**Experiment No. 11**

**Aim:** To perform Digital Signature Scheme Experiment using Virtual lab.

**Theory**:

**What Are Digital Signatures?**

The objective of digital signatures is to authenticate and verify documents and data. This is necessary to avoid tampering and digital modification or forgery during the transmission of official documents.

With one exception, they work on the public key cryptography architecture. Typically, an asymmetric key system encrypts using a public key and decrypts with a private key. For digital signatures, however, the reverse is true. The signature is encrypted using the private key and decrypted with the public key. Because the keys are linked, decoding it with the public key verifies that the proper private key was used to sign the document, thereby verifying the signature's provenance.

**RSA Signatures**

The RSA public-key cryptosystem provides a digital signature scheme (sign + verify), based on the math of the modular exponentiations and discrete logarithms and the computational difficulty of the RSA problem (and its related integer factorization problem).

**Key Generation**

The RSA algorithm uses **keys** of size 1024, 2048, 4096, ..., 16384 bits. RSA supports also longer keys (e.g. 65536 bits), but the performance is too slow for practical use (some operations may take several minutes or even hours). For 128-bit security level, a 3072-bit key is required.

The **RSA key-pair** consists of:

* public key {***n***, ***e***}
* private key {***n***, ***d***}

The numbers ***n*** and ***d*** are typically big integers (e.g. 3072 bits), while ***e*** is small, typically 65537.

By definition, the RSA key-pairs has the following property:

for all ***m*** in the range [0...***n***)

**RSA Sign**

**Signing** a message ***msg*** with the private key exponent ***d***:

1. Calculate the message hash:
2. Encrypt ***h*** to calculate the signature:

The hash ***h*** should be in the range [0...***n***). The obtained **signature** ***s*** is an integer in the range [0...***n***).

**RSA Verify Signature**

**Verifying** a signature ***s*** for the message ***msg*** with the public key exponent ***e***:

Calculate the message hash:

Decrypt the signature:

Compare ***h*** with ***h'*** to find whether the signature is valid or not

If the signature is correct, then the following will be true:

**Vlab output**:



