

EXPOSYS DATA LABS

SOFTWARE DEVELOPMENT PROJECT

-PRIYANGA P

ABSTRACT:

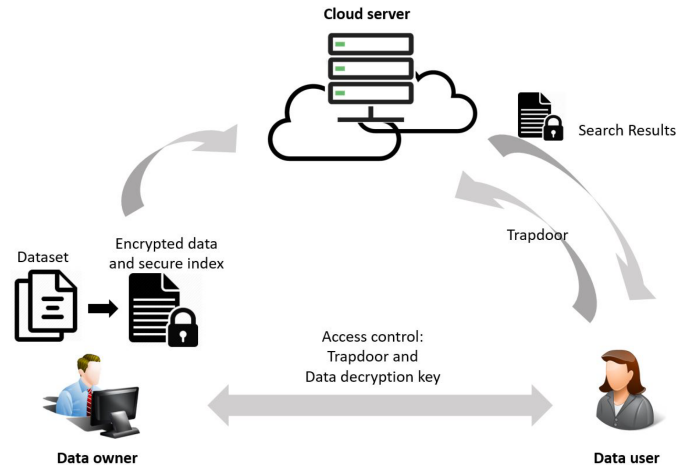
SMS is widely used by people all over the world to send messages and files to others. This transmission takes place on a cloud platform. Cloud Security and Cloud Storage was introduced with the rising need for huge data storage space within a secure environment. Cloud storage is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service. It's delivered on demand with just-in-time capacity and costs, and eliminates buying and managing your own data storage infrastructure. An encrypted message prevents any outside parties from reading the message, even if they are able to intercept it. Without the proper keys to unencrypt the message, no one except the sender and recipient can read the message's contents.

TABLE OF CONTENTS:

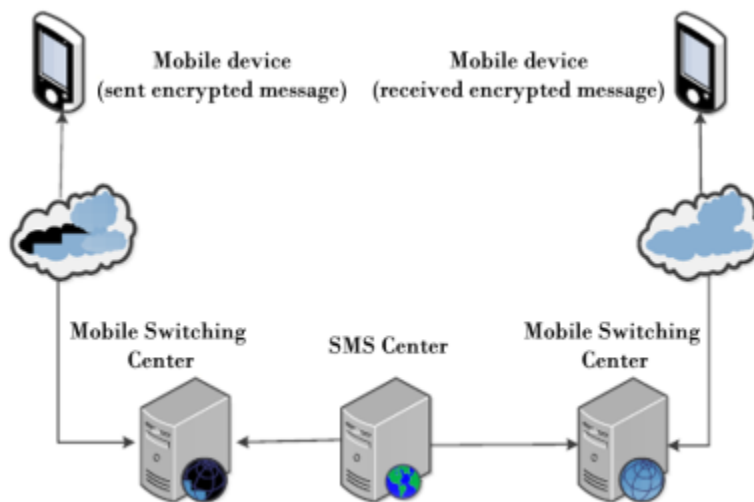
- ❖ Introduction
- ❖ RSA Algorithm
- ❖ Modified RSA Algorithm
- ❖ Methodology
- ❖ Implementation
- ❖ Conclusion
- ❖ References

INTRODUCTION:

With the advancements in technologies, the need for data storage and management in a secure, cost effective, and efficient way lead to the rise of cloud based storage resources such as AWS, Google Drive, Microsoft Azure etc. This also introduces the risk of privacy when messages are being transferred to a cloud platform. Considering any message or text file being sent from one person to another, there is a higher risk of it being accessed by a third party.



