Implementation of RSA in python and Simulation of AES, DES, Triple DES using Crypttool

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.

RSA implemented in python is demonstrated below

```
def decrypt(pk, ciphertext):
    #Unpack the key into its components
    key, n = pk
    #Generate the plaintext based on the ciphertext and key usi
    plain = [chr((char ** key) % n) for char in ciphertext]
    #Return the array of bytes as a string
    return ''.join(plain)


if __name__ == '__main__':
    '''
    Detect if the script is being run directly by the user
    '''
    print "RSA Encrypter/ Decrypter"
    p = int(raw_input("Enter a prime number (17, 19, 23, etc):
    q = int(raw_input("Enter another prime number (Not one you
    print "Generating your public/private keypairs now . . ."
    public, private = generate_keypair(p, q)
    print "Your public key is ", public ," and your private key
    #message = raw_input("Enter a message to encrypt with your
    f = open("sample text.txt", "r")
    message = f.read()
```

```
Python 2.7.18 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.18 (v2.7.18:8d21aa21f2, Apr 20 2020, 13:19:08) [MSC v.1500 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: E:\avrutti\AR\ecc_python_project\RSA\rsa_27.py ==========
RSA Encrypter/ Decrypter
Enter a prime number (17, 19, 23, etc): 23
Enter another prime number (Not one you entered above): 11
Generating your public/private keypairs now . . .
Your public key is  (107, 253)  and your private key is  (183, 253)
Your encrypted message is:
153942291313913921614214796929620122914324 6
Encryption Time  7
Decrypting message with public key  (107, 253)  . . .
Your message is:
Avrutti Research
Decryption Time  3
>>>
```

Encryption of a file with the AES cipher using the content of the component "Key" as key and the parameters of the AES component for key size, block mode and padding. For 128 bit AES you should enter 32 hex characters as key. Non-hex characters are extracted by the "StringDecoder" component

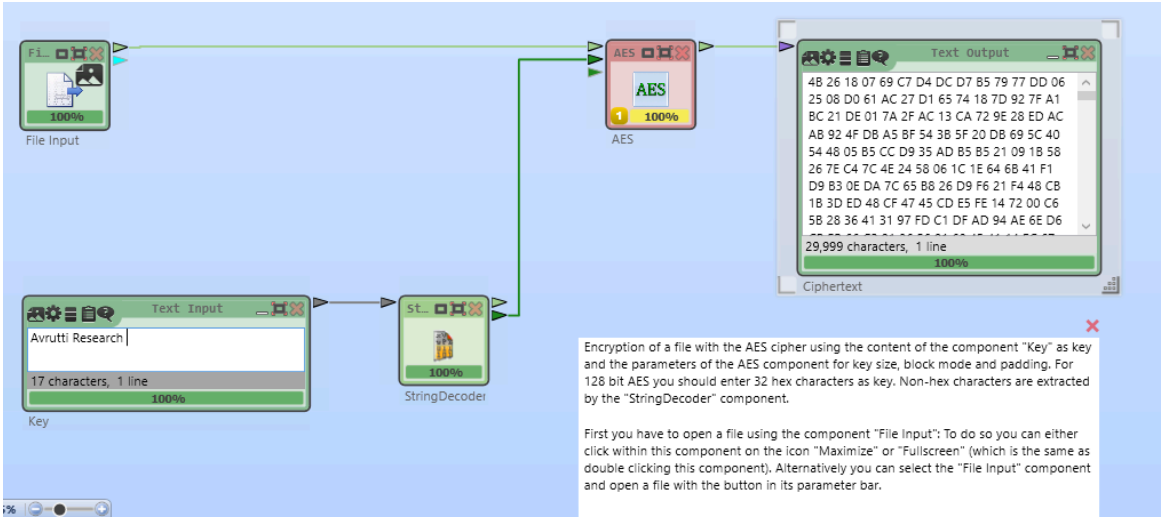

Fig 1. AES implementation using Crypttool

Data encryption standard uses cryptographic algorithm that can be used to protect electronic data. DES algorithm makes use of symmetric cryptographic method. Block cipher algorithm is used for encryption and decryption purpose and the message is divided into blocks of bits. DES processes the input data (Original message) of block size 64- bits and a secret key of 64-bits to provide a 64-bit cipher text.

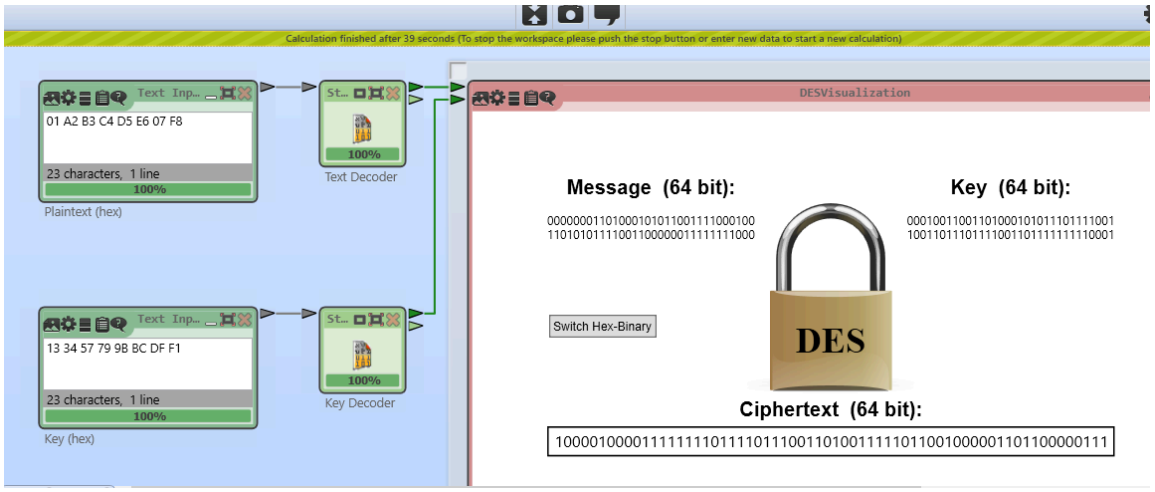

Fig 2. DES visualization using Crypttool

Triple DES is the upgrade of traditional DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. The procedure for encryption is exactly the same as regular DES, but it is repeated three times, hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

