Implementation of RSA-KEM and Exploration of latest Advancements

MAJOR PROJECT-II REPORT

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:

MANISH KUMAR
2K19/ISY/10

Under the supervision of
DR. SEBA SUSAN
PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042
June, 2021

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Manish Kumar, Roll No. 2K19/ISY/10 student of M.Tech, Information Systems, hereby declare that the Major Project-II titled " **Implementation of RSA-KEM and Exploration of latest Advancements** " which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Manish 27/06/21

Place: Delhi MANISH KUMAR

Date: June , 2021

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Major Project-II titled " **Implementation of RSA-KEM and Exploration of latest Advancements** " which is submitted by Manish Kumar, Roll No. 2K19/ISY/10 Information Technology, Delhi Technological University, Delhi in fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted int part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **DR. SEBA SUSAN**
Date: June, 2021 **SUPERVISOR**

EPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042
June, 2021

ABSTRACT

Cryptography is used to protect information. Encryption and decryption can confirm the confidentiality, integrity of information and protect information from tampering, forgery and counterfeiting. RSA (Rivest, Shamir, Adleman) uses two keys: - private key and public key. RSA-KEM (Key Encapsulation Mechanism) is a hybrid encryption algorithm that uses RSA trapdoor permutation along with a key derivation function. RSA contains three functions:- Key generation, Encryption and Decryption. In RSA-KEM, password based key derivation function is used for key stretching. Password, salt, iteration is used for generation of derived key. Post-quantum cryptography is cryptography under the assumption that the attacker has a large quantum computer and cryptosystems aim to remain secure even in this scenario. This paper proposed an implementation of a RSA-KEM encrypt/decrypt based on the study of RSA public key algorithm and KEM. Shor's Algorithm is used for integer factorization which is polynomial time for quantum computer. This can be threat for RSA security. In this paper matlab implementation of Shor's algorithm is presented. This paper also discusses popular methods for making qubits like Silicon based Qubits in which electron is put inside nano material which is used as a transistor. In Superconducting circuit method insulator is used as a sandwich in between two metal layers. Used by Google, IBM, Intel, Microsoft. In Flux qubits method very small size loop of superconducting metal is used. This paper also discusses Quantum Proof Algorithm like Lattice-based cryptography used concept of good and bad base. In Learning with

errors method if we have more equation then variable, It is Over defined system. In Code based cryptography Some matrix's allow for efficient error correction (good matrix) but most matrix's does not (bad matrix) concept is used. In Hash based signatures scheme have long signatures or keys, but they are secure. Also discuss Multivariate Quantum proof algorithm.

Keywords: RSA, Encryption, Decryption, RSA-KEM, qubits, quantum computer, Shor's algorithm, quantum proof algorithms.

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042

JUNE, 2020

ACKNOWLEDGEMENT

I am very thankful to **Dr. Seba Susan** (Professor, Department of Information Technology) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Manish 27/06/21

MANISH KUMAR
2K19/ISY/10

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS

RSA	Rivest, Shamir, Adleman
KEM	Key Encapsulation Mechanism
PKC	Public key cryptography
AES	Advanced Encryption Standard
PBKDF	Password based key derivation function
XOR	Exclusive disjunction
DES	Data Encryption Standard
CPU	Central processing unit
GPU	Graphics processing unit

Exploration and implementation of RSA-KEM algorithm

1.0 Introduction

The principles and methods of transforming plaintext into ciphertext, and then converting ciphertext into plaintext is cryptography, and method and technique of converting ciphertext into plaintext without using key is known as cryptanalysis. In symmetric key, only one key is used for encryption and the same key is again used for decryption. Both parties have the same key. In public key cryptography, one of the keys is available in public domain, and other related key is kept secret [11]. It is not possible to derive one key with use of other key and algorithm. In encryption we use an algorithm and one of the keys. Using different key, we get different ciphertext using the same algorithm. After the ciphertext is sent to the receiver, using the other key and algorithm, the original plaintext is derived. Techniques like RSA PKCS#1[10] use deterministic padding scheme. When server confirm the padding and in some way leaks the outcome, it may be threat for system. In RSA-OAEP [9], keys are less efficient as compared to other schemes [12]. It can leak information through timing attack. In RSA-KEM [8] the keys are two large randomly generated prime numbers. It's hard to factor a large prime number which is a product of two prime numbers. RSA cryptosystem is universally accepted. RSA based system has two keys one private and other one is public. One key (public) is available for everyone and another key (private) is with only one person. If 'X' has to send some data to 'Y', 'X' does this by using public key of 'Y'. After receiving data from 'X', 'Y' uses its private key to get the data. RSA-KEM is an accepted key encapsulation mechanism. It uses RSA trapdoor permutation along with a key derivation function (KDF). The security depends on algorithm used and key length. In Public key cryptography two keys are required. Public key is available in public domain and can be used by everyone for sending messages and for signature verification. Private key is known by only one person and is important for decrypting messages and digital signature. Some of the attacks and security services are given in Table 1. In classical computer two bit can represent any one of four 00, 01,10,11, but any one can use i.e. two bits information. But in quantum mechanics it is possible to make superposition of each one of these four states. To find out state of two spin system four coefficients or numbers required, but in classical for two bits only two numbers. Two qubits hold four bits of information. For three qubits systems having eight different states. For 'n' qubits system is analogous to 2^n classical bits. Qubits exits any of the combination of states, but when we measured, it fall any one of the basis states. Quantum teleportation, quantum entanglement and other makes it possible to break present cryptosystem.

Cryptanalytic attacks	**Description**
Ciphertext only	Same ciphertext is available for decryption.
Known plaintext	Only copy of ciphertext and its plaintext is available to the cryptanalyst for decryption.
Chosen plaintext	Only use encryption algorithm/machine and can be used for many plaintext and corresponding ciphertext for try to find key
Chosen ciphertext	Only use decryption algorithm/machine and can be used for many symbols or string of plaintext to find key.

Table 1. Different Types of Cryptanalytic attacks

2.0 RSA (Rivest, Shamir, Adleman) Algorithm

In early days all cryptographic systems used permutation and substitution, and after that rotor encryption and decryption machine was used. Public key cryptography is the latest development which uses mathematical functions. Security for cryptography [6] system is based on key length and the effort involved in cracking the cipher. Now a days due to computational requirement we use PKC in key management, digital signature. Some of the PKC algorithms are given in Table 2.

PKC Algorithm	Encryption/Decryption	Digital signature	Key exchange
RSA	Used	Used	Used
Elliptic curve	Not used	Used	Used
Diffie-Hellman	Not used	Not used	Used
DSS	Not used	Used	Not used

Table 2. Different Algorithms for Public Key Cryptography

Asymmetric algorithm works on two keys, encryption by one key and decryption by other connected key. It's unworkable to find the decryption key when only encryption key and cryptography algorithm is known. RSA algorithm [2] is accepted as a general purpose technique for public key cryptography [1],[4],[16],[17]. RSA processing of multiple blocks is shown in Fig 1.

2.1 Algorithm

Key generation

In RSA, each party who wishes to communicate using encryption required to make a pair of keys:- one is public key and other one is private key. We select two distinct primes p and q picking randomly and close in magnitude but varying in length by few numbers, that makes factoring unbreakable. Using primality test we find prime numbers p and q and it is kept secret. Public key is available in public domain, can be used by everyone for sending messages and for signature verification. Private key is known by only one person, and it is important for decrypting messages and digital signature.

Key generation for RSA algorithm [2].

i. Take two prime numbers p and q.

   Prime integers p and q be taken randomly, and near same bit-length.

ii. Calculate number n = p q

iii. Calculate Euler's totient function equal to $(q - 1)\ (p - 1)$

   Same as $[n - (q + p - 1)]$.

iv. Take 'e' which is greater than 1 and less than Euler's totient function and gcd equal to 1.

   'e' is public key and of small bit-length.

v. Calculate d which is congruent to $[\ e^{-1} \ (\text{mod } \varphi(n))]$.

Solving for an example:

Take two prime numbers p = 17 and q = 11.

Compute n = p q = 17 * 11 = 187

Compute ϕ(n) = (p-1)(q-1) = 16 * 10 =160.

Pick e such that e is relatively prime to ϕ(n) = 160 and less than ϕ(n). picking e = 7.

Decide d such that d*e ≡ 1 (mod 160) and d < 160, value of d = 23

Finally, public key = (7,187) and private key = (23,187)

Encryption

'X' keeps his public key open for all and keeps the private key confidential. ‘Y’ wants to send message ‘m’ to ‘X’.

‘Y’ first uses public key of ‘X’ which is available for everyone.

Ciphertext that ‘Y’ sends is $m^e(\text{mod } n)$.

‘Y’ then sends ciphertext to ‘X’.

Decryption

When ‘X’ gets the ciphertext sent by ‘Y’, it uses its private key to decrypt it.

This is done by calculating c^d (mod n).

This is equal to the original message ‘m’.

A can recover the original message ‘m’ by reversing the scheme.