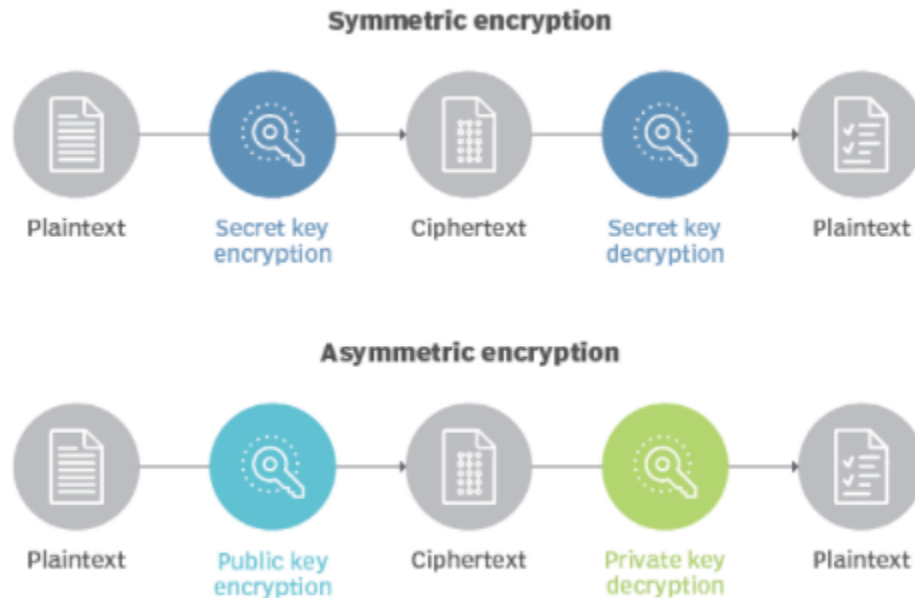
People share various private conversations through SMS. Architecture of SMS messaging depicted in the figure below shows clearly that the message from the sender doesn't reach the receiver directly but passes through the mobile switching center where message routing is performed and then stored in the SMS center where the message is directed for transmission process. This shows that the message is at high risk of getting accessed by any third party such as mobile operators or any hackers during transmission. Here arrives the vital role of message encryption.



Message encryption ensures that the sender and the intended receiver are the only people that can access the message's content with the help of encryption key which is required to decrypt the message. Various encryption algorithms such as AES, RSA, Diffie-Hellman, ElGamal Algorithms were developed to ensure cloud security. These techniques are used to create better data security in cloud storage and ensure secure communication. Encryption techniques convert message or plaintext into ciphertext, and decryption techniques extract the original message or plaintext into the same ciphertext.

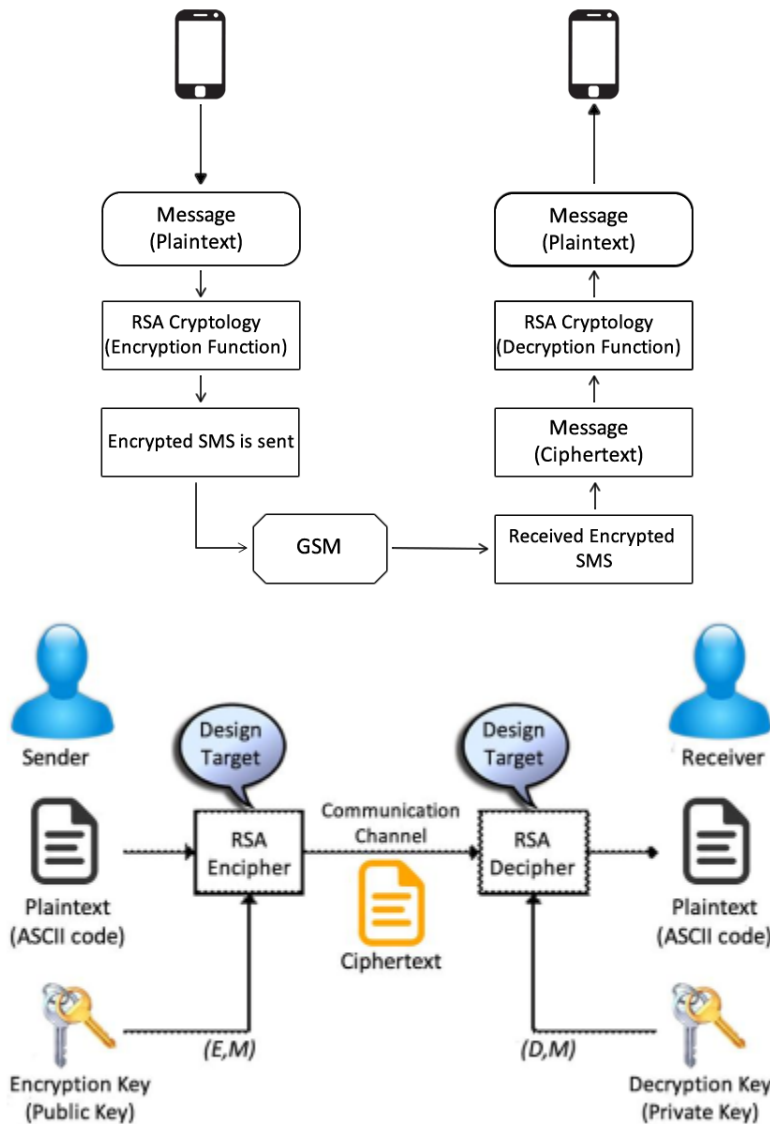
Encryption Techniques used in cloud security algorithms are broadly categorized into two symmetric key encryption and asymmetric key encryption. Symmetric key encryption uses a single key to encrypt and decrypt the data. In contrast, asymmetric key encryption uses two keys:

