

PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



## Department of Information Technology

Academic Year: 2019-20

Semester: V

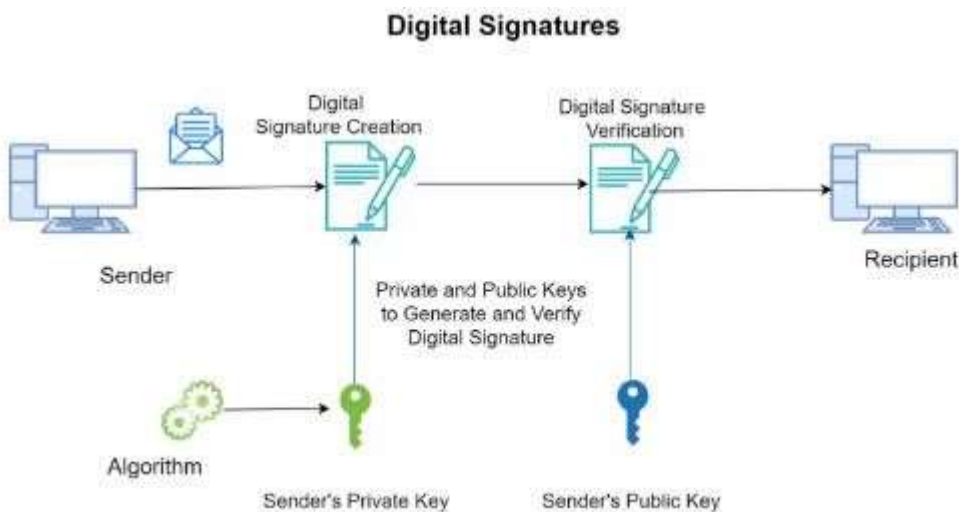
Class / Branch: TE IT

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 :**



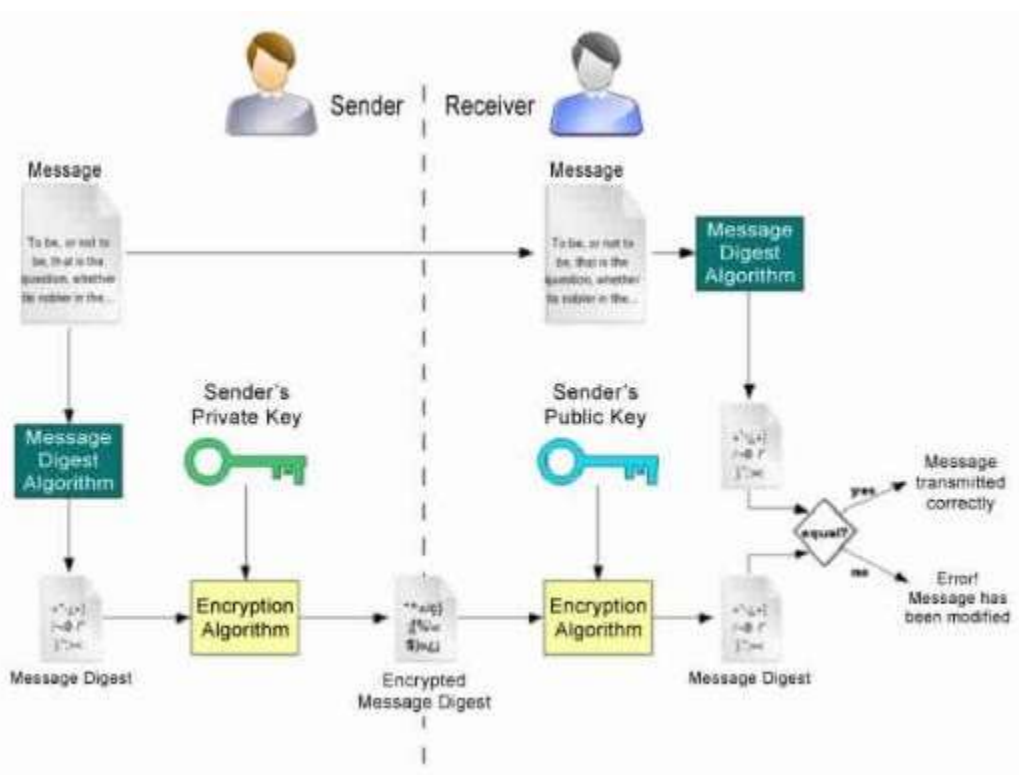
Digital signatures can provide the added assurances of evidence of origin, identity and status of an electronic document, transaction or message and can acknowledge informed consent by the signer.