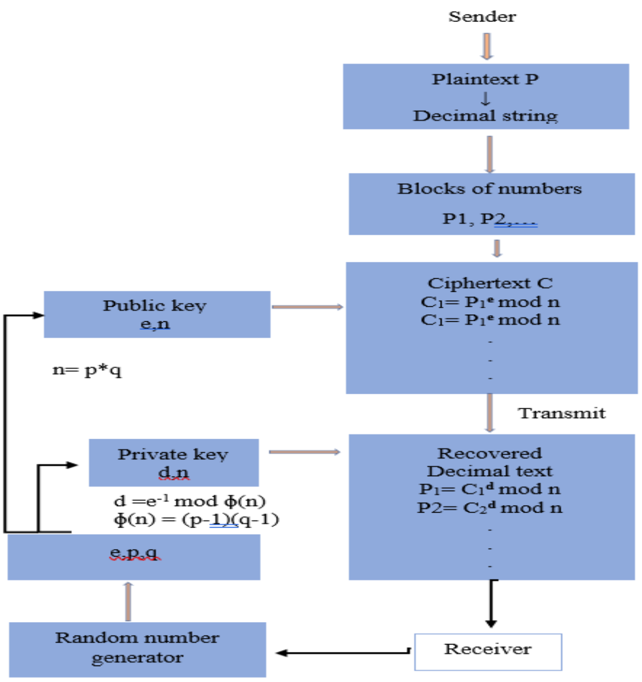

Fig. 1. RSA processing of multiple blocks

2.2 Primality Testing of a number

Miller Rabin primality testing [3],[4] is used to test the primality of a given number. For a given number ‘n’, we use this test to find whether this number is prime or not.

This used the property of primes that is explained below.

Properties of primes

Whenever p is prime and q is +ve odd integer and q less than p.

Subsequently a^2mod p = 1

only possible in two cases

(1) a mod p = 1

(2) a mod p = -1

let p be prime number >2,

then p -1 equal to 2^k q, where k>o,

if 'a' is integer such that 1< a < p-1,

Then any one of the two states is correct.

1. a^q mod p = 1

or equal to $a^q = 1 (\text{mod } p)$

2. $a^q, a^{2q}, a^{4q}, \ldots, a^{2^{(k-1)} q}$ is congruent to -1 mod p.

Miller Rabin Algorithm

Input: n>3, number that is to be tested for prime and it should be odd number.

Find k and q, k is positive number and q is odd

Using n -1 = 2^k.q

Taking random 'a' such that 'a' is positive and less then n-1.

If we get a^q mod n = 1, its outcome be

"undetermined" not able to tell it's not prime.

when j = 0 to (k - 1)

if $a^{2j\cdot}q$ mod n = n-1,then

then result is confirmed that the number is not prime, but composite.

3.0 RSA-KEM (RSA - Key Encapsulation Mechanism) Algorithm

RSA-KEM [1],[7],[13],[14],[38] is a hybrid RSA algorithm incorporating key encapsulation mechanism. It uses RSA trapdoor permutation along with a key derivation function (KDF). The RSA-KEM Key Transport Algorithm uses store-and-forward concept for sending data to a beneficiary. This is done by using the beneficiary RSA public key. Key Encapsulation Mechanism is a way to secure a symmetric key to send information from one place to another using public-key algorithm. In symmetric key, the encryption and decryption keys are available to both parties, one is sender and the other is receiver. The encryption key is known to all parties, and the decryption key is derived from it. In some cases only one key is used for the encryption and decryption. In public key algorithm encryption and decryption are slow and it is not reliable for transmitting long messages. In public key cryptography, one key is available in public domain, and other related key is kept secret. It is not possible to derive one key with use of the other key and algorithm. Public-key which is available in public domain, is known by everyone, and can be used to encrypt messages, and verify signatures. In private-key system only the receiver knows the private key, that is used to decrypt messages and sign signatures. Public key algorithms are good for exchange of symmetric keys. After exchange of key, this key is further use for encryption of long messages. Public-key system is used to exchange symmetric keys, which are relatively short, then this symmetric key is used to encrypt long messages. First a random symmetric key is generated, then it is encrypted using public-key algorithm. It is sent to the recipient, where it is decrypted to get the symmetric key. In RSA-KEM [8], password based key derivation function (PBKDF) is used for key derivation. Advanced Encryption Standard (AES) is used for key-wrapping. AES has 10 or more rounds depending on the key size. Rounds consist of substitution, transposition and XOR operations, and the output of AES is 128 bits long which is given as input to RSA. Complete workflow of RSA-KEM is given in Fig 2.

3.1 Algorithm

For sending a symmetric key with public key algorithm, first we

- Generate a random symmetric key
- Encrypt it with public key algorithm
- Send it to recipient
- Recipient decrypts it using public key and get symmetric key.

Generally symmetric key is small, we use padding for security purpose, but padding security is not secure. In KEM we first generate random element using finite field. Then we use hashing for deriving the symmetric key.

M = symmetric key 128 / 256 bits length,

we make it larger m, $1<m<n$, then

$c = m^e (\mathrm{mod}\ n)$

Alice can get m using

$m = c^d (\mathrm{mod}\ n)$

the M can be derived by reversing padding scheme used.

In place of generating random symmetric key M, Bob generates random m, $1 < m < n$.

Deriving symmetric key M = KDF (m)

$c = m^e (\mathrm{mod}\ n)$

$m = c^d (\mathrm{mod}\ n)$

then,

symmetric key M = KDF (m)

M is calculated from m, but reverse is not possible. KDF is one way function. If attacker somehow gets M, it would not be possible to get the plaintext m.

RSA-KEM creates a random integer ‘r’, manages symmetric encryption key by ‘r’ along key derivation function (KDF), then encodes ‘r’ with RSA.

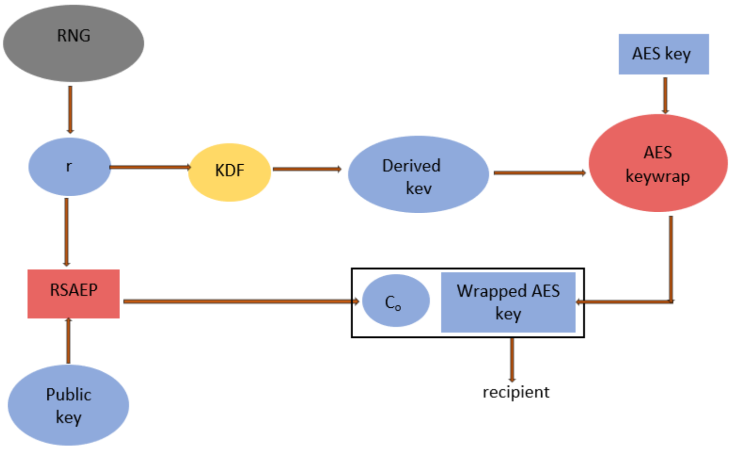

Fig. 2. RSA-KEM workflow

3.2 Password based key derivation function (PBKDF)

Traditionally, the password is hashed and stored in database, which can be easily attacked by Brute force or Rainbow table attack. Hashed password can be used as a key in encryption and decryption in 3DES, AES and other algorithms. As CPU and GPU processors are getting more powerful and faster, cracking password is easy. We make password hash harder to break by using salt. Salt is a large sequence of randomly generated data that is added with the password. We use password based key derivation function. Block diagram of password hashing is given in Fig 3.

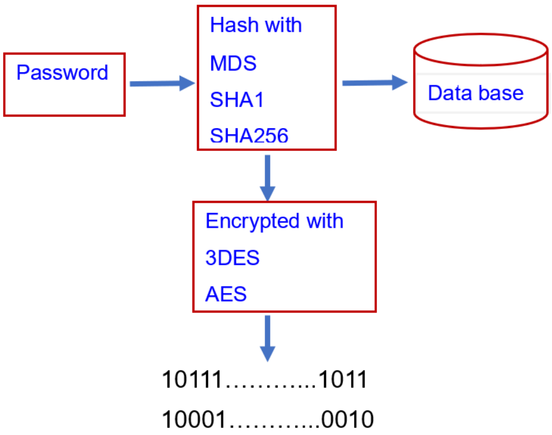

Fig. 3. Password hashing

Salt a password before hashing. It protect from Brute force attack but not from Rainbow attack. It can be attack by attacker using rainbow table, only rainbow table size increase by adding salt, as permutation of password added into table. Password based key derivation function [5] is use for security against brute force attack and rainbow table attack. In this method time takes to test each possible case in increased. Number of iterations is added. How many times its execute before returning the hash password. It's slow down the key generation and safe guard against rainbow table attack. For protection from Rainbow attack, we use Password based key derivation function (key stretching algorithm). Block diagram of PBKDF is given in Fig 4.