- Public key for encryption
- Private key for decryption.



RSA ALGORITHM:

RSA algorithm is an asymmetric cryptography algorithm. 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 the private key is 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. The figures below depicts SMS transmission using the RSA algorithm.



Key Generation Phase

Step 1: Choose two prime numbers, p and q , p not equal to q .

Step 2: Compute $n = p * q$

Step 3: Compute $\phi(n) = (p - 1) * (q - 1)$.

Step 4: Choose an integer e such that $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$.

Step 5: Calculate private key, d such that $e * d = 1 \mod \phi(n)$.

The public key is (n, e) and the private key (d, p, q) . Keep all the values d, p, q and $\phi(n)$ secret.

Encryption Phase

A sender sends a message to a receiver using n and e .

Original Message/Plaintext: P

Cipher text can be calculated as $C = P^e \bmod n$.

Decryption Phase

Receiver keeps $\phi(n)$ and d private. When the receiver receives the cipher text, private key is used to decrypt the message.

Plain text can be calculated as $P = C^d \bmod n$.

Significance of RSA Algorithm

Digital signatures serve the purpose of authentication and verification of documents and files. This is crucial to prevent tampering during official papers' transmission and prevent digital manipulation or forgery. They work on the public key cryptography architecture, barring one small caveat. Typically, the asymmetric key system uses a public key for encryption and a private key for decryption. However, when dealing with digital signatures, it's the opposite. The private key is used to encrypt the signature, and the public key is used to decrypt it. Since the keys work in tandem with each other, decrypting it with the public key signifies it used the correct private key to sign the document, hence authenticating the origin of the signature.

Limitations of RSA Algorithm

- The RSA Algorithm can completely fall apart if the private key is accessed by an unauthorized entity as the security of RSA is completely based on the private key.
- As the RSA modulus n is a large number, factoring it is a rather hard task. The RSA algorithm employs lengthy calculations for both encryption and decryption.
- RSA cryptography public key is used by the sender to encrypt the message. Thus only authenticated users can participate in the encryption procedure.

MODIFIED RSA ALGORITHM:

The RSA Algorithm is very slow due to the computational complexity of the numbers. The RSA encryption as well as decryption requires one modular exponentiation. If the RSA modulus n is a k -bit number, then, typically d is also a k -bit number. Decryption requires k squaring and $k/2$ multiplication modulo n . For example, if a 1024-bit number is the RSA modulus, there are 1024 squaring and 512 multiplications. Due to the large decryption exponent size, the RSA algorithm is very slow.

Improvising RSA algorithm can be done by the following:

Four Prime Numbers p , q , r and s used instead of 2 prime numbers in the original algorithm.

Where 'q' is a fixed Proth Number

'q' is a fixed Mersenne Prime Number and

's', 'r' are two Balanced Prime

Numbers such as,

n is the common modulus

e is the public key

d is the private key

The modified algorithm consists of only two stages: Encryption and Decryption. The key generation phase is not required in this improvised method. Key is generated only once and saved in a cloud based database. The indices are fetched from database instead of actual values and encryption/decryption is performed.

This modified 4-prime RSA Algorithm employs a cloud based database such as MongoDB. MongoDB is an open source database that uses a document-oriented data model and a non-structured query language. It is one of the most powerful NoSQL systems and databases around, today. It does not use the usual rows and columns that are so much associated with relational database management. It is an architecture that is built on collections and documents. The basic unit of data in this database consists of a set of key value pairs. It allows documents to have different fields and structures. The data model that MongoDB follows is a highly elastic one that can combine and store data of multivariate types without having to compromise on the powerful indexing options, data access, and validation rules.

