Personal and Ubiquitous Computing
https://doi.org/10.1007/s00779-021-01607-3

ORIGINAL ARTICLE

An optimised homomorphic CRT-RSA algorithm for secure and efficient communication

Rabia Abid[1] · Celestine Iwendi[2] · Abdul Rehman Javed[3] · Muhammad Rizwan[1] · Zunera Jalil[3] · Joseph Henry Anajemba[4] · Cresantus Biamba[5]

Received: 29 October 2020 / Accepted: 16 July 2021

Abstract

Secure and reliable exchange of information between devices is crucial for any network in the current digital world. This information is maintained on storage devices, routing devices, and communication over the cloud. Cryptographic techniques are used to ensure the secure transmission of data, ensuring the user's privacy by storing and transmitting data in a particular format. Using encryption, only the intended user possessing the key can access the information. During data or essential transmission, the channel should be secured by using robust encryption techniques. Homomorphic Encryption (HE) techniques have been used in the past for this purpose. However, one of the flaws of the conventional HE is seen either in its slow transmission or fast key decryption. Thus, this paper proposes an optimized Homomorphic Encryption Chinese Remainder Theorem with a Rivest-Shamir-Adleman (HE-CRT-RSA) algorithm to overcome this challenge. The proposed Technique, HE-CRT-RSA, utilizes multiple keys for efficient communication and security. In addition, the performance of the HE-CRT-RSA algorithm was evaluated in comparison with the classical RSA algorithm. The result of the proposed algorithm shows performance improvement with reduced decryption time. It is observed that the proposed HE-CRT-RSA is 3–4% faster than the classical Rivest-Shamir-Adleman (RSA). The result also suggests that HE-CRT-RSA effectively enhances security issues of the cloud and helps to decrease the involvement of intruders or any third party during communication or inside the data/server centers.

Keywords Homomorphic encryption · CRT · RSA · Security · Fast communication

1 Introduction

With the proliferation of data and various communication devices all around, the security of data both while at rest and in motion has gained significance [5, 6, 17, 19, 36]. Data has become the world's most valuable asset, and its use, either positive or negative, can make a significant impact on individuals, businesses, organizations, and governments. Data security and privacy is the primary pillar for communication, either this communication is between humans, machines or human to machine [20, 23, 31]. Cryptographic algorithms are used to maintain a high level of secrecy for communication between individuals, among groups, and organizations. Cryptography is a method that offers protection of information by giving access to a closed group with encryption/decryption keys [27, 28]. Researchers are seeking to find the most secure and complex cryptographic algorithms for clouds, networks, individuals [24]. In cryptography, transforming data or information from plain-text into some secret code (cipher-text) is known as encryption. This process is the most efficient and effective way of securing data.

With the advent of technologies in cloud computing, people can easily access and purchase cloud drive space to store their information and essential data. The cloud saves data in online database servers, which the cloud service providers manage, thus reducing the fear of data breaches. Cryptographic algorithms play a crucial role in the security of data or information [3], and thereby reduce or eliminate the fear of data breach. In other words, many optimized approaches are used by users for encryption to obtain security.

Security issues related to threat areas along the lines of external information storage, dependency on the "public"

✉ Cresantus Biamba
cresantus.biamba@hig.se

Extended author information available on the last page of the article.

Published online: 01 September 2021

internet, loss of control, multi-tenancy, and integration with inner security have been compared to conventional technology. It was noted that the cloud has many unique features, including its huge scale belonging to cloud carriers were distributed, heterogeneous, and virtualized [8]. Therefore, traditional security mechanisms consisting of identity, authentication, and authorization are insufficient for clouds in their present-day form.

The architecture of the cloud has challenges as well as successes. The cloud storage issues can be classified in personal or management: integrity, confidentiality, backups, data access, authenticity, and authority. The breach of security may be ethical in (Fig. 1) some cloud classification issues based on private or public [18]. Furthermore, cloud computing might also present exclusive dangers to a business enterprise than conventional IT answers. Unfortunately, integrating protection into those answers is frequently perceived as making them extra rigid.