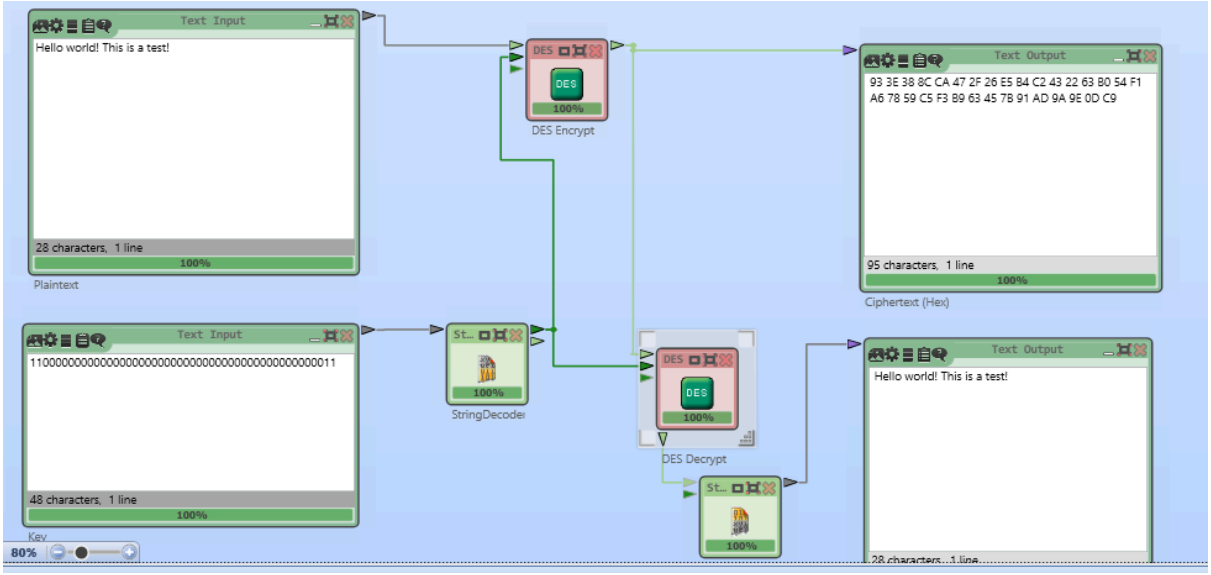

Fig 3. Triple DES simulation Crypttool

Message Digest Algorithm 5 (MD5) is a cryptographic hash algorithm that can be used to create a 128-bit string value from an arbitrary length string. MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function that results in a 128-bit hash value. The 128-bit (16-byte) MD5 message digest hashes typically are represented as 32-digit hexadecimal numbers (for example, ad57d3e696d289f2afd663725127abdc)

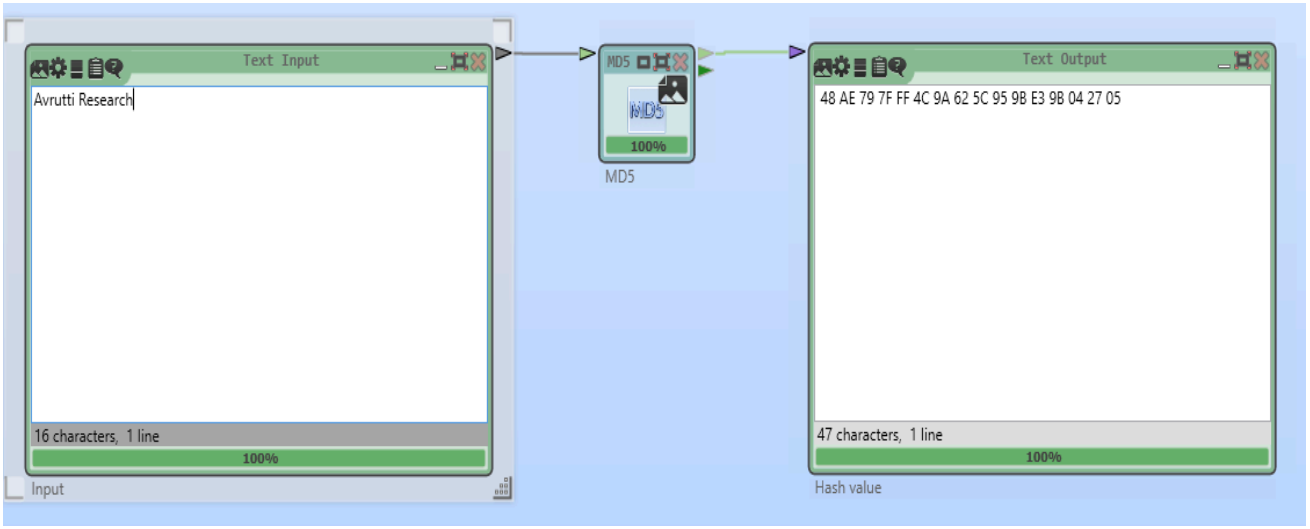

Fig 4. MD5 simulation in Cryptool