not altered during the 2-way communication between the client and server.

Digital Signature

Digital Certification is the act of certifying the accuracy of a document. When a document is digitally certified, its page content is locked to prevent changes. The certifier can opt to allow limited changes that do not affect this content (for example, adding markups, completing form fields or applying digital signatures). Public key cryptography, or asymmetric cryptography, is any cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner. Digital signatures are also based on public key cryptography.

About

Digital signatures are based on **public key cryptography**, also known as asymmetric cryptography. Using a public key algorithm such as **RSA**, one can generate two keys that are mathematically linked: one private and one public. To create a digital signature, signing software (such as an email program) creates a **one-way hash** of the electronic data to be signed. The private key is then used to encrypt the hash. The encrypted hash – along with other information, such as the hashing algorithm – is the digital signature. The reason for encrypting the hash instead of the entire message or document is that a hash function can convert an arbitrary input into a fixed length value, which is usually much shorter. This saves time since hashing is much faster than signing.

The value of the hash is unique to the hashed data. Any change in the data, even changing or deleting a single character, results in a different value. This attribute enables others to validate the integrity of the data by using the signer's public key to decrypt the hash. If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed. If the two hashes don't match, the data has either been tampered with in some way (integrity) or the signature was created with a private key that doesn't correspond to the public key presented by the signer.

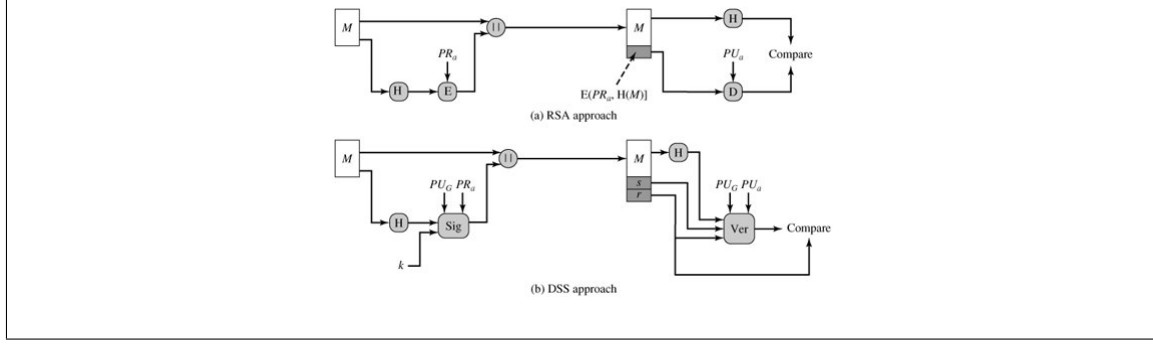


Figure 4: Digital Signature

In our project the server first checks and verifies the details sent by the user into its database. The server then sends reply accordingly by creating the digital signature of the response and sending it along to ensure authentication and integration of the response. This digital signature is generated by generating hash using SHA256 and further encrypting it with RSA using its own private key.

SHA256

The Secure Hash Algorithm is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS), including:

- SHA-0: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA".
- SHA-1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm.
- SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32 byte words where SHA-512 uses 64 byte words.

SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the National Security Agency (NSA). Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of

cryptographic hash functions is their collision resistance: nobody should be able to find two different input values that result in the same hash output.

SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function – it cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures.

SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available.