Many devices perform well using RSA and the Chinese Remainder Theorem (CRT) when the fault part attacks the network. In most of the scenarios, the combination of the two algorithms is targeted. So, to prevent Fault Attacks, many generic techniques were implemented in the cloud security sector and proved to be a realistic approach. Construction of the combination of RSA and CRT cloud or data can be protected from attacks and shows a robust analysis [1]. Based on the latest optimized algorithm, different methodologies of computing the CRT RSA approach most securely exist. Hence, research shows an optimized approach which neither decreases execution time nor increases implementation time.

The author [35] provides that an effective technique for encryption named the HE technique helps to save cloud security from different fault threats. This scheme helps to encrypt and decrypt data without third-party disturbance. The experiment with numerous cryptographic algorithms on the cloud is used to keep the security and process of the encryption on data. It permitted complex numerical activities to be performed on encoded information without trading off the encryption. Generally, Homomorphic Encryption is relied upon to significantly impact distributed computing, permitting organizations to store encrypted information in an open cloud and exploit the cloud supplier's logical administrations.

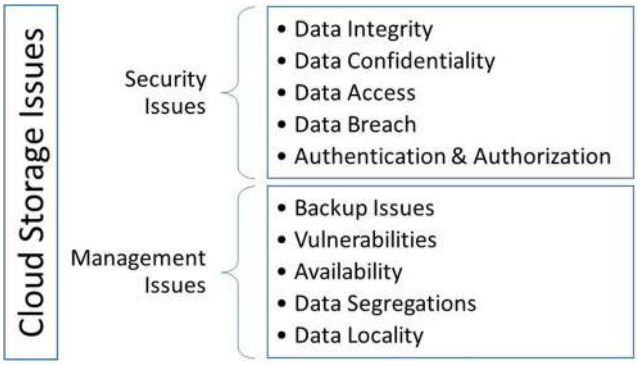


Fig. 1 Storage issues in cloud computing

The performance of the HE technique resolves the issues of cloud security, confidentiality, and privacy of data [7] and is a type of encryption procedure that shows the result as plain text after the operation or computation. HE is of two types (i.e., (1) Partial Homomorphic Encryption (PHE) and (2) Fully Homomorphic Encryption (FHE)). The latter is considered the most secured and efficient during the computation of third-party intervention. Based on the research done in past years, identification of the problem and future directions are discussed [39]. The RSA algorithm's efficient working with the key lengths of 1024 and 2048 bit was experimented to get the result of a faster speed of RSA encryption and slow decryption. Considering the security requirements of RSA with a key length of 2048 was selected as the working model of the hybrid cipher scheme, and the effectiveness and efficiency of the optimized cipher algorithm technique were tested. Therefore, by experimenting several times on the practical online interface, the encryption of data connected with the decryption of data [13].

Encryption is one of the best ways to guarantee the security of delicate data. It gives the instrument in data privacy, yet also worked with computerized signature, confirmation, mystery sub-keeping, framework security. Accordingly, the motivation behind adjusting the encryption method guarantees the data's classification, trustworthiness, and assurance, keeping data from altering, fraud, and falsifying. The two primary kinds of cryptographic calculations are a Symmetric Key cryptography and Asymmetric Key cryptography and a few models in each sort. The RSA algorithm already has a vast experimental aura as two integer values, less speedy or less secure than a single algorithm. When we combine two different techniques, the selected theorem and algorithm make the best combination and multiple key lengths and input values. Combining the Chinese Remainder Theorem and RSA technique (CRT-RSA algorithm) is currently actualized with homomorphic encryption/decryption calculation for better and greater security. Another contribution is the discussion of Homomorphic encryption/decryption with the proposed CRT-RSA to take the cloud and data security to the next level. This calculation depends on keeping up harmony among the security and speed of a calculation.

