Department of Information Technology

Academic Year: 2019-20 **Semester: V**

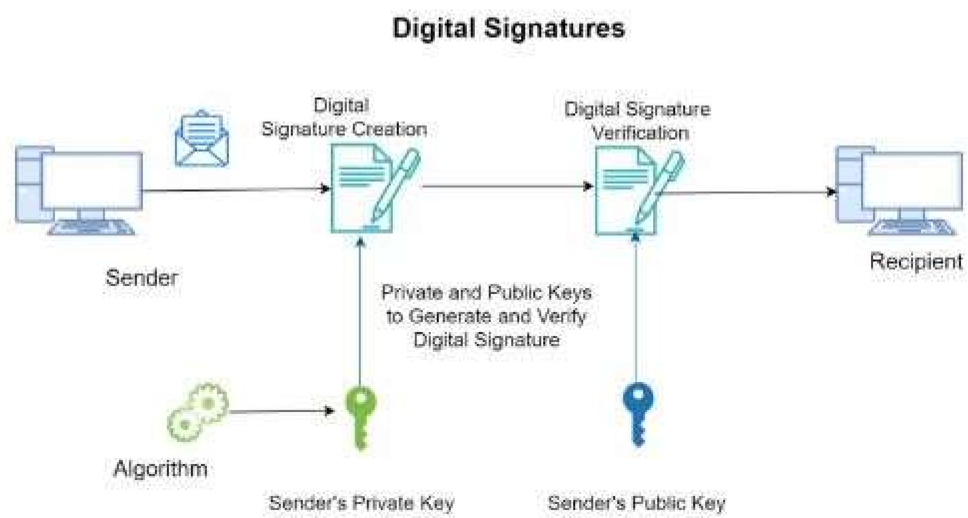
Class / Branch: TE IT

Subject: Security Lab

Subject Incharge : Prof. Apeksha Mohite

Experiment No. 12

1. **Aim: To study and analyze RSA cryptosystem and digital signature scheme.**
2. **Software Required** : CrypTool 1.4.41
3. **Theory :**

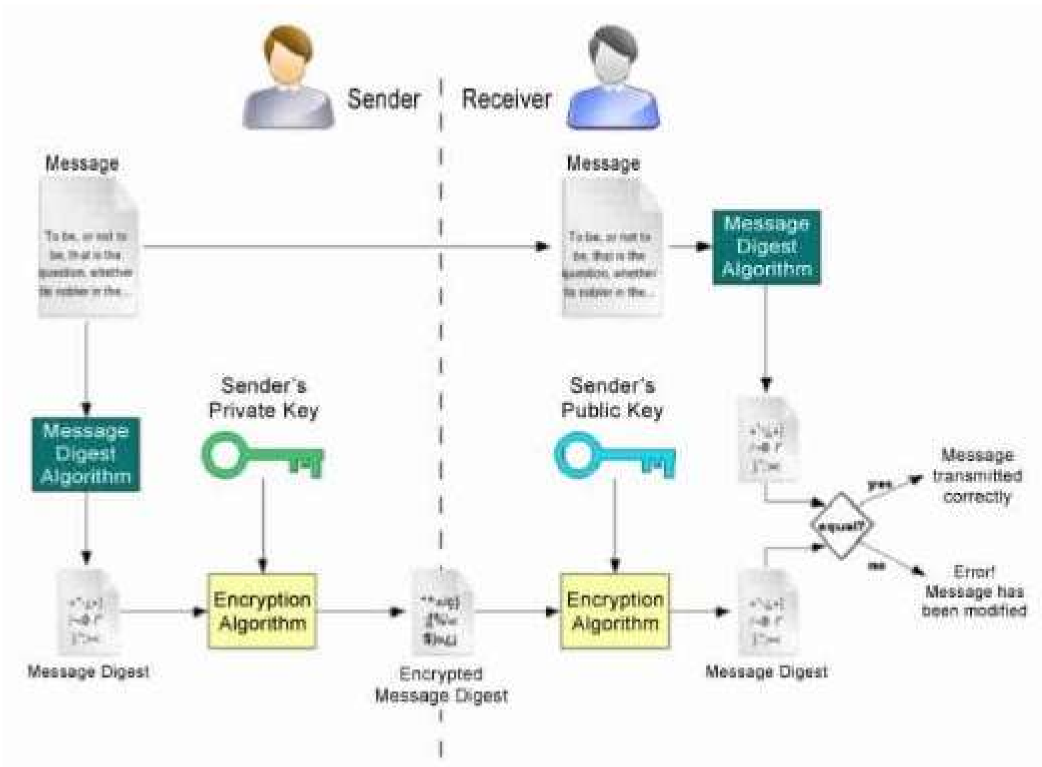


Digital signatures can provide the added assurances of evidence of origin, identity and status of an electronic document, transaction or message and can acknowledge informed consent by the signer.