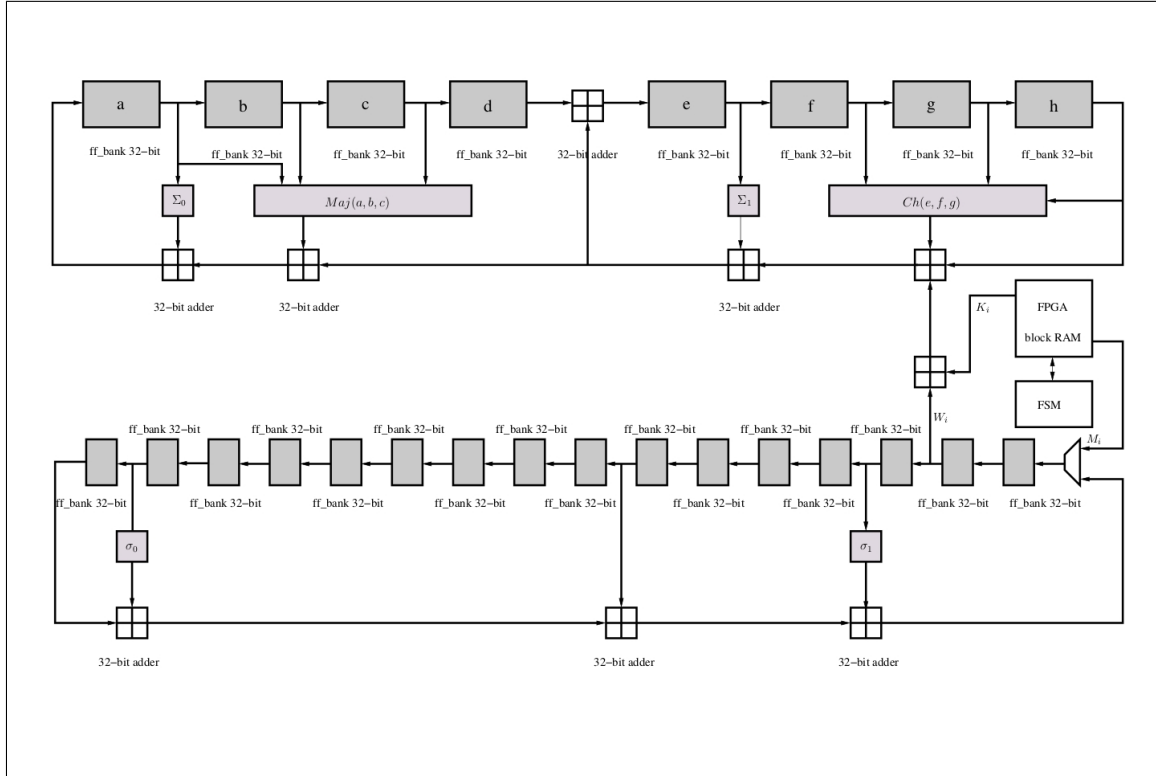


Figure 5: SHA256 Hash Algorithm

The Encryption Algorithm: RSA

RSA is a crypto system for public-key encryption, and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet. RSA was first described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology. Public-key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys, one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret. In RSA cryptography, both the public and the private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason

why RSA has become the most widely used asymmetric algorithm: It provides a method of assuring the confidentiality, integrity, authenticity and non-reputability of electronic communications and data storage. RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total – factoring – is considered infeasible due to the time it would take even using today’s super computers.

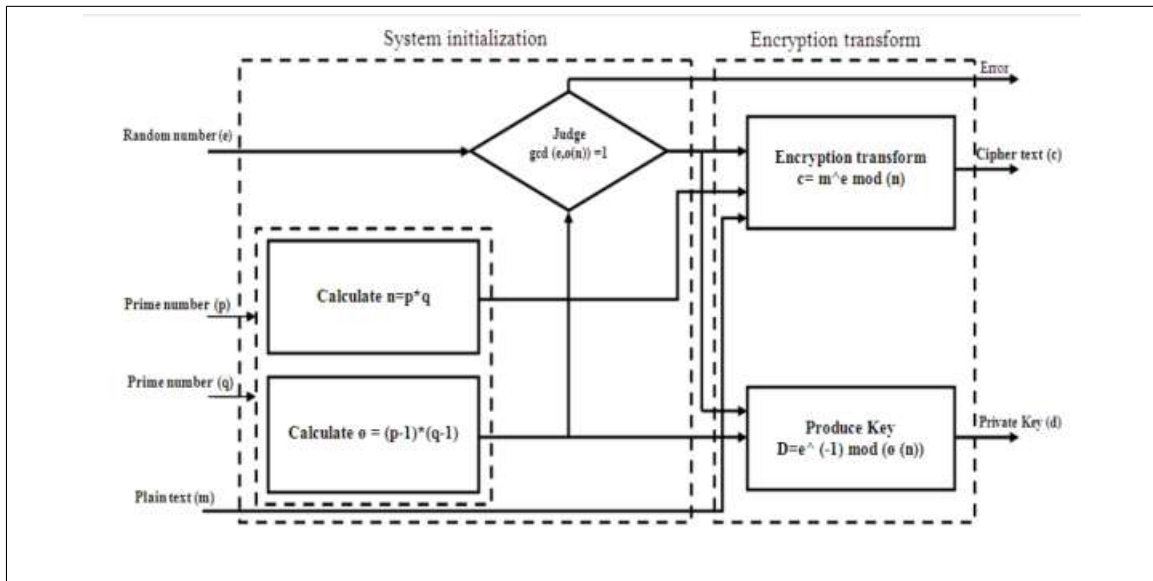


Figure 6: RSA Algorithm

After the digital signature is created using SHA256 hash and RSA encryption, the response from server is sent to the client along with the digital signature through the encrypted link formed by SSL certification.