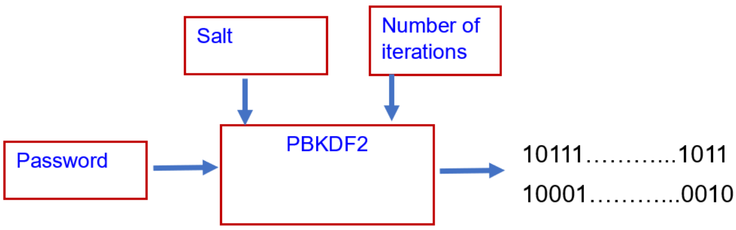

Fig. 4. Password based key derivation function PBKDF

3.2.1 Algorithm

PBKDF has these input parameters:-

DK = PBKDF (Password, salt, c, sha256, dklen)

Password = like manish, India, killer…

C is number of iteration > 1000

salt is a sequence of bits, 32 bits

dklen is length of key we want to generate, 128 bits

DK is derived key

Derived key DK = K1 + K2 + K3 +… Kdklen/hlen

Ki= F (Password, Salt, c, i)

Function F is XOR of pseudorandom function

F(Password, Salt, c, i) = J1 ^ J2 ^… ^Jc

Ji = PRF (Jprevious)

3.3 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) [14],[15] is used for key-wrapping scheme. 128 bits / 192 bits or 256 bits of block size, and 128 bits of key as a input to AES is required. AES has 10 or more rounds depending on key size. Rounds consist of substitution, transposition and XOR operations, and output of AES is 128 bits, which is used as the input to RSA. Block diagram of AES and Key size with rotations is given in Fig 5 and Table 3 respectively.

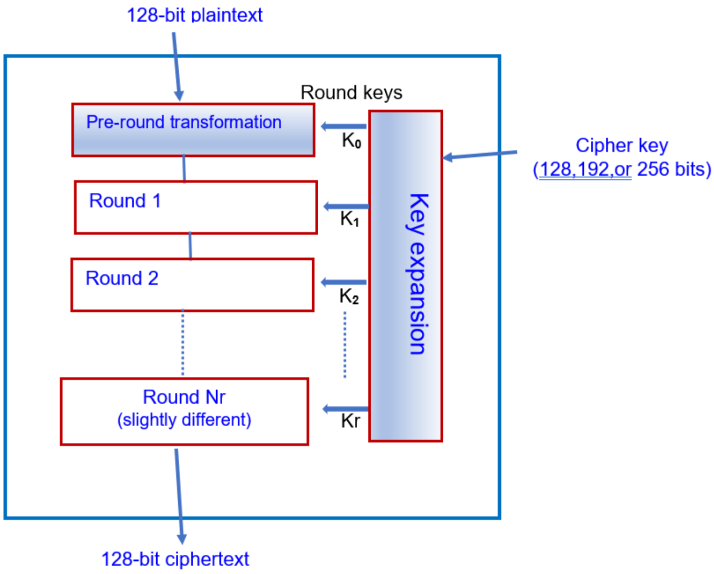

Fig. 5. Block diagram of Advanced Encryption Standard

r	Key size
10	128
12	192
14	256

Table 3. Rotation and key size

Steps of AES [14,15]

a. Substitution Bytes

b. Shift Rows

c. Mix Columns

d. Add round key

a. Substitution Bytes:

In this each byte ai,j is replaced by S(ai,j). Every byte is changed by a different value. S-box is used which is multiplicative inverse over Galois Field GF(2^8) for changing the value that is replaced.

b. Shift Rows :

In this circular right shift is done. Each row is shifted by fixed number of bytes. For the first row (R_0), we do nothing. In second row (R_1),1 bit is shifted, in the third row (R_2), 2 bits are shifted, and so on. Rows and their rotations are given in Table 4.

1st Row	Rotated by 0 bytes
2nd Row	Rotated by 1 bytes
3rd Row	Rotated by 2 bytes
4th Row	Rotated by 3 bytes

Table 4. Rows and their rotations

c. Mix Columns:

In this segment, each column is operated individually for generating a new column. Column's bytes and pre-defined matrix is used for output. Multiplication is done with column's bytes and pre-defined matrix.

d. Add round key:

In this segment, the output of the mix column is XORed with the key. Output of this stage is given as input for the next round of operation. If this is the last round then it is ciphertext.

3.3.1 Algorithm

Initial round

- AddRound key- Bitwise XOR is done on each byte of round key and states

Main round

- SubBytes- Each byte ai,j is replaced by S(ai,j), using S-box.
- ShiftRows- Circular right shift is done.
- mixColumns – Multiplication with column's bytes and pre define matrix.
- AddRound Key - Output of the mix column is XORed with key.

Final round

- SubBytes
- ShiftfRows
- AddRound Key

An example of AES [38], with the plaintext (in hexadecimal), key and resulting ciphertext, is shown below. Expansion of 16 byte key into 10 rounds keys is given in Table 5, and progression of state through AES encryption process is in Table 6.

Plaintext	0123456789abcdeffedcba9876543210
Key	0f1571c947d9e8590cb7add6af7f6798
Ciphertext	ff0b844a0853bf7c6934ab4364148fb9

Key Words	Auxiliary Function
W0 = 0f 15 71 c9 W1 = 47 d9 e8 59 W2 = 0c b7 ad d6 W3 = af 7f 67 98	RotWord (w3) = 7f 67 98 af SubWord (x1) = d2 85 46 79 Rcon (1) = 01 00 00 00 Y1 ⊕ Rcon (1) = d3 85 46 79 = z1
W4 = w0 ⊕ z1 = dc 90 37 b0 W5 = w4 ⊕ w1 = 9b 49 df e9 W6 = w5 ⊕ w2 = 97 fe 72 3f W7 = w6 ⊕ w3 = 38 81 15 a7	RotWord (w7) = 81 15 a7 38 = x2 SubWord (x4) = 0c 59 5c 07 = y2 Rcon (2) = 02 00 00 00 Y2 ⊕ Rcon (2) = 0e 59 5c 07 = z2
. . .	. . .
W36 = w32 ⊕ z9 = fd 0d 42 cd W37 = w36 ⊕ w33 = 0e 16 e0 1c W38 = w37 ⊕ w34 = c5 d5 4a 6e W39 = w38 ⊕ w35 = f9 6b 41 56	RotWord (w39) = 6b 41 56 f9 = x10 SubWord (x9) = 7f 83 b1 99 = y10 Rcon (10) = 36 00 00 00 Y10 ⊕ Rcon (10) = 49 83 b1 99 =z10
W40 = w36 ⊕ z10 = b4 8e f3 52 W41 = w40 ⊕ w37 = ba 98 13 4e W42 = w41 ⊕ w38 = 7f 4d 59 20 W43 = w42 ⊕ w39 = 86 26 18 76	

Table 5. Expansion of 16 byte key into 10 round keys.

Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key
01 89 fe 76 23 ab dc 54 45 cd ba 32 67 ef 98 10				0f 47 0c af 15 d9 b7 7f 71 e8 ad 67 C9 59 d6 98
0e ce f2 d9 36 72 6b 2b 34 25 17 55 ae b6 4e 88	ab 8b 89 35 05 40 7f f1 18 3f f0 fc E4 4e 2f c4	ab 8b 89 35 40 7f f1 05 F0 fc 18 3f C4 e4 4e 2f	B9 94 57 75 E4 8e 16 51 47 20 9a 3f C5 d6 f5 3b	dc 9b 97 38 90 49 fe 81 37 df 72 15 B0 e9 3f a7
. . .	. . .	. . .	. . .	. . .
Cc 3e ff 3b A1 67 59 af 04 85 02 aa A1 00 5f 34	4b b2 16 e2 32 85 cb 79 F2 97 77 ac 32 63 cf 18	4b b2 16 e2 85 cb 79 32 77 ac f2 97 18 32 63 cf	4b 86 8a 36 B1 cb 27 5a Fb f2 f2 af Cc 5a 5b cf	B4 8e f3 52 Ba 98 13 4e 7f 4d 59 20 86 26 18 76
Ff 08 69 64 0b 53 34 14 84 bf ab 8f 4a 7c 43 b9				

Table 6. Progression of state through AES encryption process.

4.0 Impact of Quantum Computing on Cryptography[18,19,21,27,37]

We have always in mind that is quantum computer is a replacement of classical computer in present scenario. Quantum computer only faster in a special type of calculation. It done computation in parallel. It will not affect activity like browsing internet, writing documents, watching HD videos. Searching a particular detail of a number in a telephone directory, like find the person which number belongs to him. If the entry in the telephone dictionary is one million then in quantum computer required square root steps. In classical computer two bit can represent any one of four 00, 01,10,11. Four numbers but any one can use i.e. two bits information. But in quantum mechanics it is possible to make superposition of each one of these four states. To find out state of two spin system four coefficients or numbers required, but in classical for two bits only two numbers. Two qubits hold four bits of information. For three qubits systems having eight different states. For 'n' qubits system is analogous to $2^{\mathbf{n}}$ classical bits. Qubits exits any of the combination of states, but when we measured, it fall any one of the basis states. We cannot compute superposition, only compute basis states (up or down). The present knowledge we had, the most possible architecture of a quantum computer might be able to break RSA 2048 bits required about 20 million physical qubits. Because we need error correction, we can't do it with 50 or 100 qubits, because we lacking the ability to correct the error. We can do it with few thousand perfect qubits of zero error, but we never do it, because we always have errors. But tomorrow anyone can come with better quantum algorithm or better quantum error correction code. Then it possible with less number of qubits required to break RSA 2048. Both qubits and gates must be error free. But as on today, there is no perfect qubits. Number of qubits required to break RSA 2048. Quantum teleportation, quantum entanglement and other makes it possible to break present cryptosystem.

a. Quantum teleportation[22,23]

If someone want to send quantum information to other person. He cannot send quantum states as he cannot do copy of the quantum states. He can use entangled qubit and classical bits for transfer the stat, that is called quantum teleportation. First party do some operation on his qubits and send to second party, after receiving results, second party do some operations on it. This way information is transported. Two particles (photons) which are entangled are shared in two different location irrespective of distance between them, information can be teleported. This involve only transportation of quantum states not physical states. In today world teleportation up to 44 kilometre long with more than 90% accuracy is done in fiber optic network and 1200 km using satellite arrays.

b. Quantum entanglement [22,23]

It is a quantum mechanical phenomenon where two or more object's quantum states relate to each other irrespective of distance between them. If we have two entangled particles (photons, electrons, molecules etc) then if one is detect in one direction then other particle must be detect in opposite direction. If entangled particles have total spin zero, then if one particle's spin is in clockwise then other particle spin must be in anti-clockwise. Entangled photons is used in quantum holography also. At present a photon entangled with an ion is send 50 km long in optical fiber.

c. Quantum superposition

Separate unrelated quantum states exist in same time of a quantum system. It's a union of definite quantum states. Qubits may be in a superposition of both basis sates of |0> and |1>. 'n' qubits may be in a superposition of 2^{n} states. At quantum level particles act like waves. Just like various waves overlaps each other, quantum particles also do overlapping to form a unique wave.