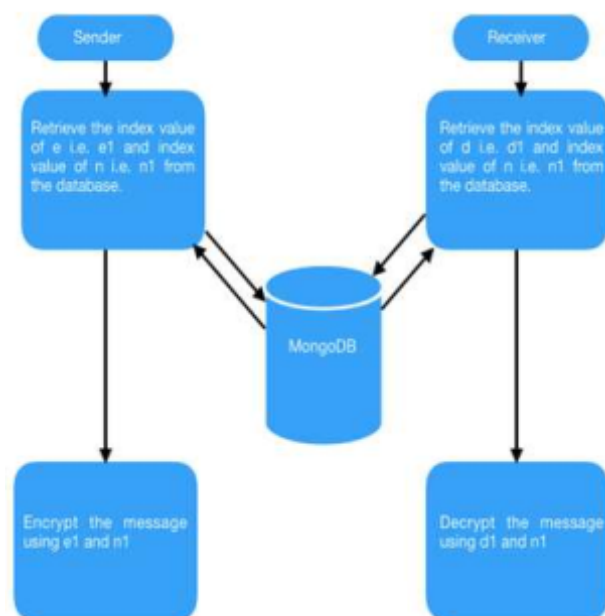


Figure: Architecture of 4-prime RSA algorithm

Proth Prime Number-

A Proth Prime number is a positive prime integer of the form

$$n = k * 2^n + 1$$

Mersenne Prime Number-

Mersenne Prime is a prime number that is one less than a power of two. In other words, any prime is Mersenne Prime if it is of the form $2^k - 1$ where k is an integer greater than or equal to 2.

Balanced Prime Number-

A Balanced Prime is a prime number with equal-sized prime gaps above and below it, so that it is equal to the arithmetic mean of the nearest primes above and below. Or to put it algebraically, given a prime number p_n , where n is its index in the ordered set of prime numbers,

$$p_n = \frac{p_{n-1} + p_{n+1}}{2}$$

METHODOLOGY:**Key Generation:**

Step 1: Choose a Proth prime number 'p', a Mersenne Prime Number 'q', two Balanced Primes 'r' and 's'.

Step 2: Compute $n = p * q * r * s$

Step 3: Compute $\phi(n) = (p-1) * (q-1) * (r-1) * (s-1)$.

Step 4: Choose an integer e such that $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$.

Step 5: Calculate private key, d such that $e * d = 1 \bmod \phi(n)$.

The public key is (n, e) and the private key (d, p, q) .

Keep all the values d, p, q and $\phi(n)$ secret.

These values are stored in a cloud based database such as MongoDB.

Encryption Phase:

The index of e and n are fetched from the database i.e. e_1 and n_1 are fetched from the database.

A sender sends a message to a receiver using n_1 and e_1 .

Original Message/Plaintext: P

Cipher text can be calculated as $C = P e_1 \bmod n_1$.

