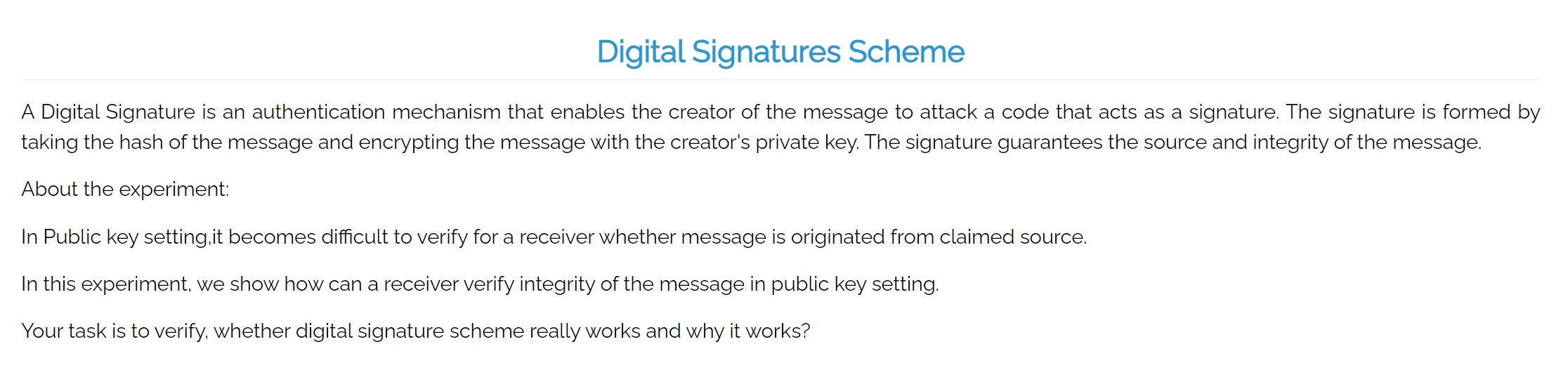
Name : Shreyas Narule

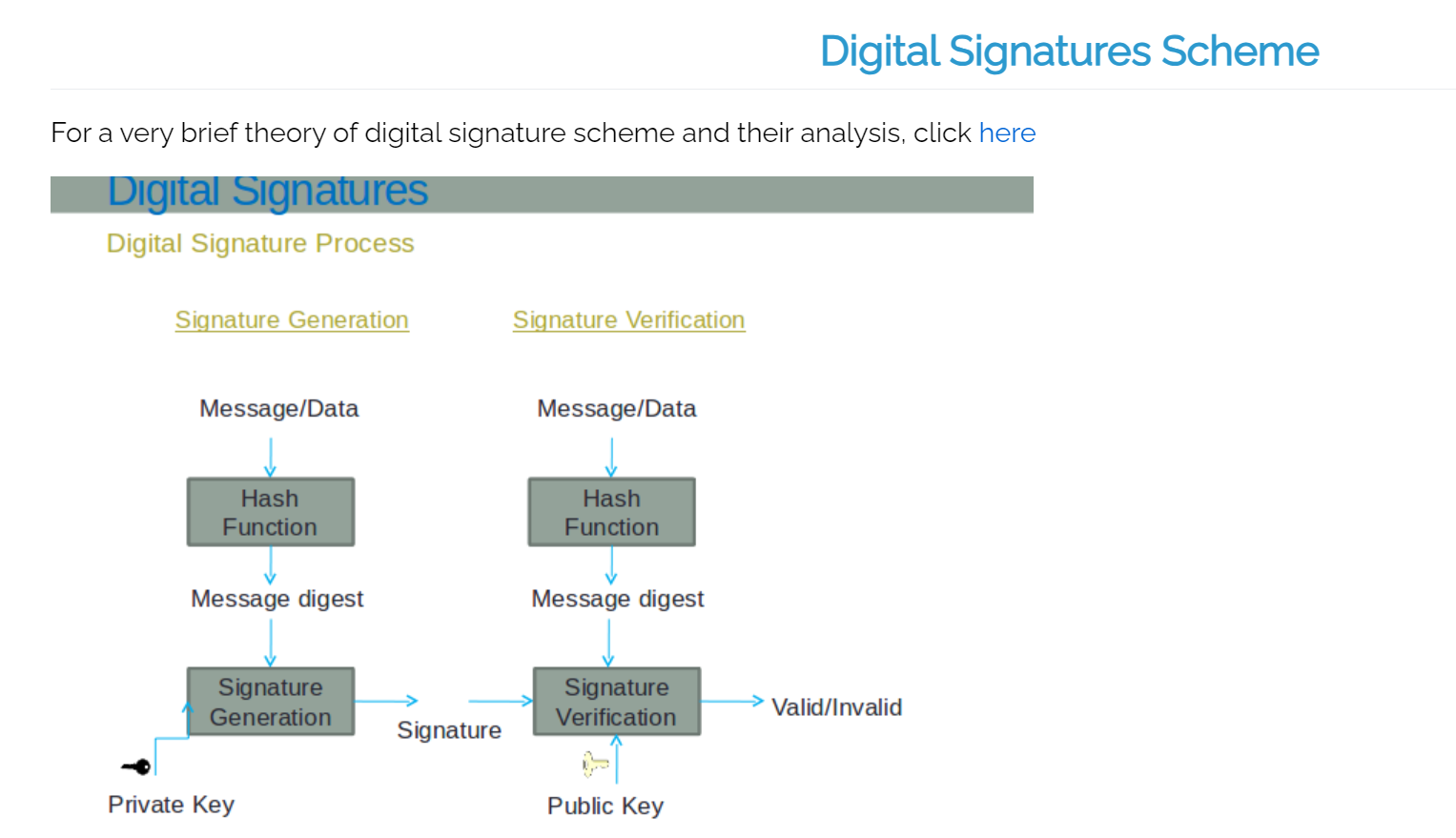