4.1 Current Industry methods for making Qubits [29,30,31,37]

a. Silicon based Qubits

In this electron put inside Nano material is used as a transistor. By doping pure silicon with Group V elements such as phosphorus, extra valence electrons are added that become unbonded from individual atoms and allow the compound to be an electrically conductive. Using silicon-based CMOS (complementary metal-oxide-semiconductor) technology for making Quantum Qubits. Using silicon and phosphorus atom for making qubits. In silicon qubits it provides less noisy environment. More than 95% of Silicon that is available naturally have nuclear spin-0. Phosphorus impurities use as a doner. Crystalline silicon and with phosphorus atoms can be used, spin qubits can read by nuclear magnetic resonance techniques. Drain 'D' and source 'S' is made of modified silicon having impurities, Fig-3. When concentration is high more electrons are present. Highly metallic silicon electrode is used. When we apply voltage electrons accumulate at insulator surface which is in between two metallic silicon. Size can reduce to very small in nanometre that just hold few electrons.

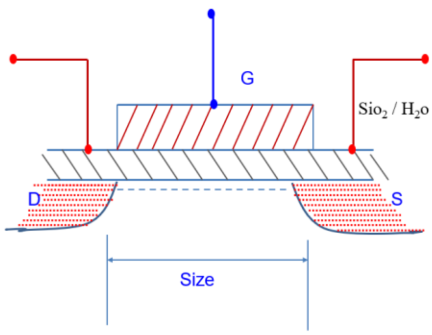

Fig. 6. Drain and Source in silicon CMOS

b. Superconducting circuit

Superconducting circuit is used by Google, IBM, Intel, Microsoft. This is most advanced technology. Insulator is used as a sandwich in between two metal layers. Is called Josephson junction. This use as a controller of energy level. As temperature decreased, electrical resistivity decreases in metallic conductors. At below critical temperature resistance of superconductor become zero. In a loop of superconducting wire, a electric current flow with no power source.

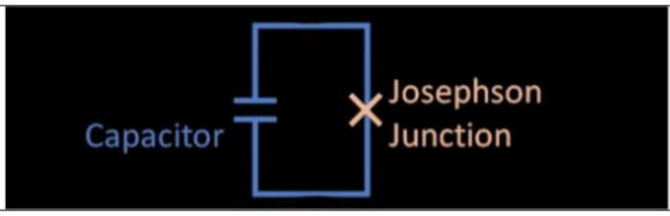

Fig. 7. Capacitor with Josephson Junction

c. Flux qubits

Flux qubits is used by D wave company. In this code 0 and 1 is given as, current flow clockwise or anticlockwise direction. Current flowing in superposition of clockwise and anticlockwise. It's a very small size (micro meter) loop of superconducting metal. Operations are done by using microwave radiation on qubits and that energy is corresponding to the gap of the two basis states. Appropriately selected frequencies set qubit into quantum superposition. Flux qubit state is measured by superconducting quantum interference device (SQUID).

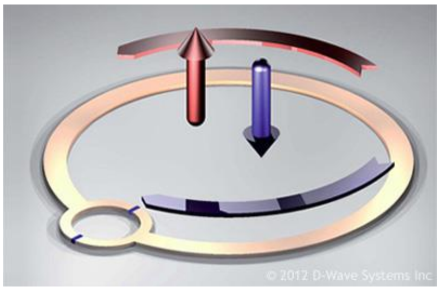

Fig. 8. Superconducting Qubit

4.2 Shor's Algorithm [35,36]

With quantum mechanics it is possible for factorization of large number into its prime factors in polynomial time (O(log N)) using Peter Shor's factorization algorithm, previously it takes exponential time $(O(\log N)^{k})$ in classical methods [26,33]. This is big threat for data security. It consists of both classical part as well as quantum part. In classical part we convert the problem of factoring into finding the period problem, and for finding the period we use quantum Fourier transform which is in quantum part.

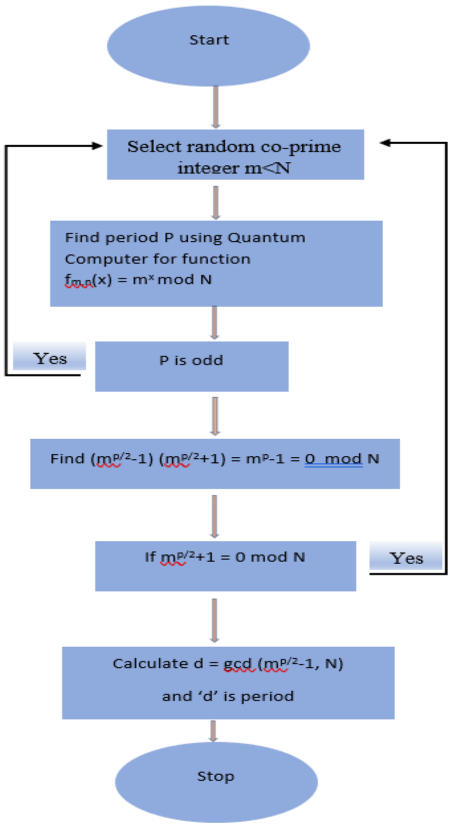

Fig. 9. Flow chart of algorithm

Quantum part of Shor's algo. (Order finding)

Select a power of 2,

$Q = 2^L$ such that $N^2 < Q < 2N^2$

'f' restricted to $\{0,1,2,\ldots,Q-1\}$

Where $f(y) = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1} |f(x) > \omega^{xy}$

1 Initial state of Register1(R_1) and Register2(R_2)

$|\psi_0> = |R_1> |R_2> = |0> |1>$

2 Applying Fourier transform to R_1

$$|\psi_0 > = |0 > |1 > \xrightarrow{f \otimes I} |\psi_1 > = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1} |x > |1 >$$

3 Applying unitary transformation Uf to R2

$$|\psi_1 > = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1} |x > |1 > \xrightarrow{Uf} |\psi_2 > = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1} |x > |f(x) >$$

4 Applying Fourier transform to R_1

$$|\psi_2 > = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1} |x > |f(x) > \xrightarrow{f \otimes I} |\psi_3 > = \frac{1}{Q}\sum_{x=0}^{Q-1} \sum_{y=0}^{Q-1} \omega^{xy} |y > |f(x) >$$

5 For 'y' we measure register 1 and using continued fractions for y/2L we get period P.

4.3 Quantum Proof Algorithm[20,24,25]

These families of crypto algorithm are considered quantum proof algorithms.

a. Lattice based

b. Code based

c. Hash based

d. Multi variate

a. Lattice-based cryptography

lattice is set of intersection point in the space and these points are defined by parallel and equidistance lines going in two-dimensional space. Each intersection point is called lattice. Lattice field is defined by two vectors, called base vectors. Different bases can be used to define same lattice field.

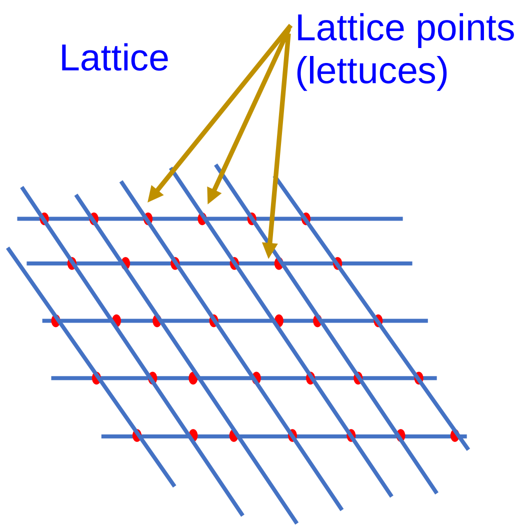

Fig. 10. Lattice and Lattice points

good base is almost orthogonal. Bad base is almost parallel. Good and bad base can be define in same lattice field.