This proposed research implemented a homomorphic encryption technique along with CRT-RSA, where this method helps to multiply Cipher-Text (as value) and get the result when we decrypt it. The research's main objective is a detailed overview of Homomorphic

CRT-RSA techniques (HE-CRT-RSA). Proposing RSA with the Chinese Remainder Theorem using multiple keys and large input values in Matlab Framework helps optimize the cryptographic algorithm. After the experiment and result discussion, the HE-CRT-RSA approach helps to overcome the cloud storage issues effectively. It helps to decrease intruders' involvement or any outside third party during the communication or in data/server centers [37].

1.1 Problem statement

There are a plethora of cryptographic techniques and algorithms practically implemented to secure wireless communication in the classical channel. According to a recent survey in 2019, RSA algorithm is the strongest of all. However, it is likely to experience prolonged data sending and low communication factors where it is robust in security. This research aims at discussing the most successful data compression techniques along with the Homomorphic RSA algorithm to make cryptographic calculations more secure compared to the previous one.

1.2 Contributions

- Propose a novel approach named *HE-CRT-RSA* that optimizes the CRT-RSA algorithm by using multiple keys to make the transmission secure.
- Provide a comparison between the classical RSA algorithm, single Key CRT-RSA with *HE-CRT-RSA*.
- The results suggest that the *HE-CRT-RSA* endeavor to redesign the execution of the RSA cryptosystem through a system that improves the speed on the RSA encryption side by using the CRT and upgrade the security of the data by using two key factors rather than a single critical factor in comparison with existing techniques.

1.3 Organization

This research study is based on an advanced cryptography algorithm to enhance security in networking or wireless communication. The paper consists of different sections. The detailed background of cryptography is discussed in Section 2, literature review is discussed in Section 3 and a detailed overview of the RSA algorithm is discussed where the pros and cons will be discussed in Section 4. Section 5 provides the optimised RSA algorithm. After that, Theorem CRT is discussed in detail in Section 6. The proposed solution where detailed discussion on advanced RSA algorithm is discussed with practical implementation, results, and analysis in Section 7. Lastly, the research will be concluding with future findings and scope in Section 8.

2 Background

Cryptography is a way that permits people to defend data and communications with the assistance of codes. Cryptocurrencies, which include Bitcoin and Ethereum are interchangeable digital approach uses cryptography to check asset transfer, manage the introduction of extra units, and secure transactions. Cryptography brings a similar level of protection to the cloud by protecting information saved with encryption. It is worthy to note that cryptography can protect sensitive cloud information without delaying the transmission of data. Various organizations discuss cryptographic protocols for cloud computing to preserve stability, safety, and efficiency. Physical manipulation over cloud databases is impossible. The most effective way to stabilize a bit is to defend it with cryptography using cryptographic keys. There are diverse forms of cryptographic keys needed for cloud safety.

2.1 Symmetric key cryptography

Symmetric cryptography is the algorithm that utilizes the way into the encryption procedure and is equivalent to the key for the decoding procedure. The symmetric cryptography algorithm is separated into two classes:

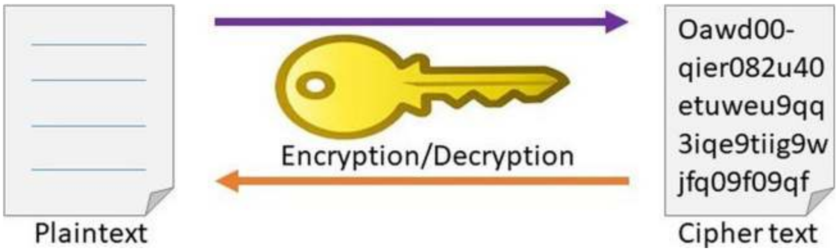
- Flow algorithm (Stream Ciphers)
- Block algorithms (Block Ciphers)

The upside of this symmetric cryptography algorithm is the procedure time for encryption and decryption that is moderately quick because of the effect of the key generator. Since the procedure is generally quick, this calculation is reasonable for advanced correspondence framework progressively like GSM. Nevertheless, sending messages to various clients takes an unexpected key in comparison to the past one. So the number of catches will be legitimately corresponding to the number of clients. The flow for the symmetric key generation's working is shown in Fig. 2.