PRN: 21510077

Batch : B4

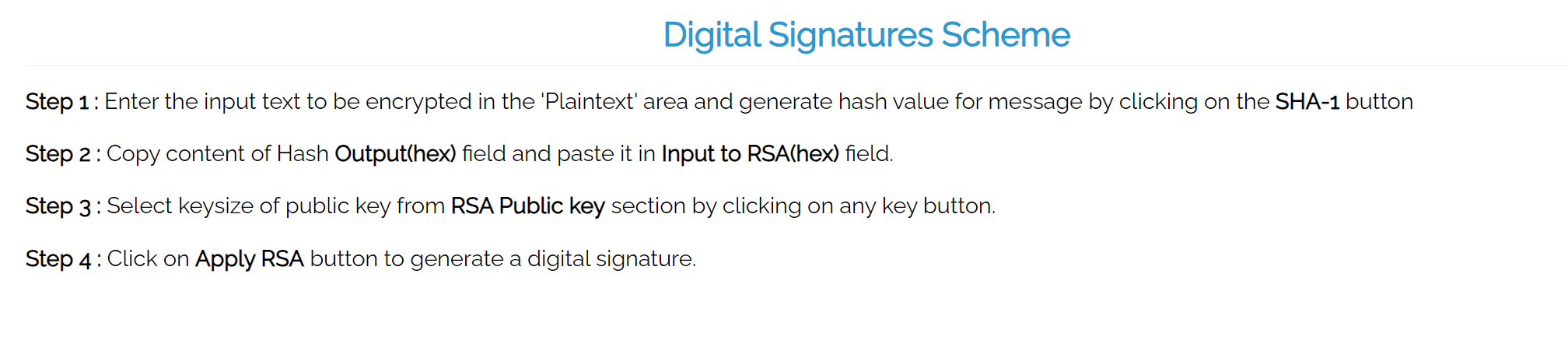
Aim :