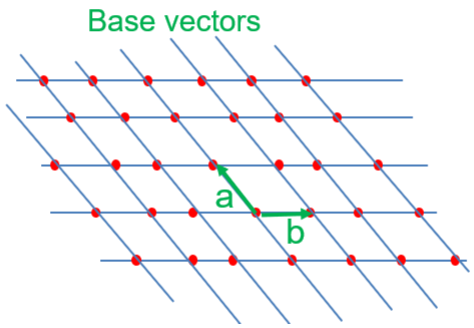

Fig. 11. Good base vector

“Good” base

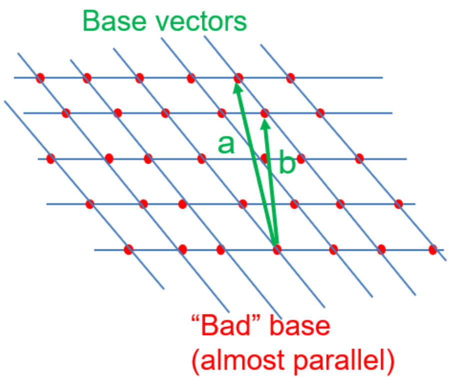

Fig. 12. Bad base vector

Which lattice is closest from a given point in two-dimensional space, it is easy to get, but if the lattice field has 250 dimensions? It is extremely difficult to find closest lattice. Answer is easy if we have good base but it is difficult to answer if we have bad base. This is the concept behind lattice-based cryptosystem.

- **Goldreich- Goldwasser-Helevi Encryption (GGH)**

Alice private key is a good base in a lattice field. Alice public key has bad base define in same lattice field. Encoding the message is not difficult but decoding is extremely difficult. Alice can decrypt because she knows good base, attacker cannot decrypt because he knows only bad base. this way (GGH) works, it is a quantum proof.

- **Learning with errors LWP method**

We have system of equation, It can be solved by Gauss elimination method or by modular arithmetic. if we have more equation then variable, It is Over defined system. we are talking over defined system where a solution exists.

system of linear equations
294.x + 629.y + 321z = 38
701.x + 29.y + 91z = 462
613.x + 339.y + 201z = 636

Table 6 System of equations

system of linear equations	Modulo arithmetic
294.x + 629.y + 321.z = 38	(mod 797)
701.x + 29.y + 91.z = 462	(mod 797)
613.x + 339.y + 201.z = 636	(mod 797)

Table 7: System of equations with x,y,z

system of linear equations	Modulo arithmetic
294.x + 629.y + 321.z = 38	(mod 797)
701.x + 29.y + 91.z = 462	(mod 797)
613.x + 339.y + 201.z = 636	(mod 797)
256.x + 94.y + 115.z = 522	(mod 797)
704.x + 629.y + 322.z = 477	(mod 797)
391.x + 23.y + 743.z = 213	(mod 797)

290.x + 620.y + 201.z = 40	(mod 797)
211.x + 339.y + 381.z = 510	(mod 797)

Table 8: System of linear equations

In this Alice's private key is solution of the equation. In right side of the equation we add errors like + 1 - 2 - 1 + 2 adding very small errors and hide this error errors. We can find errors without knowing X, Y and Z but is a very laborious work. This leads to a trapdoor function, It is easy to compute in one direction but difficult in other direction, this is called learning with errors trapdoor function. Adding Errors is easy but finding error is difficult unless we know X Y and Z (variables value). This is known as Regev encryption.

Adding errors in equations	
294.x + 629.y + 321.z = 38 +1	(mod 797)
701.x + 29.y + 91.z = 462 -2	(mod 797)
613.x + 339.y + 201.z = 636	(mod 797)
256.x + 94.y + 115.z = 522 +1	(mod 797)
704.x + 629.y + 322.z = 477	(mod 797)
391.x + 23.y + 743.z = 213 -1	(mod 797)
290.x + 620.y + 201.z = 40 +2	(mod 797)
211.x + 339.y + 381.z = 510 +1	(mod 797)

Table 9: Adding errors in equations

and have public key is equation system itself with incorrect solutions, added small errors on the right side.

New added equation can be used to encrypt one bit.

For Encrypt '0'

Add small errors to the result of equations.

For Encrypt ‘1’

Add big number (big error) to the result of equations. This way one bit is encoded. If Bob encrypt something, he selects some equations and left other equations. this is a random process, generally half of the equations are left, then add all the equations we have. Alice known value of variable X, Y and Z. She can easily check whether there is a small or big error. A small error means 0 and big error is means it is 1.

Added errors in equations	
294.x + 629.y + 321.z = 39	(mod 797)
613.x + 339.y + 201.z = 636	(mod 797)
290.x + 620.y + 201.z = 42	(mod 797)
400.x + 791.y + 723.z = 717	**(mod 797)**

Table 10: Added errors in equations

for attacker it’s very difficult to decrypt because he needs to invert learning with errors trapdoor function. It’s a quantum proof algorithm but only encrypt one bit at a time. they are more efficient variant of learning with errors.

b. Code based cryptography

Its start with error correcting codes. Parity bit (an error detecting code). Three-times code (its error connecting code but not very efficient). We need better error correcting code. For this linear error correcting codes are better alternative.

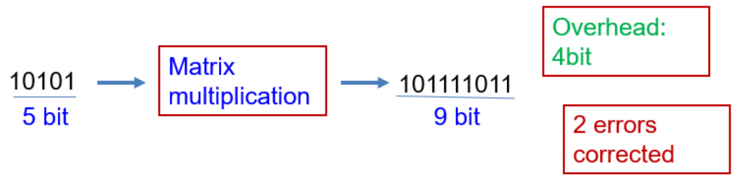

Fig. 13. Code based cryptography

In general, overhead of ‘n’ bits, ‘n/2’ errors corrected. For error correction a error correction algorithm is used.

For multiplication different matrixes can be used. Some matrix’s allow for efficient error correction (good matrix) but most matrix’s does not (bad matrix). A good matrix can be changed into a bad metric if multiply by blend Matrix. This can be used for encryption and in this length of public key equal to 1 Mb, but in RSA public key 2 kb, but it is quantum proof.

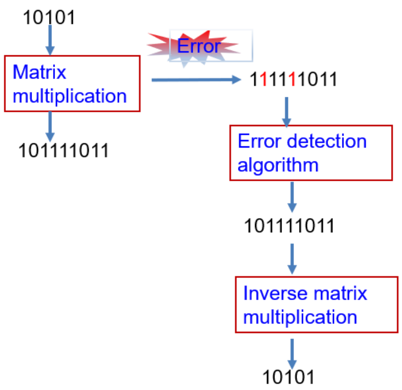

Fig. 14. Code based cryptography

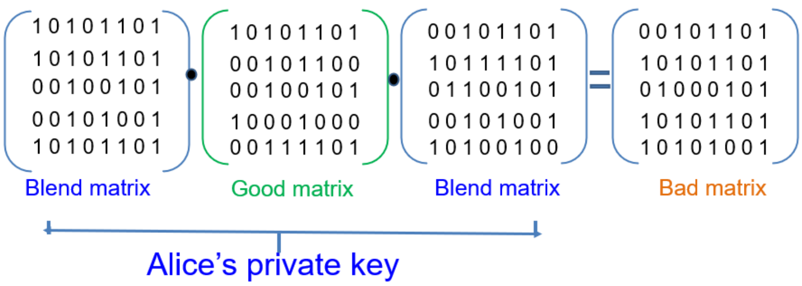

Fig .15. Encryption and Decryption

c. Hash methods[28,34]

Hash based signatures scheme have long signatures or keys, but they are probably secure. One of the scheme is Lamport Signature[32]. We use RSA, digital signature algorithm for sign messages. But after quantum computers these scheme are not safe. One method that is quantum robust is Lamport signature given by Leslie B. Lamport.

In this

- We make two sets A and B of 256 random 256-bit numbers. The private key value is 512.
- Taking hash of every numbers. The public key is 512 hashes.
- Using SHA-256 we hash the message. For 0 we take from set A, for 1 we take set B for ith number.
- Then 256 random number is the signature. And public key is 512 hashes.

Lamport method is use single time for signing. Using hash tree we can do multiple time signing.

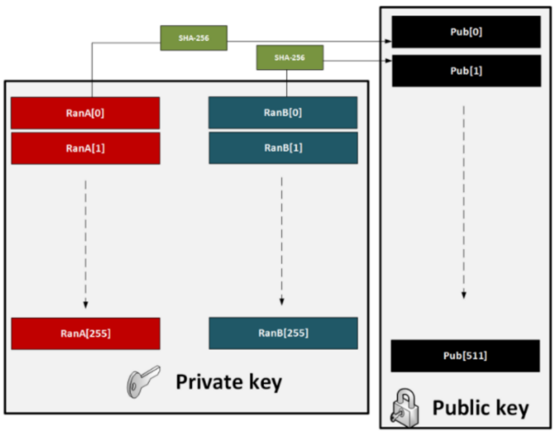

Fig .16. Encryption & Decryption in Lamport

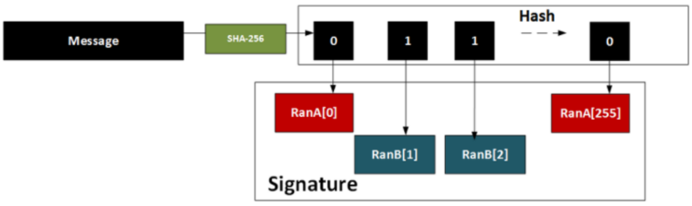

Fig .17. Encryption & Decryption in Lamport

d. Multivariate algorithm

In multivariate public key cryptosystem, public key are set of multivariate polynomials. Complexity to solve system of multivariate equations is idea behind this. Its used for signatures. One of the schemes is unbalanced oil-and-vinegar scheme. Unbalanced oil and vinegar scheme is used for digital signature. Its security based on NP-hard problem. Finding solution of 'm' equations with 'n' variables is NP-hard problem. if m is larger or smaller than n, its easy comparable when both m and n are equal.