#### 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.



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



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



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



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** [X]

☐ 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  $d = e^{-1} \pmod{\phi(N)}$ .

☐ 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:

Prime number q:

**RSA parameters**

RSA modulus N:  (public)

$\phi(N) = (p-1)(q-1)$ :  (secret)

Public key e:

Private key d:

**RSA encryption using e / decryption using d [alphabet size: 256]**

Input as: ☒ text ☐ numbers

Input text:

The Input text will be separated into segments of Size 1 (the symbol '#' is used as separator).

Numbers input in base 10 format.

Encryption into ciphertext  $c[i] = m[i]^e \pmod{N}$



PARSHVANATH CHARITABLE TRUST'S

# A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



### 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  $d = e^{-1} \pmod{\phi(N)}$ .

☐ 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  $p$

Prime number  $q$

RSA parameters

RSA modulus  $N$   (public)

$\phi(N) = (p-1)(q-1)$   (secret)

Public key  $e$

Private key  $d$

RSA encryption using  $e$  / decryption using  $d$  [alphabet size: 256]

Input as ☐ text ☒ numbers

Ciphertext coded in numbers of base 10

Decryption into plaintext  $m[i] = c[i]^d \pmod{N}$

Output text from the decryption (into segments of size 1; the symbol '#' is used as separator).

Plaintext



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



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  $e d \equiv 1 \pmod{\phi(N)}$ , where  $\phi$  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,  $\sigma$ , such that  $\sigma \equiv m d \pmod{N}$ . To verify, the receiver checks that  $\sigma e \equiv m \pmod{N}$ .





PARSHVANATH CHARITABLE TRUST'S

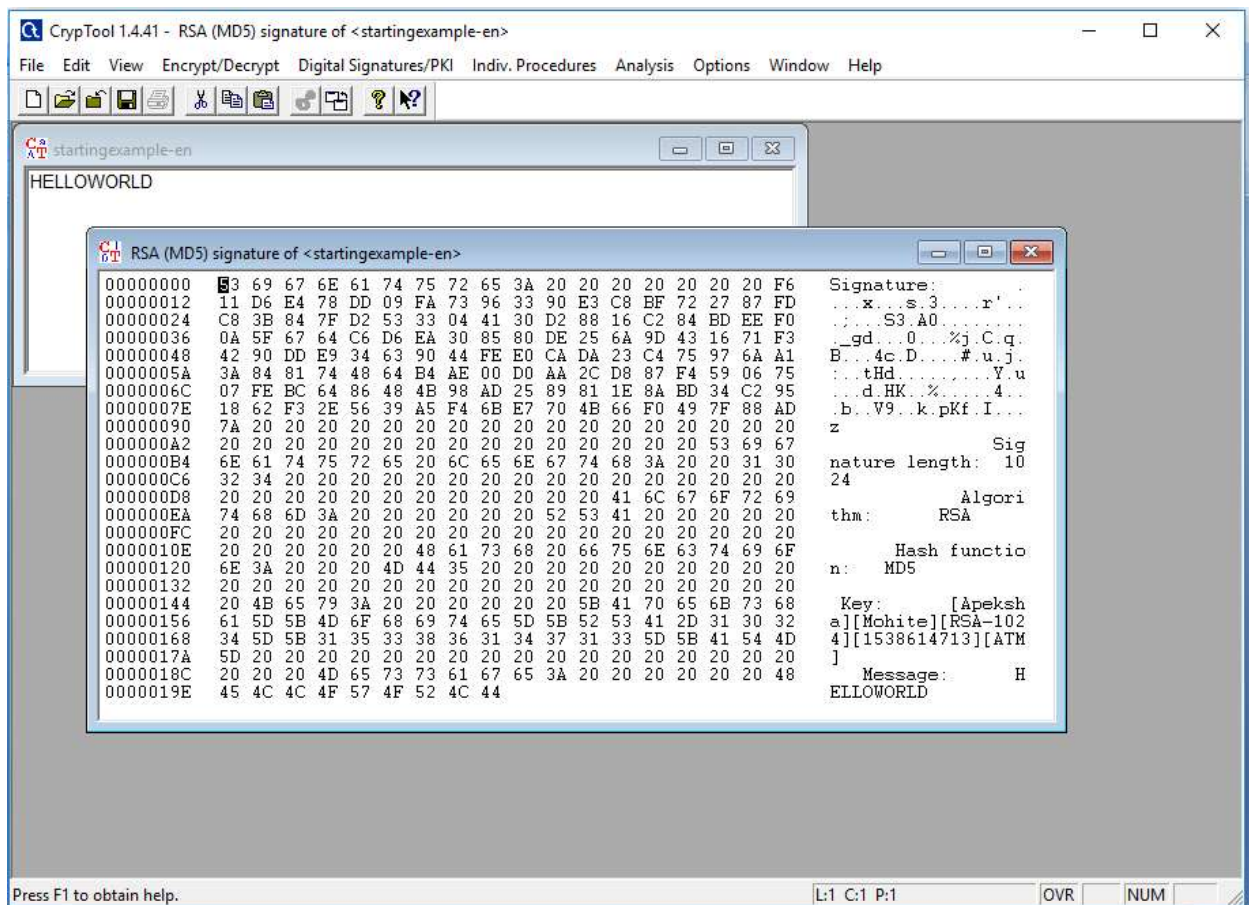
**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



## Digital Signatures using Cryptool

### MD5 digital signature example





PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

(All Branches NBA Accredited)



## IIT VIRTUAL LAB FOR CRYPTOGRAPHY

**CRYPTOGRAPHY LAB**

[Home](#) [Cryptography Lab](#)









# Welcome to Cryptography lab

[INTRODUCTION](#) [EXPERIMENTS](#) [TARGET AUDIENCE](#) [COURSES ALIGNED](#) [PRE-REQUISITE COURSES](#) [FEEDBACK](#)

### Introduction

Welcome to the Cryptography lab. In this lab, we will do virtual experiments to understand the basic mathematical foundations of cryptography, to gain insightful experience by working with fundamental cryptographic applications and to train in the art of design and analysis of information security protocols.

# Digital Signatures Scheme

[INTRODUCTION](#) [THEORY](#) [OBJECTIVE](#) [EXPERIMENT](#) [MANUAL](#) [QUIZZES](#) [PROCEDURE](#) [FURTHER READINGS](#)

### Introduction

A Digital Signature is an authentication mechanism that enables the creator of the message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

### About the experiment:

In Public key setting, it becomes difficult to verify for a receiver whether message is originated from claimed source. In this experiment, we show how can a receiver verify integrity of the message in public key setting. Your task is to verify, whether digital signature scheme really works and why it works?





PARSHVANATH CHARITABLE TRUST'S

# A. P. SHAH INSTITUTE OF TECHNOLOGY

(All Branches NBA Accredited)



## Digital Signatures Scheme



### 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.

## Digital Signatures Scheme



### Experiment

Digitally sign the plaintext with Hashed RSA.

Plaintext (string):

first digital certificate

Hash output(hex):

49b2952581d7cc58a42e65e92bcb58b74092f4f6

Input to RSA(hex):

49b2952581d7cc58a42e65e92bcb58b74092f4f6

Digital Signature(hex):

8ad81803e31cc30f64858d5673c123b6c41035ecd3789c4a2e9ff007caf777  
3e590948818247bf9250019b44a49789dd114b37d56e37ed1a778592614a12  
478a6d1815a752884e93d538561420801cac97018ed52b727bba7e13233d99d  
132577b784e555b100bfbc7a6d186505636e00ba00ee412ed56049b71cf69

Digital Signature(base64):

mtgYA+MwvU/hW1WcBjEpxOQNez9N4nPSi9B/AK9/c+WQlqYjIhV5JQgZEpJdp  
3IEUs31W437Rp3hVkmFKEkKbRgVp1KTPVP2WBQggByslwFu1Sye7p+EyM9md  
EyV34Tl8V9OCu8em32tQVjbgAlaA7kEu1YBjuz2k=

Status:

Time: 16ms

### RSA public key

Public exponent (hex, F4=0x10001):

f0601

Modulus (hex):

45261939975940bb7a58dffa5f54e6504989175f5a09288810b0975871e99  
af3b5de94057b0fd07635f5f97444504fa351694461d0d30cf0192e307727c08  
5168c786771c561a9400fb49175e9e6aa4e23e11af69e9412da23b0cb6684c4  
c2429bce139e848ab26d0629073351f4acd36074eafcd036a5eb3359d2a699dc3

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3)