The client is redirected to following page if the UID entered by user is wrong:

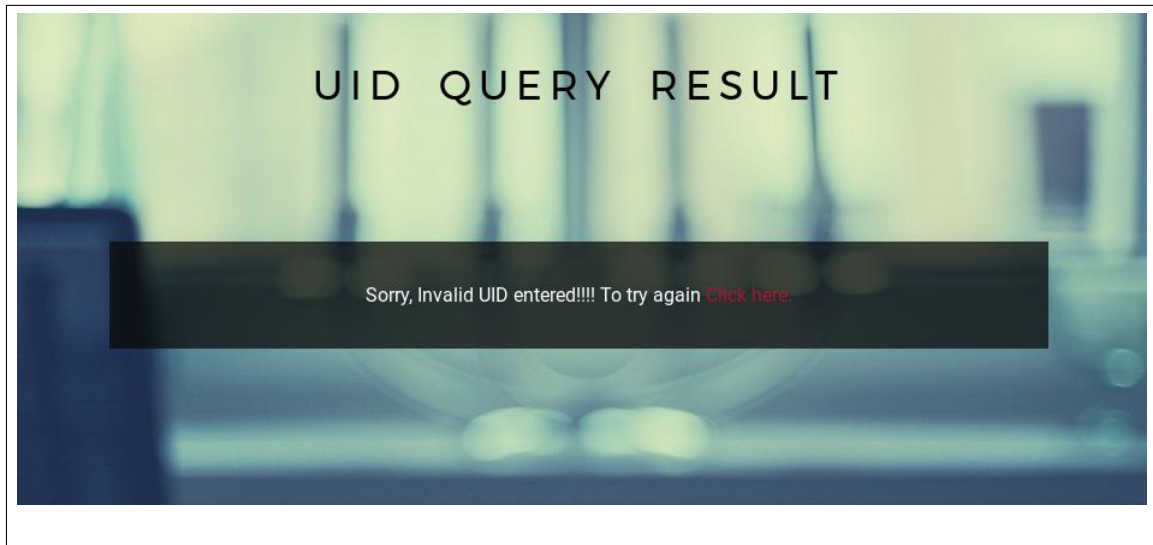


Figure 7: Error Page: If the UID entered by user is wrong

The client is redirected to following page if UID entered is corrected so as that user can download response of server from here.

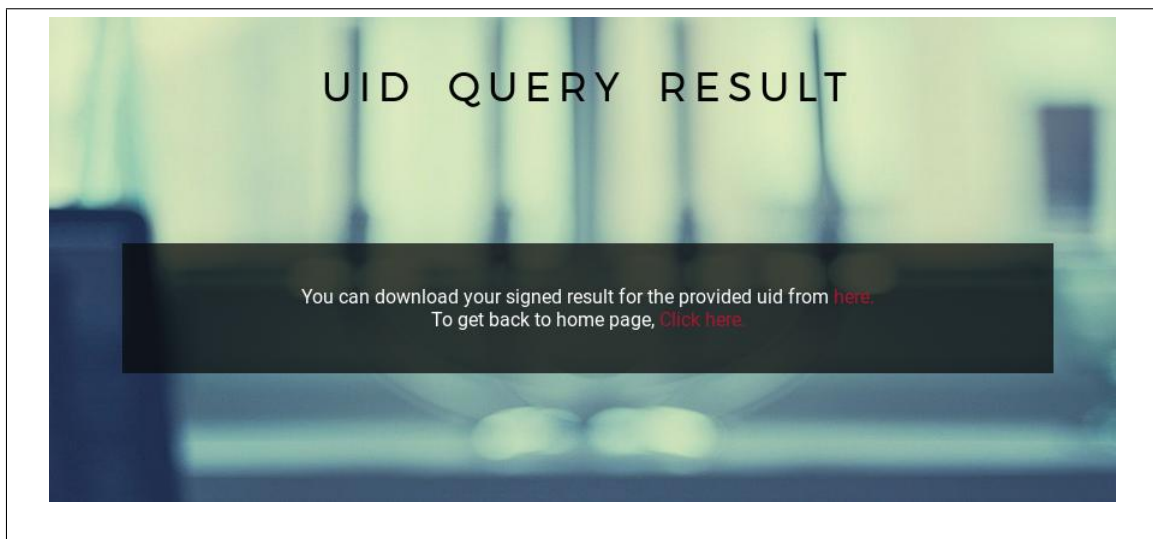


Figure 8: Fwd Page: If the UID entered by user is wrong

Verifying the Response

After the server response is downloaded by the client, the client can check the response by running the executable present on index page using public key present on index page. In the signed response is taken and its digital signature is separated from the response. This signature is then decrypted using **RSA algorithm** using the **public key** of the authorized organization already present with the client from index page. Then the hash of the document (from which digital signature is removed) using **SHA256** is calculated. This is done through the following GUI:

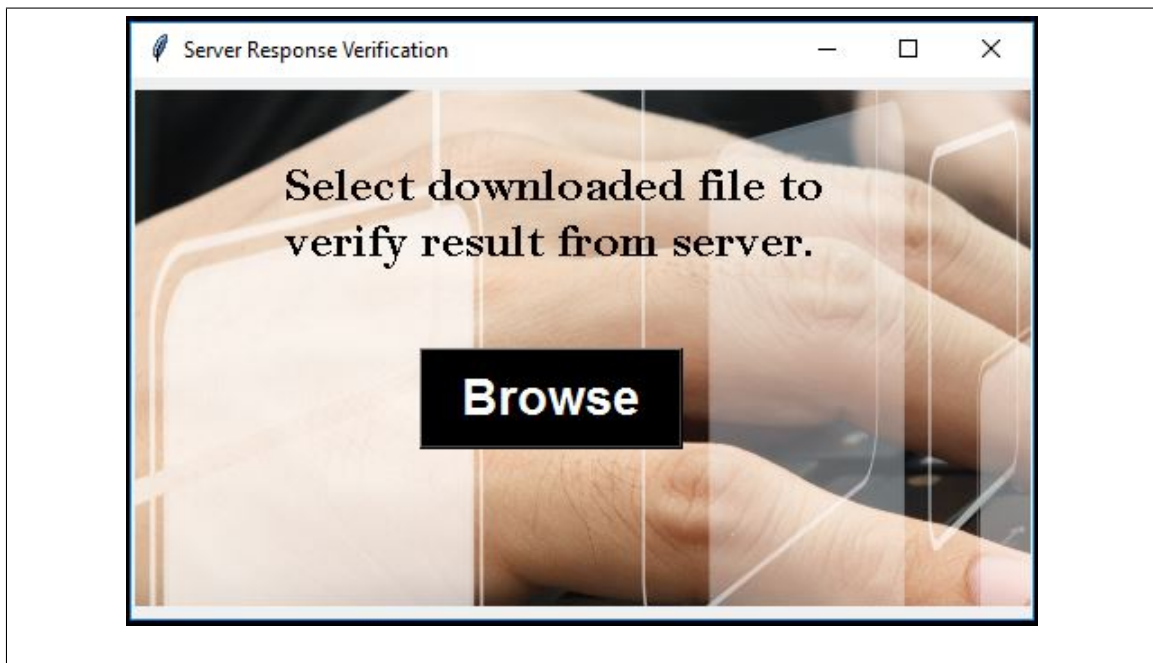


Figure 9: GUI for checking server response

This hash is then compared with the decrypted digital Signature. If the two values are equal the system returns the following server response.