How digital signatures work

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.

Digital signatures work because public key cryptography depends on two mutually authenticating cryptographic keys. The individual who is creating the digital signature uses their own private key to encrypt signature-related data; the only way to decrypt that data is with the signer's public key. This is how digital signatures are authenticated.



How to create a digital signature

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 as hashing is much faster than signing.

The value of a hash is unique to the hashed data. Any change in the data, even a change in a single character, will result 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 -- authentication.

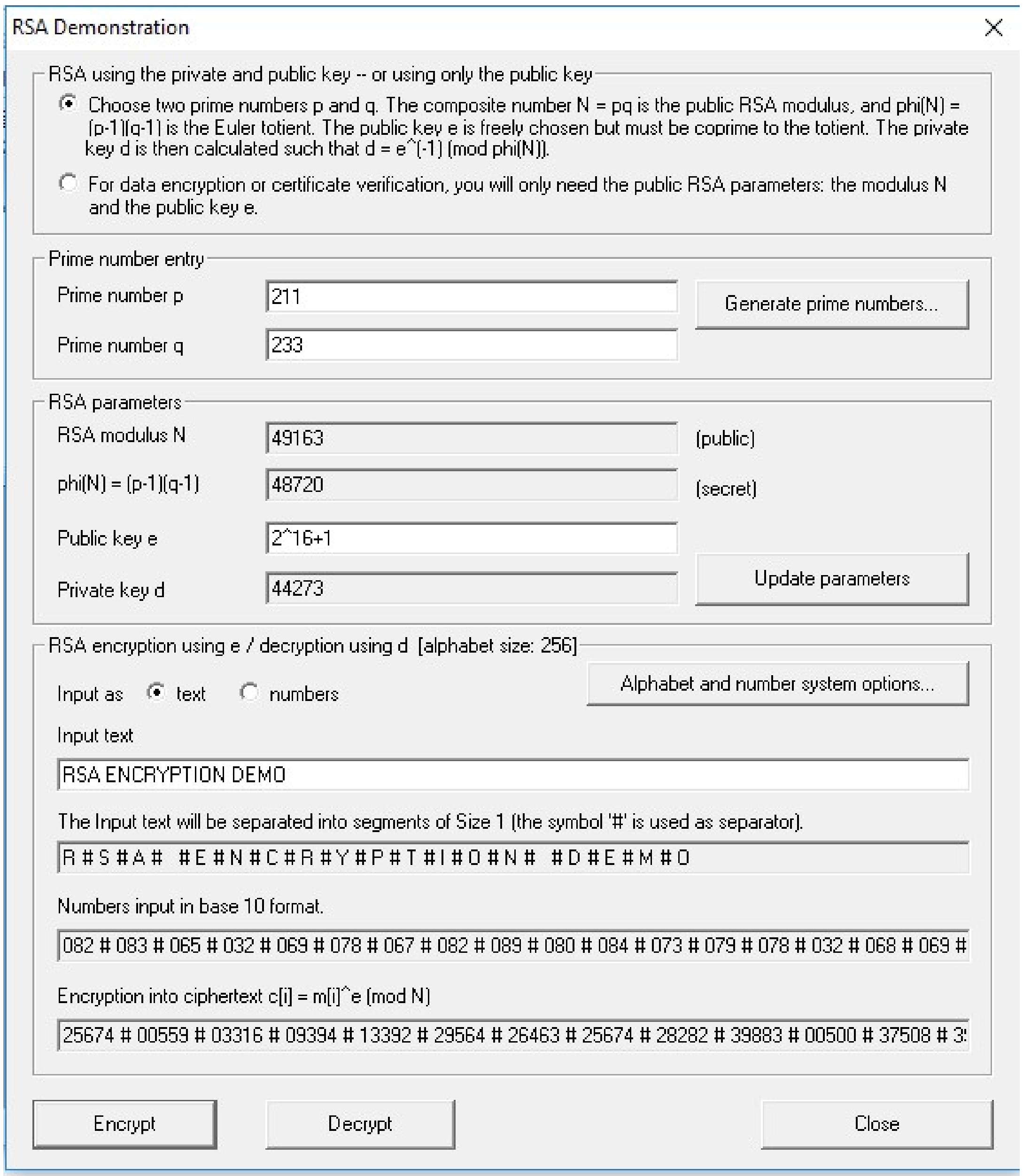
RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private.
An example of asymmetric cryptography :