To make a effective signature, solution of these equations required.

$y1 = f1\ (x1, \ldots, xn)$

$y2 = f2\ (x1, \ldots, xn)$

.

.

.

$ym = fm\ (x1, \ldots, xn)$

here $y = (y1, y2, \ldots, ym)$ is message that is signed.

The effective signature is $x = (x_1, x_2, \ldots, x_n)$.

first message is change to suited in equation system. Each single equation has form

$y_i = \sum \gamma_{ijk}\, a_j\, a'_k + \sum \lambda_{ijk}\, a'_j\, a'_k + \sum \xi_{ij}\, a_j + \sum \xi'_{ij}\, a'_j + \delta_i$

each coefficients γ_{ijk}, λ_{ijk}, ξ_{ij}, δ_i taken in secret.

Vinegar variable a'_j is selected randomly.

Solution of derive linear system of equation give us a_i.

Signature validation is done by public key

$y_1 = f^*_1 (x_1, \ldots, x_n)$

$y_2 = f^*_2 (x_1, \ldots, x_n)$

.

.

.

$y_m = f^*_m (x_1, \ldots, x_n)$.

Attacker not access to the coefficients, oil and vinegar variables. Each equation has to solve for signature verification.

5.0 Results

We have implemented RSA and RSA-KEM algorithm using GNU multi precision library in Linux platform. The application uses a 512 bit modulus RSA implementation, which is sufficient for non-critical applications. The libraries GNU MP Arbitrary Precision library (C/C++) and Open SSL crypto library (C/C++) are used. The GMP library is a cross-platform library, implying that our application should work across platforms with least modifications.

In RSA algorithm we take two prime numbers, p=53 and q=59 as the input. Then n=3127. 'Y' take e=3, Finally, 'Y' chooses d=2011, Values n=3127 and e=3 is public (3, 3127) and value d=2011 secret (2011,3127). 'Y' wants to send the letters 'm', 'a', 'n', 'i', 's', 'h' to 'X'. Putting letter as a number from 1 and 26 ('a'=1 and 'z'=26). 'Y' and 'X' perform encryption and decryption as in Table 7 and Table 8. Input as "manish" is given in RSA and its encryption and decryption is given in Fig 6, Fig 7 and in Fig 8.

Message letter	**Corresponding Number 'm'**	**m^e**	**Encrypted message m^e mod n**
m	13	2197	2197
a	1	1	1
n	14	2744	2744
i	9	729	729
s	19	6859	605
h	8	512	512

Table 11. Y's encryption using e=3, n = 3127

Encrypted message m^e mod n	**c^d**	**c^d mod n**	**Original message letter**
2197	2.631749003395505379259667456847 1e+6720	2197	m
1	1	1	a

2744	3.894388255334056297358066304617 7e+6914	2744	n
729	8.811694817037149859498962689621 8e+5756	729	i
605	1.288422872982337407264299668195 8e+5594	605	s
512	2.197310962228713700992556454807 58e+5448	512	h

Table 12. X's decryption using, d=2011, n=3127

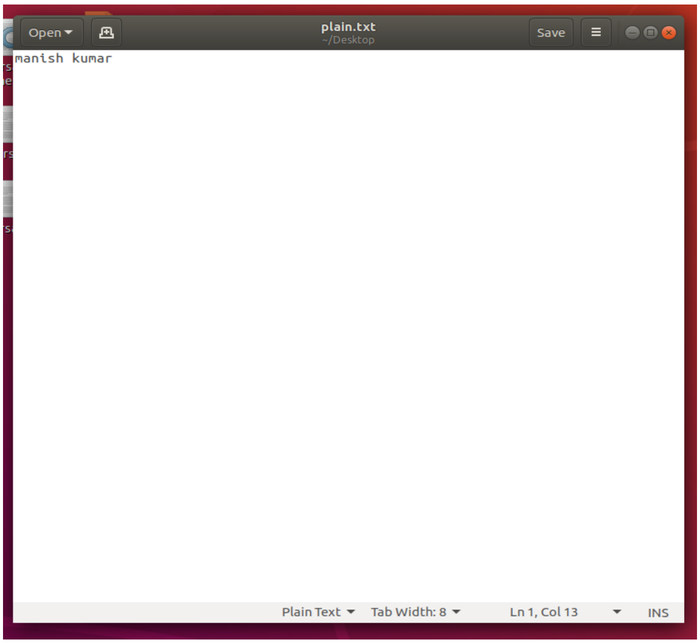

Fig.18. Plaintext for RSA algorithm

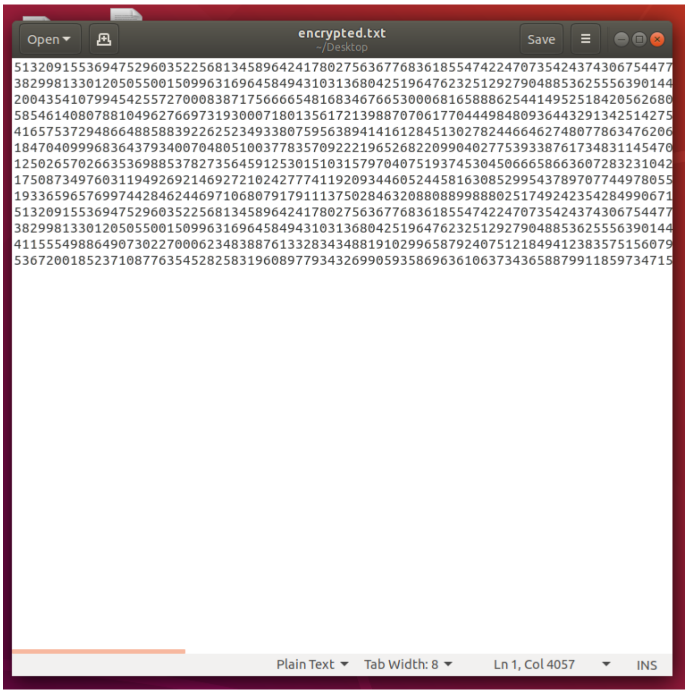

Fig.19. Encrypted output of plaintext for RSA algorithm

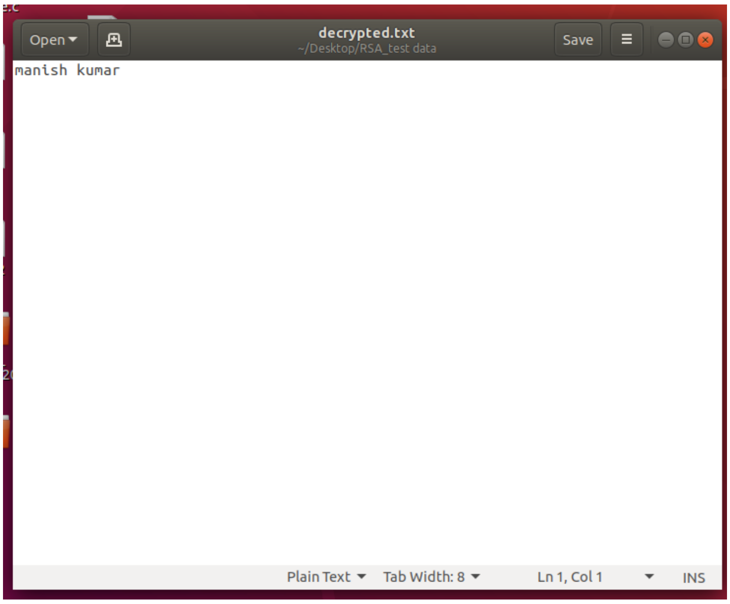

Fig. 20. Decrypted output of RSA algorithm

In Miller Rabin primality testing we gave input of 20 digits, 24 digits and 25 digits numbers and check the primality of the numbers; test results are given in Table 9 and the screenshot of Miller Rabin test is given in Fig 9.