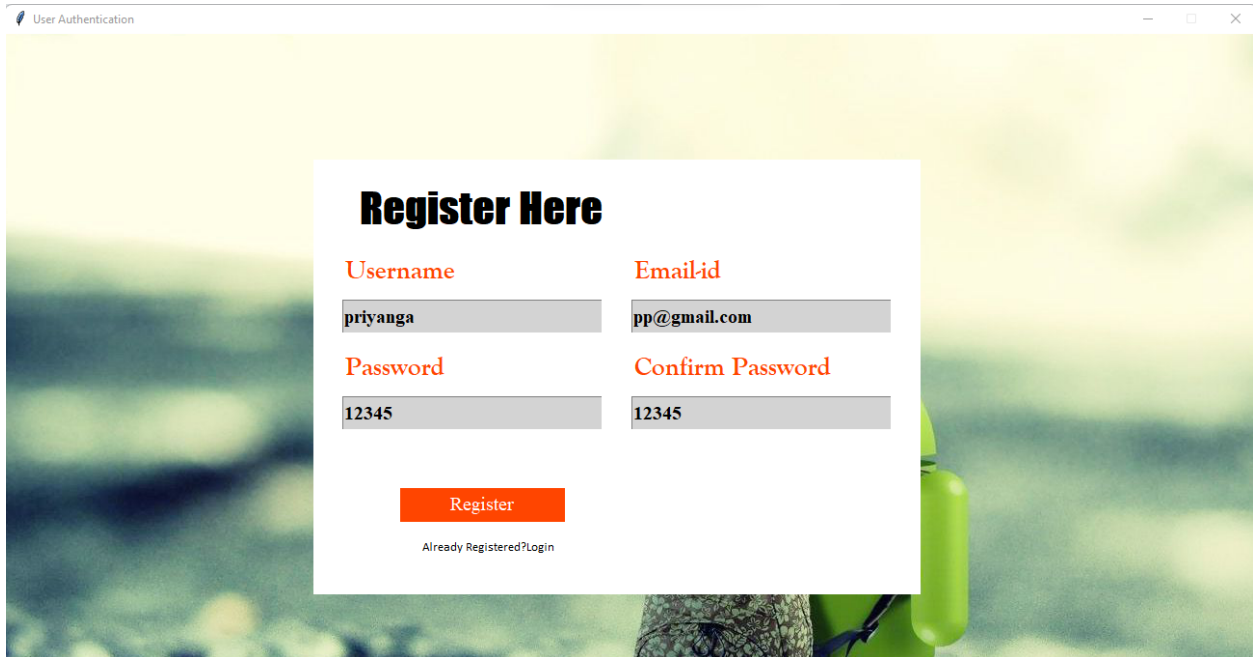
Decryption Phase:

Receiver keeps d private. When the receiver receives the cipher text, the index of private key and n i.e. d_1 and n_1 are fetched from the database and are used to decrypt the message.

Plain text can be calculated as $P = C d_1 \bmod n_1$.

IMPLEMENTATION:

User Authentication-

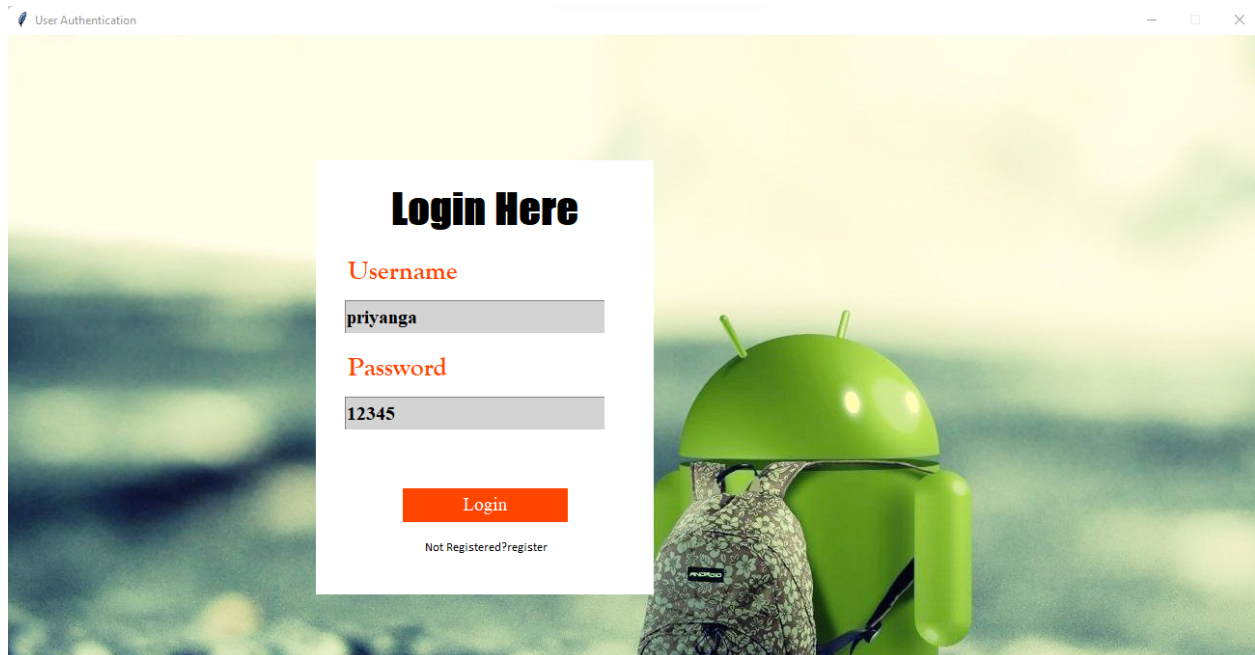


The screenshot shows a web browser window with the title "User Authentication". The background of the page is a blurred image of a green field. In the center, there is a white rectangular box containing the registration form. The form has the heading "Register Here" in bold black text. Below the heading, there are four labels in red text: "Username", "Email-id", "Password", and "Confirm Password". Each label is followed by a gray input box. The input boxes contain the following text: "priyanga", "pp@gmail.com", "12345", and "12345". Below the input boxes, there is a red button with the text "Register" in white. At the bottom of the form, there is a link that says "Already Registered?Login".

Username	Email-id
priyanga	pp@gmail.com
Password	Confirm Password
12345	12345

[Register](#)

[Already Registered?Login](#)



```
MySQL 8.0 Command Line Client
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

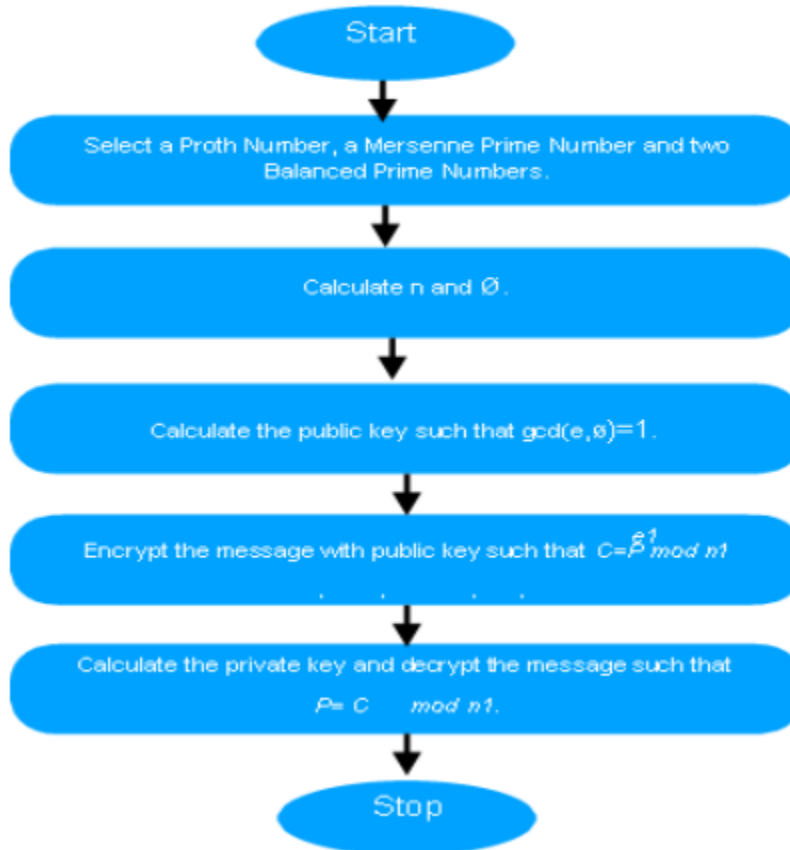
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use pythongui;
Database changed
mysql> show tables;
+-----+
| Tables_in_pythongui |
+-----+
| register             |
+-----+
1 row in set (0.43 sec)

mysql> select * from register
-> ;
Empty set (0.14 sec)

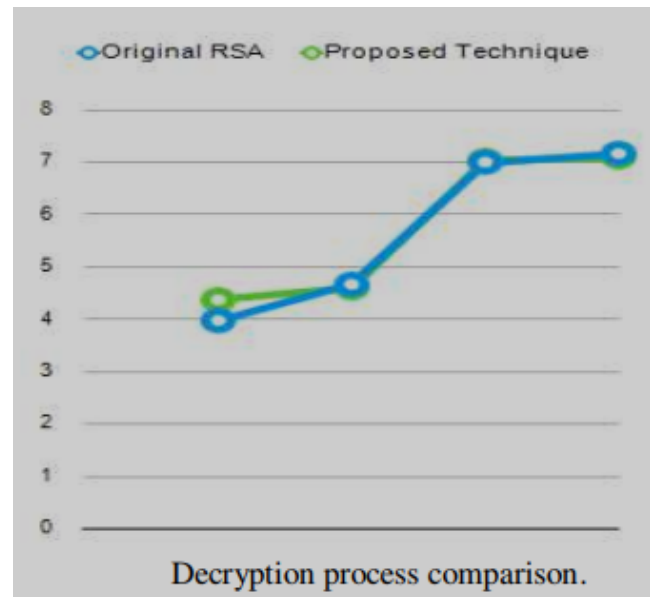
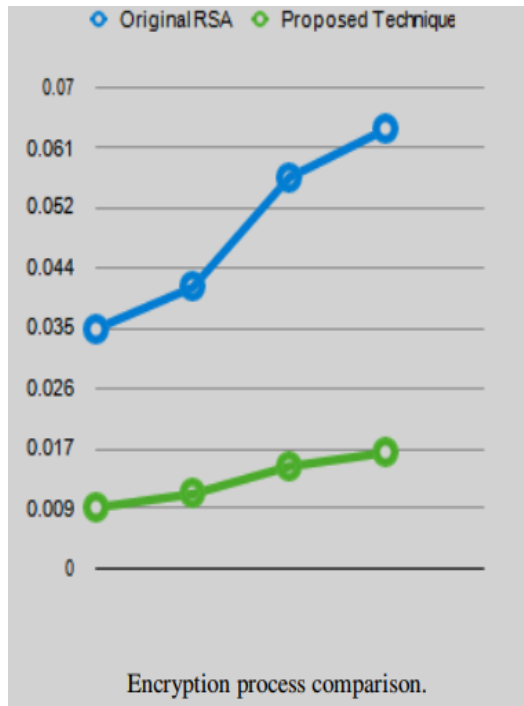
mysql> select * from register;
+-----+-----+-----+-----+
| username | emailid      | password | confirmpassword |
+-----+-----+-----+-----+
| priyanga | pp@gmail.com | 12345    | 12345           |
+-----+-----+-----+-----+
```