Different points of interest of utilizing symmetric key calculations are higher working velocity when contrasted and deviated key calculation albeit legitimately corresponding with the expansion of document size, for example, speed of encryption and decoding process relies upon record size; the greater the document size, the additional time required for encryption and decryption. Be that as it may, there is likewise the weakness of this symmetric key calculation: for every conveyance of messages with various client required various keys, so there will be challenges in key administration [15]. A few calculations utilizing symmetric key calculation:

1. Data Encryption Standard (DES) is a mainstream square figure calculation since it fills in as a standard

Fig. 2 Symmetric key cryptography



balance encryption key calculation, even though the new AES calculation has now supplanted it since DES is hazardous. DES has a place with a cryptographic arrangement of balance and has a place with a square figure type. DES works on a 64-piece square size. DES portrays 64 bits of plain text to 64 bits of cipher-text by utilizing 56 bits of inner key (interior key) or sub-key. The inner key is produced from an outer key 64-piece length.

2. RC6 is a cryptographic calculation that is remembered for the asymmetric key. This calculation is the improvement of the RC5 calculation remembered for Advanced Encryption Standard (AES). RC6 utilizes the center of the data-dependent pivot. RC6 is a calculation that utilizes obstruct to 128 bits, with key sizes utilized changing between 128, 192, and 256 bits. RC6 isolates the 128-piece hinder into four 32-piece squares and keeps the six essential working principles: number, subtraction, selective OR, whole number increase, sliding bits to one side, and bowing bits to one side—the fundamental procedure right now the key planning and unscrambling of encryption.

2.2 Asymmetric key cryptography

In the mid-1970s Whitfield Diffie and Martin Hellman developed the Asymmetric algorithm. This encryption method upset the universe of cryptography. An Asymmetric key is a couple of cryptographic keys utilized for the encryption procedure and the other for unscrambling. Any individual who gets an open key can utilize it to encode messages, while just a single individual has a specific mystery private key to dismantle the secret phrase sent to him Fig. 3:

However, this Asymmetric Encryption system has a shortcoming; for example, the encryption procedure is much slower than the symmetric key. Along these lines, for the most part, the message is not encoded with an unbalanced key, yet just symmetric keys are encrypted with a lopsided key. At the point when a message is sent after it is encoded with a symmetric key.

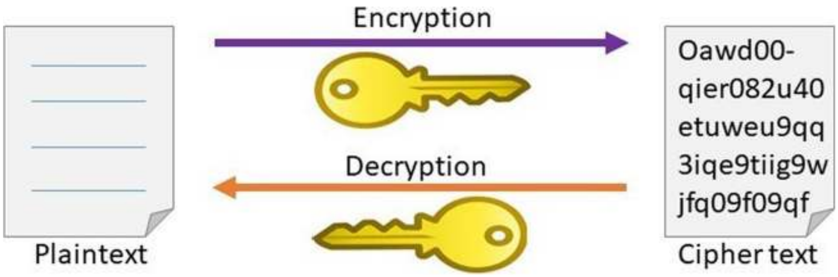
Rivest-Shamir-Adleman (RSA) calculation is the primary calculation appropriate for computerized marks and encryption and one of the most developed in the field of open key cryptography since first depicted in 1977. The RSA calculation includes a few stages, for example, key age, encryption, and decoding.

At present, the most popular and most broadly utilized open key framework is RSA, which was first proposed in the paper "A technique for getting computerized marks and open key cryptosystems" by RL Rivest et al. in 1978. An unbalanced (open key) cryptosystem depends on the number hypothesis, a square figure framework. Its security depends on the trouble of the vast number of prime factorization, which is a significant numerical issue that has no viable arrangement [32]. RSA open key cryptosystem is one of the most average ways to utilize open key cryptography in encryption and computerized signature guidelines.