No of digits	**Number**	**Output**
20	10013236879455627894	Composite
20	10089886811898868001	Prime
24	250000000000000000000015	Composite
24	253977540775422754427545	Prime
25	1000000000000000000000061	Composite
25	1015910163101691017710181	Prime

Table 13. Test results of Miller-Rabin Algo

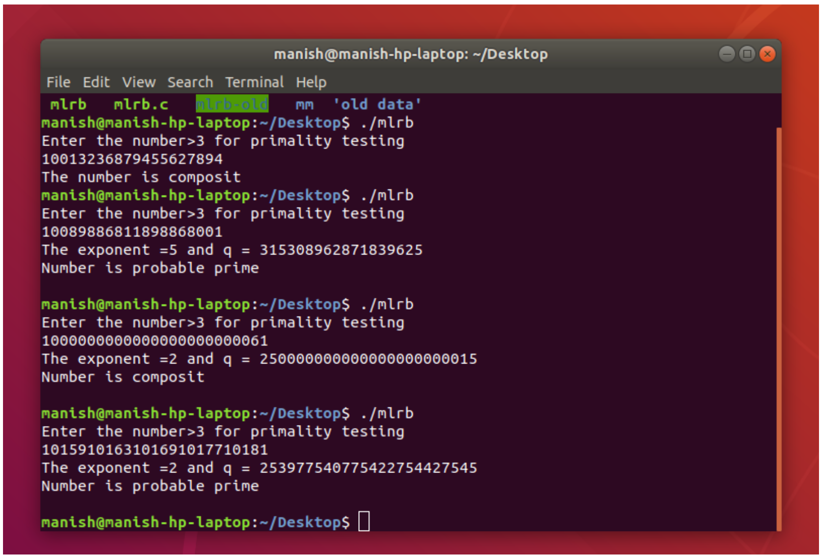

Fig. 21. Decrypted output of RSA algorithm

In Password based key derivation function we gave input plaintext as delhi, technological, university and the outputs are shown in Table 10. Number of iterations are taken as 1000 and salt is a 32 bit sequence.

Sl.No.	Plaintext	Hash function output (PBKDF)
1.	delhi	2dc4518bfdeeb54a48136242b0df4470d9d2933bee1593e2d46912d0d69c1ede
2.	technological	2b0e27f4d53899137f00daacb0feeb9d3438483fc48a87daf6f81866ccefe629
3.	university	362e2e0b74f956e3e595f76b2f897798ea66b5ca4cfa8d83d174f3354128f54c

Table 14: Output of PBKDF

In AES, we use key as “manish dtu” and plaintext as “info system” and when the ciphertext is decrypted we get the same plaintext. For different keys and plaintexts, the output is given in Table 11.

Sl. No.	Key	Plaintext	Ciphertext	Decrypted ciphertext
1.	manish dtu	info system	B72243ff5024b7bcd8a83b90d5c47484	info system
2.	manish kumar	this is secret	65eb480610e6ed25414f78707fdbbb	manish kumar
3.	I am key	Database2	A222960c3cfc2dfa14aa8be681c7a2d	I am key

Table 15: Results of AES

Output of AES is given as input for RSA module. Output of RSA is sent to recipient. Getting message from sender, recipient decrypts it using its private key and gets the symmetric key. Screenshot of AES is given in Fig 10.

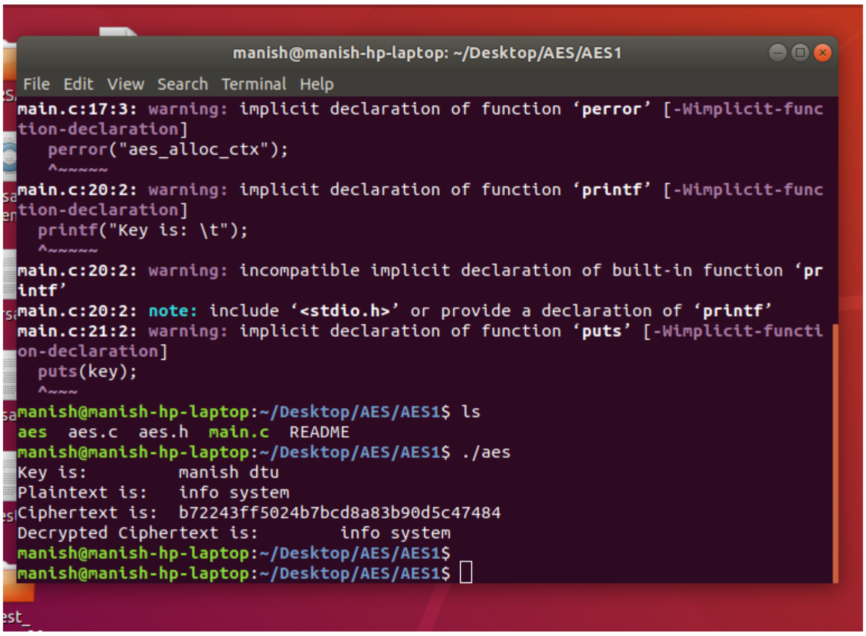

Fig. 22 Screenshot of output of AES

Example for number 323 and 15

Enter the RSA number of the form p*q

323

The coprime number selected is:

a = 16

The one factor of the RSA number is:

P = 19

The other factor of the RSA number is:

q = 17

the number has been factored

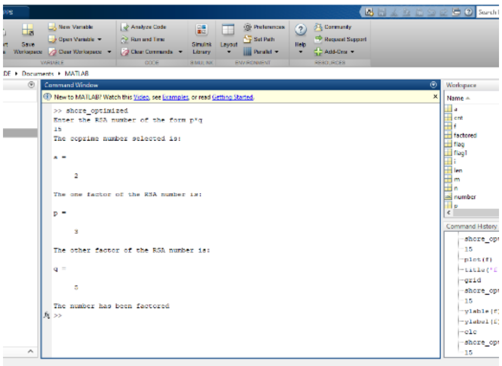

Fig. 23 Output for n=15

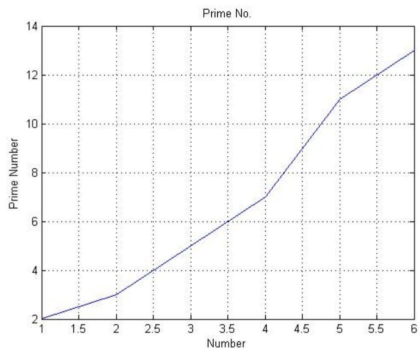

Fig.24 Prime number plot n=15

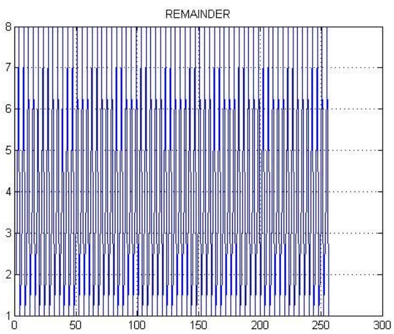

Fig. 25. Remainder plot for n= 15

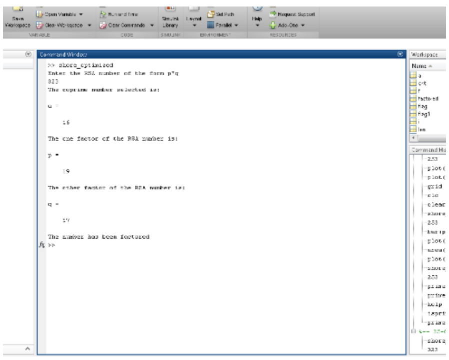

Fig .26. Output for n=323

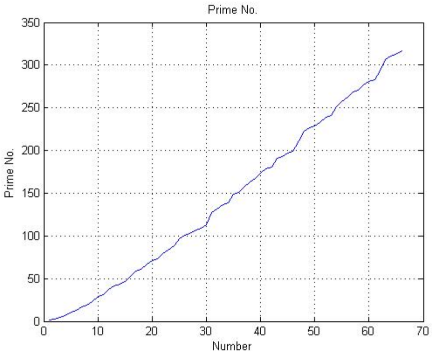

Fig .27. Prime number plot n=323

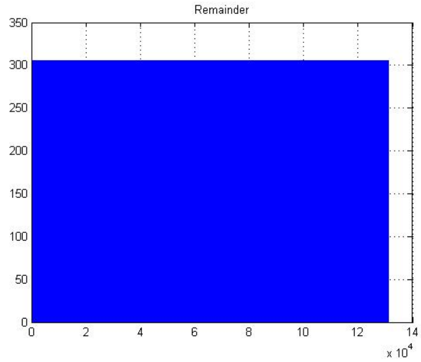

Fig .28. Remainder plot for n= 323

6.0 Conclusion

It is hard to factor a large prime number which is a product of two prime numbers. For primality testing in RSA, Miller-Rabin algorithm is used. It takes the integer as an input and test whether that number is prime or not. RSA contains three functions, Key generation, Encryption and Decryption, It takes number of bits for n where (n=pq) and number of bits for e (public exponent) as a input. In RSA-KEM, password based key derivation function is used for key stretching algorithm. Password, salt, iteration is used for generation of derived key. Advanced Encryption Standard is used for key-wrapping scheme. AES has 10 or more rounds depending on the key size. Rounds consist of substitution, transposition and XOR operations, and output of AES is 128 bits long, which is used as an input to RSA. Future work should be in quantum computers since the architecture of a quantum computer might be able to break RSA. Shor's Algorithm is used for integer factorization which is polynomial time for quantum computer. This can be threat for RSA security. Classical cryptography can be broken by quantum computer. All current systems would be on threat. Quantum computing is a fascinate area of research. Quantum computer is not a replacement of classical computer. It will not affect activity like browsing internet, writing documents, watching HD videos. In Quantum computer number of operations required to arrive at result is exponentially small. Improvement is not in speed of individual operation, it is the total number of operations is needed for arrival at result. Its only useful in arrival of results only in some particular type of cases. Implemented Shor's algorithm in matlab is done. We used classical methods for getting few results because classical computers not engage quantum phenomena. Modification also done to put in Fast Fourier Transform for getting period of function. As number of iterations grow, probability of getting exact factor of 'n' acutely increased. Getting non trivial factor of 'n' and random variable selected both are not correlated to each other. Many new ideas and innovation are arriving daily, many modifications of Shor's original algorithm are present that required less run on quantum computer. Quantum computer with number of qubits increasing daily, we have 72 qubits quantum computer today but in near future it cross thousands of qubits and possible to factor large composite numbers or break RSA 2048. For safeguard from quantum computer effect we have many quantum safe algorithm. In future we will see these quantum proof algorithms are widely used in every field, where security is concern.