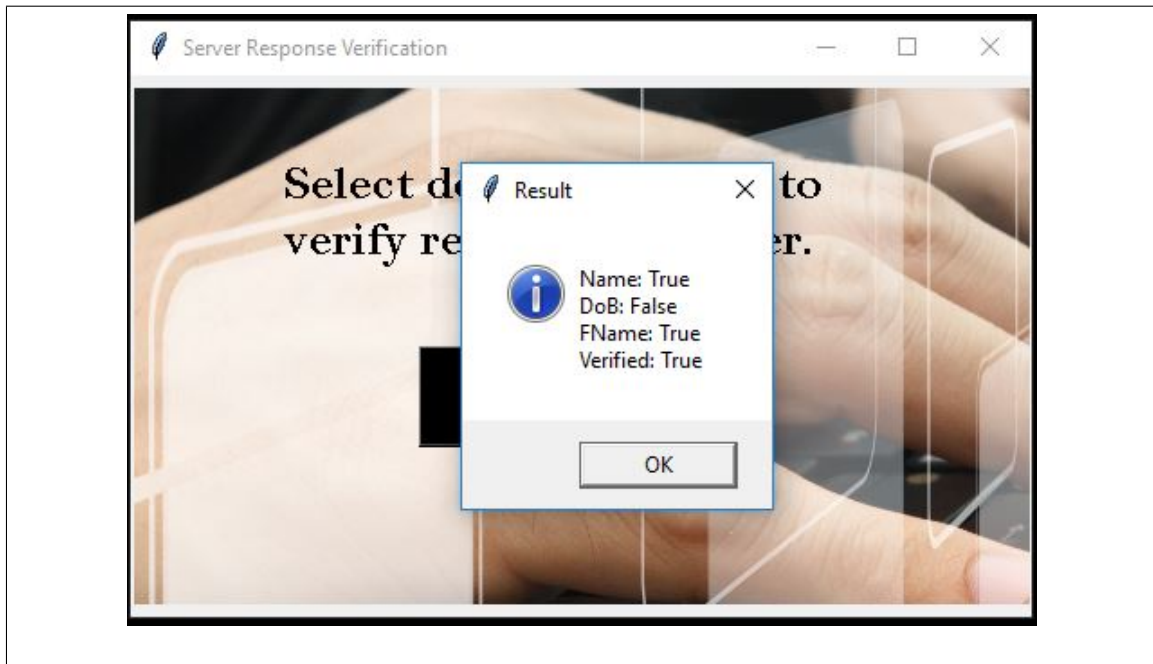


Figure 10: System response if Signature is verified

If the two values are different the system following:

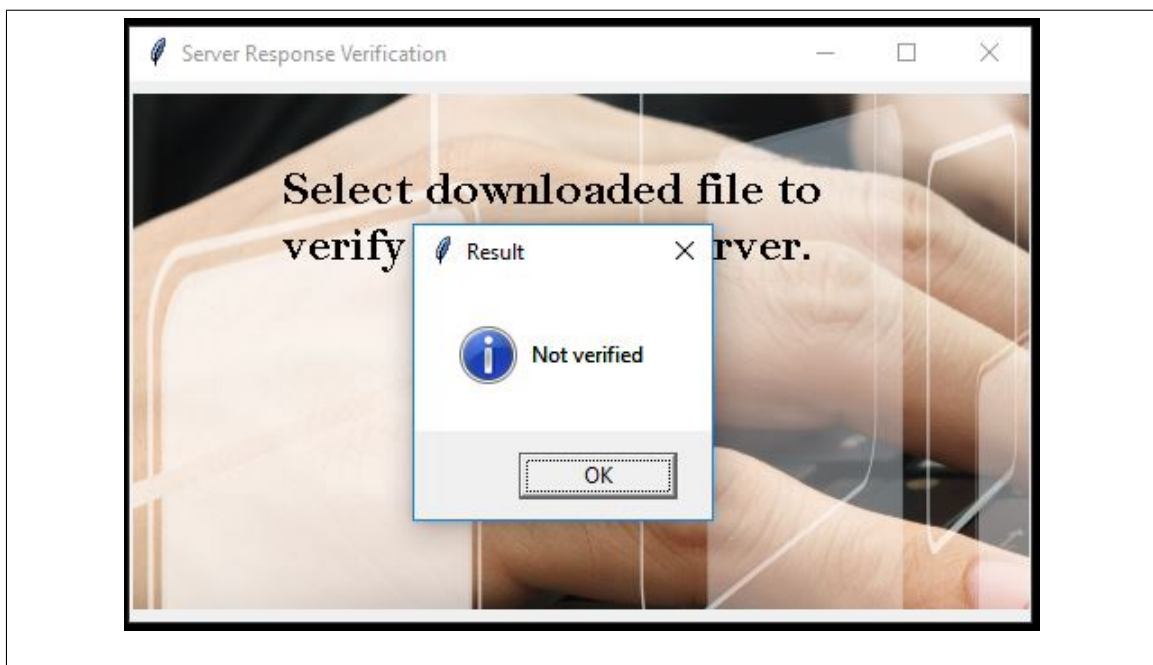


Figure 11: System response if Signature is not verified

Conclusion

In this project a central server can be accessed to determine whether some information on an individual is correct or not, but without divulging the information itself. The database will thus help in determining whether user details are correct or not, but without the database server itself volunteering such information. This is done in a secure and trusted manner by making an encrypted link between client and server through SSL thus ensuring that information is not altered during the 2-way communication between the client and server. The server replies yes/no for each query asked by client separately. The digital signature helps in authentication and integrity of the response from the server. Also access to “public-key certificate” issued by a certification authority is not an issue.