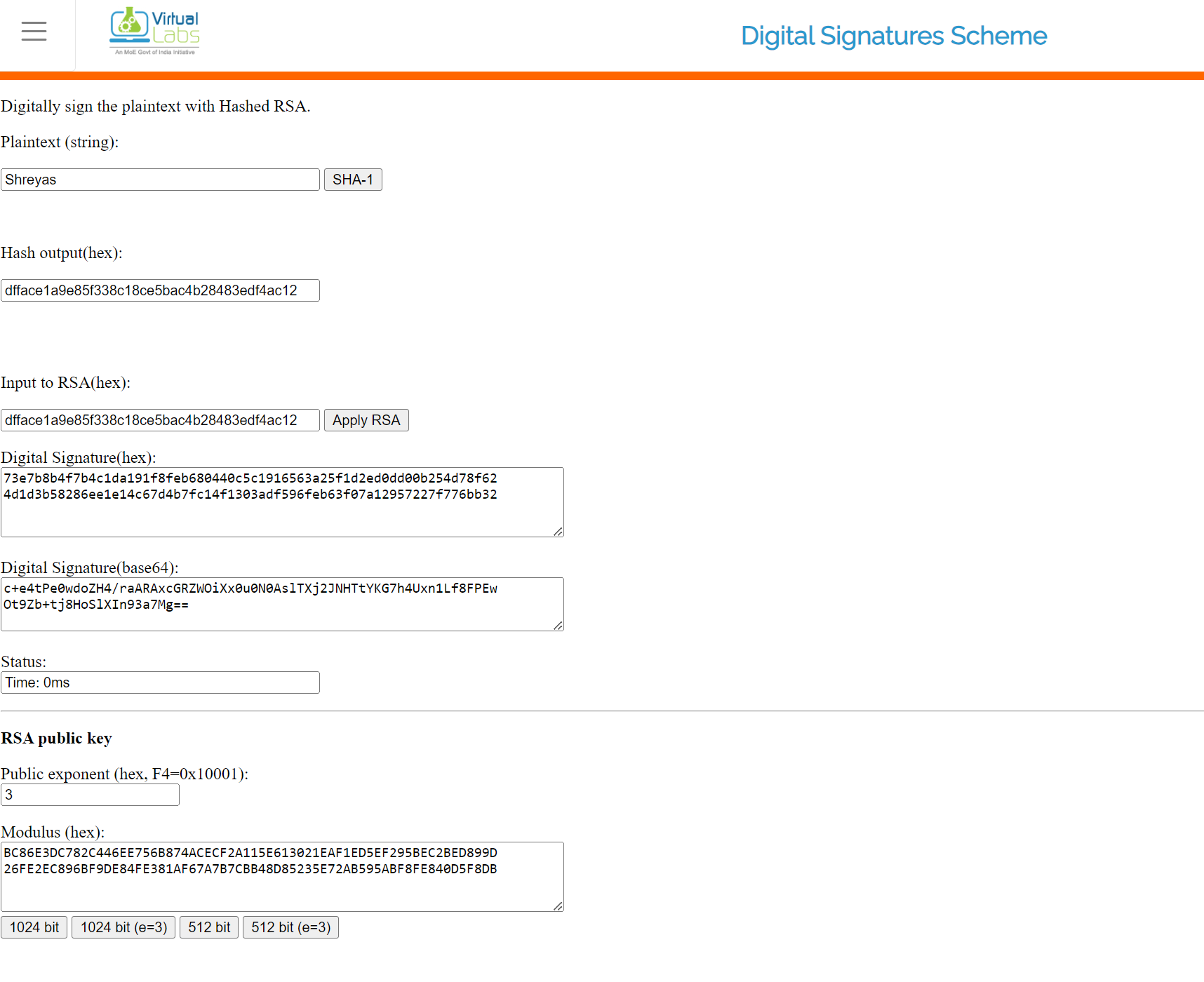
Theory :



Procedure :



Simpuliation :



Assignment :

1. Digital signature can't provide \_\_\_\_\_\_ for the message

(a) Integrity  
(b) Confidentiality  
(c) NOn repudiation  
(d) Authentication

Ans : b) Confidentiality

1. Digital signature uses \_\_\_\_\_\_ for generating valid signature  
   (a) Private key  
   (b) Public key  
   (c) Secret key  
   (d) None of the above

Ans : a) Private key

1. Verification Algorithm uses \_\_\_\_\_\_ for validating digital signature  
   (a) Private key  
   (b) Public key  
   (c) Secret key  
   (d) None of the above

Ans : b) Public key

1. Is digital signature scheme possible without public key cryptography

(a) Yes  
(b) No  
(c) May be exist  
(d) None of the above

Ans : b) No

1. Explain importance of Hashing(using experiment)and explain why Hashing is needed ?

Ans :

**Importance of Hashing**: Hashing plays a crucial role in digital signatures and cryptographic applications by ensuring the integrity of a message. A hash function takes an input (message) and produces a fixed-size string (hash), which is unique for different inputs.

**Experiment**:

1. Take a file or message and generate its hash value using a hash function (e.g., SHA-256).
2. Modify the message slightly (even changing one letter) and generate the hash again.
3. You will notice that even a tiny change in the message results in a completely different hash value. This property is called the **avalanche effect**.

**Why Hashing is Needed**:

* **Integrity**: Hashing ensures that any modification in the message can be detected. Even a small change in the input will produce a drastically different hash.
* **Efficiency**: Hashing converts a message into a short, fixed-length representation (hash), which can then be signed instead of signing the entire message, saving computational resources.
* **Uniqueness**: A good cryptographic hash function generates a unique output for different inputs, which helps in message authentication.

1. Suggest a scheme that does not use any hashing scheme

Ans :

One such scheme is the **RSA Digital Signature** scheme without hashing. In this scheme, the entire message is encrypted with the private key to generate a signature, and the public key is used for verification. However, this approach is inefficient, as it requires encrypting and decrypting the entire message, which is computationally expensive.

1. Explain why digital signature schemes works ?

Ans :

Digital signature schemes work based on **asymmetric cryptography** using a pair of keys: a **private key** (kept secret) and a **public key** (shared with everyone). Here's how it works:

1. **Signing**: The sender generates a hash of the message and then encrypts the hash with their private key. This encrypted hash forms the digital signature.
2. **Verification**: The receiver uses the sender’s public key to decrypt the signature and obtain the original hash. The receiver also generates the hash of the received message. If both hashes match, the message is authentic, unaltered, and came from the sender.

**Why it works**:

* **Authenticity**: Only the sender could have created the signature, as only they have the private key.
* **Integrity**: If the message is altered, the hashes won’t match.
* **Non-repudiation**: The sender cannot deny sending the message, as only their private key could have generated the signature.