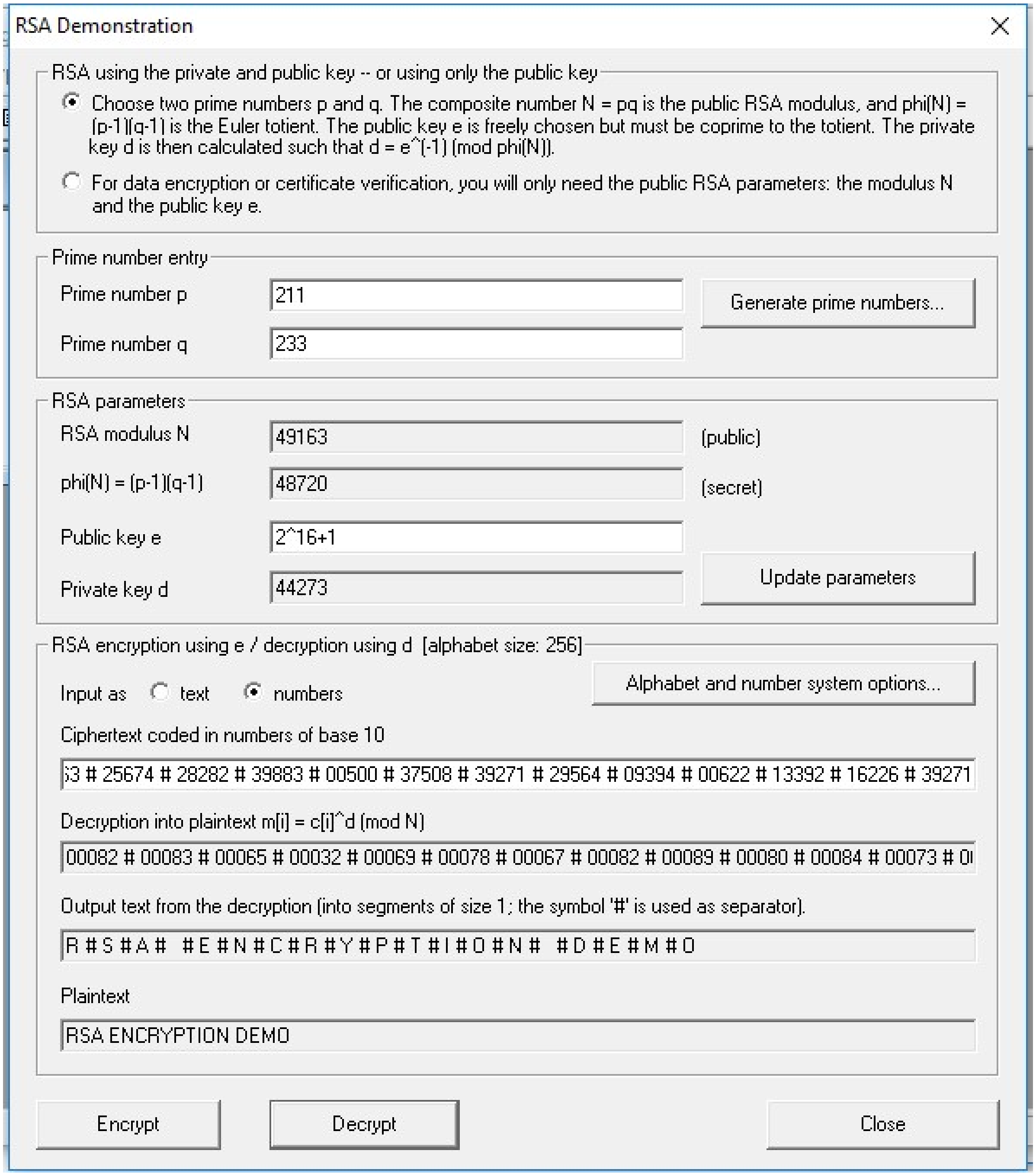
1. A client (for example browser) sends its public key to the server and requests for some data.
2. The server encrypts the data using client's public key and sends the encrypted data.
3. Client receives this data and decrypts it.