Modified RSA algorithm:



```
In [1]: runfile('C:/Users/BulBul/.spyder-py3/Exposys/enahanced_encrypt.py', wdir='C:/Users/BulBul/.spyder-py3/Exposys')

SMS Encryption and Decryption
Enter message : Hey, This is test message
Encrypting message...
p: 29440
q: 25673
x: 30097
y: 16487
Message: Hey, This is test message
e: 19417
d: 327684090457123945
N: 375041453101415680
enc: 268655641133458432 100266532559509381 33921759203226041 338405377441622784 314511186426047232
87308384001256704 14573096909342464 166014086253868585 222305091295010195 314511186426047232 166014086253868585
222305091295010195 314511186426047232 274469384101390336 100266532559509381 222305091295010195 274469384101390336
314511186426047232 137901701329120589 100266532559509381 222305091295010195 222305091295010195 61799690644457057
145591411463218983 100266532559509381
Message recieved after decryption is...
Hey, This is test message
```



Advantages:

- Faster Execution: All the key parameters which are used in our 4- prime RSA algorithm are stored before starting the algorithm.
- Ease of access: Users can access cloud databases from virtually anywhere, using a vendor's API or web interface.
- Scalability: Cloud databases can expand their storage capacities on run-time to accommodate changing needs.
- Disaster recovery: In the event of a natural disaster, equipment failure or power outage, data is kept secure through backups on remote servers.

CONCLUSION:

Various researches are still being done and many new improvised algorithms are developed to increase the efficiency and speed for the message Encryption. This method of using 4 different prime numbers instead of two prime numbers has been proven to increase the security of encryption and also seems to have increased the efficiency of the original RSA algorithm which was comparatively slow.

The modified RSA algorithm uses four prime numbers: a Proth Number, a Mersenne Prime and two balanced primes; instead of two prime numbers. Key parameters are calculated using these four prime numbers. All the calculated key parameters which are used in our 4-prime RSA algorithm are stored before starting the algorithm. Therefore, the speed of execution increases compared to the original RSA method as the decryption time is reduced considerably. Login and Register pages have also been created to validate and authenticate the user using python and tkinter.

REFERENCES:

<https://ieeexplore.ieee.org/document/6168406>
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3412776
<https://www.grin.com/document/511690>
<https://ieeexplore.ieee.org/document/7790289>