7.0 References

[1] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21, no. 2 (1978): 120-126.

[2] Kaliski, Burt. "The Mathematics of the RSA Public-Key Cryptosystem." RSA Laboratories (2006).

[3] Agrawal, Manindra, Neeraj Kayal, and Nitin Saxena. "PRIMES is in P." Annals of mathematics (2004): 781-793.

[4] Stallings, William. Cryptography and network security: principles and practice. Upper Saddle River: Pearson, 2017.

[5] Percival, Colin, and Simon Josefsson. "The scrypt password-based key derivation function." IETF Draft URL: http://tools. ietf. org/html/josefsson-scrypt-kdf-00. txt (accessed: 30.11. 2012) (2016).

[6] Bellare, Mihir, Alexandra Boldyreva, and Silvio Micali. "Public-key encryption in a multi-user setting: Security proofs and improvements." In International Conference on the Theory and Applications of Cryptographic Techniques, pp. 259-274. Springer, Berlin, Heidelberg, 2000.

[7] Abe, Masayuki, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. "Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM." In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 128-146. Springer, Berlin, Heidelberg, 2005.

[8] Randall, James, Burt Kaliski, John Brainard, and Sean Turner. "Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS)." Proposed Standard 5990 (2010).

[9] Boldyreva, Alexandra, Hideki Imai, and Kazukuni Kobara, "How to Strengthen the Security of RSA-OAEP." *IEEE transactions on information theory* 56, no. 11 (2010): 5876-5886.

[10] Jonsson, Jakob, and Burt Kaliski. *Public-key cryptography standards (PKCS)#1*: RSA cryptography specifications version 2.1 RFC 3447,February,2003.

[11] Brzuska, Christina. "On the foundations of key exchange." PhD diss., Technische Universität, 2013.

[12] Fujisaki, Eiichiro, and Tatsuaki Okamoto. "Secure integration of asymmetric and symmetric encryption schemes." In Annual International Cryptology Conference, pp. 537-554. Springer, Berlin, Heidelberg, 1999.

[13] Randall, James, Burt Kaliski, John Brainard, and Sean Turner. "Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax (CMS)." Proposed Standard 5990 (2010).

[14] Jonsson, Jakob, and Matthew JB Robshaw. "Securing RSA-KEM via the AES." In International Workshop on Public Key Cryptography, pp. 29-46. Springer, Berlin, Heidelberg, 2005.

[15] Mahajan, Prerna, and Abhishek Sachdeva. "A study of encryption algorithms AES, DES and RSA for security." Global Journal of Computer Science and Technology (2013).

[16] Chitra, Ms K., and V. PRASANNA Venkatesan. "An antiquity to the contemporary of secret sharing scheme." Journal of Innovative Image Processing (JIIP) 2, no. 01 (2020): 1-13.

[17] Shakya, Subarna. "Efficient security and privacy mechanism for block chain application." Journal of Information Technology 1, no. 02 (2019): 58-67

[18] Mavroeidis, Vasileios, Kamer Vishi, Mateusz D. Zych, and Audun Jøsang. "The impact of quantum computing on present cryptography." arXiv preprint arXiv:1804.00200 (2018).

[19] Curcic, Tatjana, Mark E. Filipkowski, Almadena Chtchelkanova, Philip A. D'Ambrosio, Stuart A.

Wolf, Michael Foster, and Douglas Cochran. "Quantum networks: from quantum cryptography to quantum architecture." ACM SIGCOMM Computer Communication Review 34, no. 5 (2004): 3-8.

[20] Chen, Lily, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Vol. 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[21] Mavroeidis, Vasileios, Kamer Vishi, Mateusz D. Zych, and Audun Jøsang. "The impact of quantum computing on present cryptography." arXiv preprint arXiv:1804.00200 (2018).

[22] Chuang, Isaac L., and Yoshihisa Yamamoto. "Simple quantum computer." Physical Review A 52, no. 5 (1995): 3489.

[23] Barenco, Adriano. "Quantum physics and computers." Contemporary Physics 37, no. 5 (1996): 375-389.

[24] McClean, Jarrod, Nicholas Rubin, Kevin Sung, Ian David Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai et al. "OpenFermion: the electronic structure package for quantum computers." Quantum Science and Technology (2020).

[25] Gheorghiu, Alexandru, Theodoros Kapourniotis, and Elham Kashefi. "Verification of quantum computation: An overview of existing approaches." Theory of computing systems 63, no. 4 (2019): 715-808.

[26] Gidney, Craig, and Martin Ekerå. "How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits." arXiv preprint arXiv:1905.09749 (2019).

[27] Chen, Lily, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Vol. 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[28] https://spectrum.ieee.org/tech-talk/telecom/security/how-the-us-is-preparing-for-quantum-computings-threat-to-end-secrecy

[29] McClean, Jarrod, Nicholas Rubin, Kevin Sung, Ian David Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai et al. "OpenFermion: the electronic structure package for quantum computers." Quantum Science and Technology (2020).

[30] Duan, Lu-Ming, and Guang-Can Guo. "Reducing decoherence in quantum-computer memory with all quantum bits coupling to the same environment." Physical Review A 57, no. 2 (1998): 737.

[31] Duan, Lu-Ming, and Guang-Can Guo. "Preserving coherence in quantum computation by pairing quantum bits." Physical Review Letters 79, no. 10 (1997): 1953.

[32] Lamport, Leslie. Constructing digital signatures from a one-way function. Vol. 238. Technical Report CSL-98, SRI International, 1979.

[33] J. O'Gorman and E. T. Campbell, "Quantum computation with realistic magic-state factories," Physical Review A 95, 032338(1–19) (2017)

[34] V. Gheorghiu and M. Mosca, "Quantum cryptanalysis of symmetric, public-key and hash-based cryptographic schemes," arXiv preprint arXiv:1902.02332 (2019).

[35] Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." In Proceedings 35th annual symposium on foundations of computer science, pp. 124-134. Ieee, 1994.

[36] Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM review 41, no. 2 (1999): 303-332.

[37] Kumar, Manish. “quantum computing and post quantum cryptography”. In *International Journal of Innovative Research in Physics (IJIIP) Published by SMART Society,* https://ijiip.smartsociety.org/vol2_issue4.html Pp. 37-41. USA.

[38] Kumar, Manish, and Seba Susan. "Exploration and Implementation of RSA-KEM Algorithm." In *Soft Computing for Security Applications*, pp. 161-179. Springer, Singapore, 2022

8.0 List of publications

Two paper submitted and presented in these Conferences.

1. “International Conference on Soft Computing for Security Applications (ICSCS 2021)”

Paper Title: “Exploration and implementation of RSA-KEM algorithm”

Indexed in DBLP, EI Compendex, INSPEC, WTI Frankfurt eG, zbMATH, Japanese Science and Technology Agency (JST), SCImago.

Springer PROCEEDINGS International Conference on Soft Computing for Security Applications (ICSCS 2021)

2. “International Conference in Advanced Physics - IEMPHYS-21”

Paper Title: “Quantum computing and post quantum cryptography”

Publish in International Journal of Innovative Research in Physics (IJIIP) Published by SMART Society, USA (Print ISSN : 2689-484X, online ISSN: 2687-7902).

Springer ICSCS DHIRAJLAL GANDHI COLLEGE OF TECHNOLOGY

Certificate of Presentation

This is to certify that

Manish Kumar

has successfully presented a paper at the International Conference on Soft Computing for Security Applications (ICSCS 2021) organised by Dhirajlal Gandhi College of Technology, Salem, India during 10-11, June 2021.

Paper Title: Exploration and implementation of RSA-KEM algorithm

Author(s): Manish Kumar; Seba Susan

Session Chair	Organizing Secretary Dr. S. Rajendran	Conference Chair Dr. J. Parthasarathy

CERTIFICATE OF PARTICIPATION

IS PRESENTED TO

* * * * *

Manish Kumar

Department of Information Technology, Delhi Technological University Delhi, India

FOR THE SESSION

Oral session 3- Quantum Information Science

FOR THE PAPER TITLED

Quantum Computing and Post Quantum Cryptography

IN INTERNATIONAL CONFERENCE ON ADVANCED PHYSICS 2021 (IEMPHYS 2021) AT KOLKATA, INDIA ON 1ST - 3RD APRIL, 2021.

KOYEL GANGULY CONVENOR, IEMPHYS 2021	SOUMYADIPTA PAL CONVENOR, IEMPHYS 2021	TRIPARNA DATTA CONVENOR, IEMPHYS 2021