Since this is asymmetric, nobody else except browser can decrypt the data even if a third party has public key of browser.
The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe

that 1024 bit keys could be broken in the near future. But till now it seems to be an infeasible task.

We demonstrate RSA with the help of cryptool




RSA Demonstration
RSA using the private and public key -- or using only the public key
Choose two prime numbers p and q. The composite number N = pq is the public RSA modulus, and phi(N) = (p-1)(q-1) is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that d = e^(-1) (mod phi(N)).
For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.
Prime number entry
Prime number p
211
Generate prime numbers...
Prime number q
233
RSA parameters
RSA modulus N
49163
(public)
phi(N) = (p-1)(q-1)
48720
(secret)
Public key e
2^16+1
Private key d
44273
Update parameters
RSA encryption using e / decryption using d [alphabet size: 256]
Alphabet and number system options...
Input as text numbers
Ciphertext coded in numbers of base 10
53 # 25674 # 28282 # 39883 # 00500 # 37508 # 39271 # 29564 # 09394 # 00622 # 13392 # 16226 # 39271
Decryption into plaintext m[i] = c[i]^d (mod N)
00082 # 00083 # 00065 # 00032 # 00069 # 00078 # 00067 # 00082 # 00089 # 00080 # 00084 # 00073 # 0
Output text from the decryption (into segments of size 1; the symbol '#' is used as separator).
R # S # A #   # E # N # C # R # Y # P # T # I # O # N #   # D # E # M # O
Plaintext
RSA ENCRYPTION DEMO
Encrypt
Decrypt
Close

A digital signature scheme typically consists of 3 algorithms;

- A *key generation* algorithm that selects a *private key* uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding *public key*.
- A *signing* algorithm that, given a message and a private key, produces a signature.
- A *signature verifying* algorithm that, given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

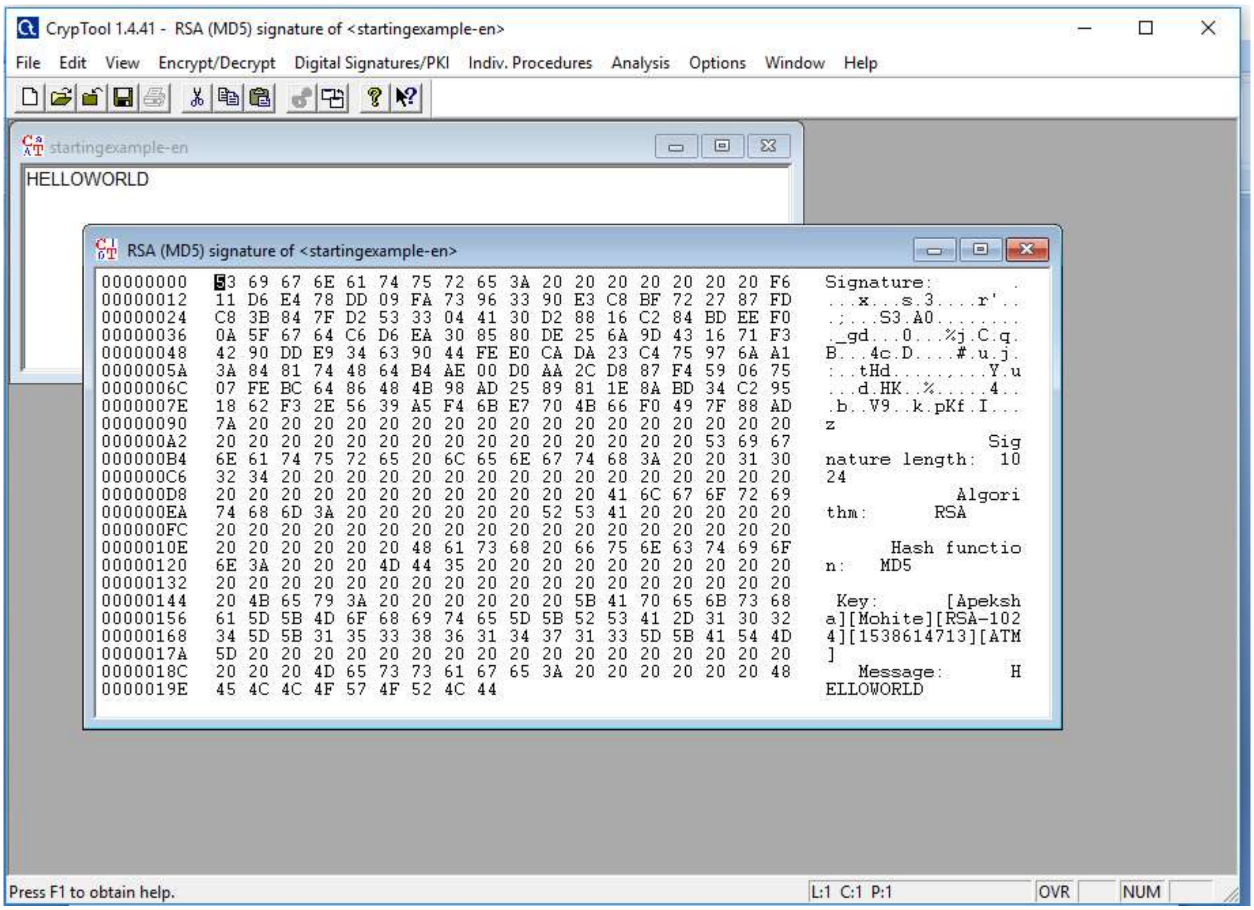
Two main properties are required. First, the authenticity of a signature generated from a fixed message and fixed private key can be verified by using the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature for a party without knowing that party's private key. A digital signature is an authentication mechanism that enables the creator of the message to attach a code that acts as a signature.

One digital signature scheme (of many) is based on RSA. To create signature keys, generate a RSA key pair containing a modulus, N, that is the product of two random secret distinct large primes, along with integers, e and d, such that $e\ d \equiv 1 \pmod{\varphi(N)}$, where φ is the Euler phi-function. The signer's public key consists of N and e, and the signer's secret key contains d.

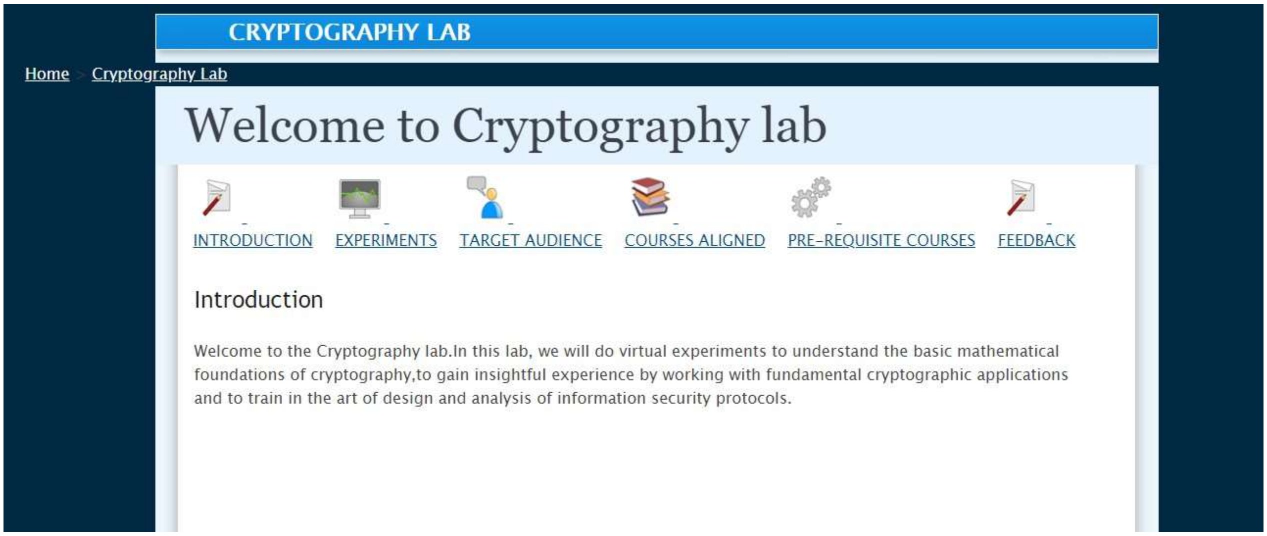
To sign a message, m, the signer computes a signature, σ, such that $\sigma \equiv md \pmod{N}$. To verify, the receiver checks that $\sigma e \equiv m \pmod{N}$.

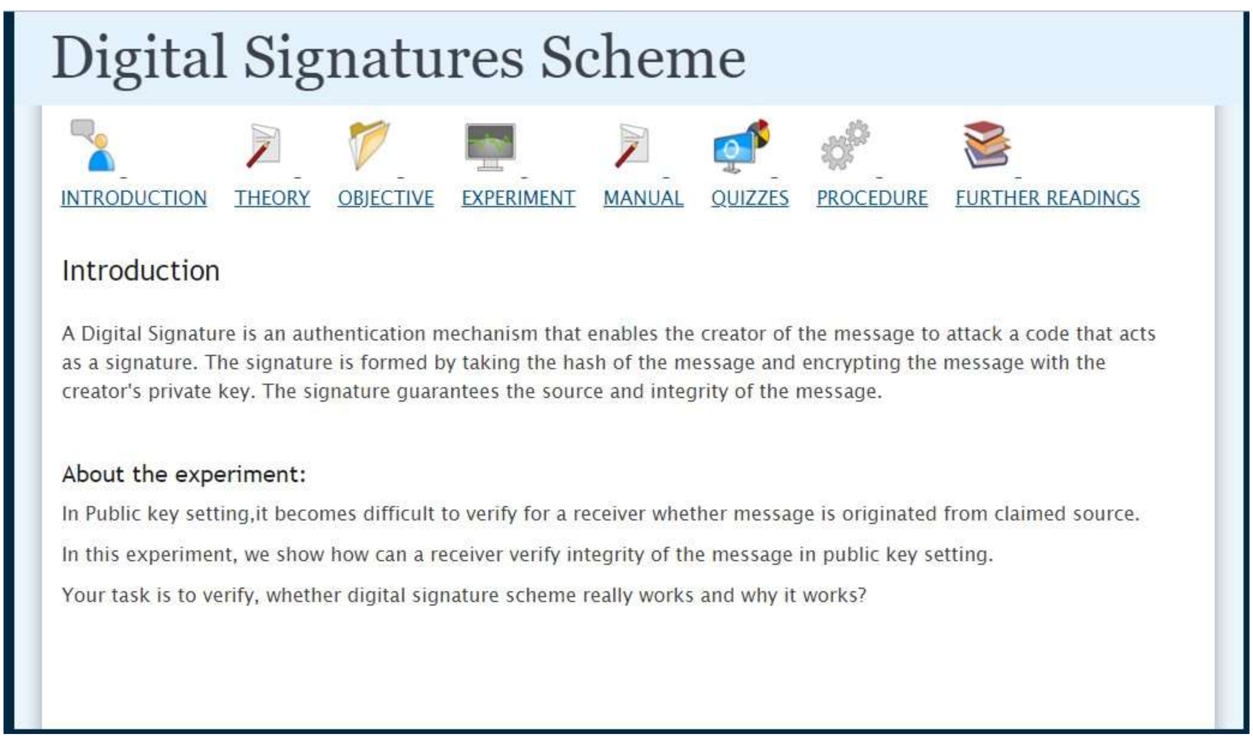
Digital Signatures using Cryptool

MD5 digital signature example



IIT VIRTUAL LAB FOR CRYPTOGRAPHY





Digital Signatures Scheme

INTRODUCTION THEORY OBJECTIVE EXPERIMENT MANUAL QUIZZES PROCEDURE FURTHER READINGS

Manual

Step 1: Enter the input text to be encrypted in the 'Plaintext' area and generate hash value for message by clicking on the **SHA-1** button

Step 2: Copy content of **Hash Output(hex)** field and paste it in **Input to RSA(hex)** field.

Step 3: Select keysize of public key from **RSA Public key** section by clicking on any key button.

Step 4: Click on **Apply RSA** button to generate a digital signature.