3 Related work

Cloud computing is an articulation used to depict a set of enlisting minds that include innumerable associated through a non-stop correspondence framework, for instance, the Internet [33]. It could be tough to get information protection in the cloud because it travels through the Internet. Cloud safety and implementation are the primary trouble at some

Fig. 3 Asymmetric key cryptography



point in implementation. Because of this advantage, each organization that desires to use these facilities anticipating the ability issuer has to guard their information against unauthorized users. To keep the cloud infers stable, the meds (calculations) and limit (databases advocated with the aid of using the Cloud issuer) [9].

According to [22] cryptography means to make cloud data safe and secure. Cryptography is based on mathematical calculations and algorithms to make the data storing platform safe and secure. It can be thought of as the security toolkit of the cloud, servers, data, and communication. Authors in [26] indicate the strategies which might be beneficial for real-time encryption. All encryption techniques have been established to have their successes and failures as well as features established to be suitable for special applications. The assessment among Symmetric and Asymmetric algorithms indicates that Symmetric algorithms are quicker than their Asymmetric counterparts. Through the primary research and the result of the assessment, authors in [40] discover that the maximum dependable set of rules is AES in the period of velocity encryption, decoding, complexity, the duration of the key, shape, and flexibility.

The author [25] effectively implemented asymmetric key from the RSA algorithm. The results show that the RSA algorithm is efficiently proven to resist the false attempts to penetrate the information despatched via the community and tampering with this information. The extra safety is primarily based totally on the factorization of the big prime integers selected to generate the key [16]. To save an attacker's discovery of the cipher-text, it is best to choose a huge prime number to range within the RSA algorithm. According to [38], the encrypted plain-text with the aid of using the RSA algorithm is regularly utilized in combination with different encryption schemes to switch the information securely. To make RSA more efficient, a plain-text will typically be encrypted first with an asymmetric set of rules, and another time using the RSA algorithm to encode or encrypt the cipher-text the usage of the asymmetric key. The last use of the symmetric key is within the scope decryption of the cipher-text and to better the authentic plain text.

According to [14], the research has three kinds of encryption that were mixed to make the most the benefits of everyone to construct a secure system. According to [4, 34], AES is used to encrypt dispatched information, exploiting its excessive encryption pace and its low RAM necessities. RSA is used to guard the encryption key against getting stolen by producing keys (personal and public). MAC is used to secure the encrypted key or information. The key and encrypted information are dispatched to the receiver and get decrypted through personal use. The typical encryption run is easy and speedy with low computational demands and gives high device secrecy.

Authors in [1] introduced changed RSA technique based on multi keys and Chinese remainder theorem (CRT) with RSA algorithm as uneven key encryption technique. The motivation of this Technique is to offer stable transmission of records among any networks. Network protection was the focus and designed to offer the utilization, safety, and integrity of the network and records [29]. The goal of their paper was to enhance the overall performance of RSA.

Authors in [21] proposed an RSA algorithm using more than one public key pairs with Homomorphic Encryption (HE). The concept is to generate a key pair from more than one key through RSA HE, which is in part HE in nature instead of an unpaired key pair. This approach utilizes one key for enciphering and a different one for deciphering. This scheme's splendor is that a single key pair is chosen from more than one key pair communicating with different parties. According to [12], the approach of more than one keys era utilizes a few mathematical logic for acquiring public keys directly, as opposed to the RSA scheme with a single key. By so doing, the attacks for locating the non-public key are minimized.

In the context of a Ciphertext Policy Attribute-based Code Encryption (CP-ABE), the statistics proprietors are allowed to add their statistics into cloud forms and open them with customers owning different credentials or characteristics, and for that reason, have a beneficial approach with a view to assure statistics safety in a cloud servers scenario [30]. However, a CP-ABE scheme cipher-text calls for a comfortable shape of access, which might also leak any records of the privileged recipients or expose the cipher-text underlying message. In this work, a homomorphic, primarily based CP-ABE version is carried out to enhance the cloud statistics' study safety. According to [11], the RSA algorithm inclined to factorize attacks is dependent on the worth "n", which showed an open key and could be breakable.

The authors in [2] proposed that e_{th} roots modulo n with non-insignificant likelihood can be utilized to factor n. Currently, the attack is the primary shortcoming of the RSA algorithm dependent on the "e" and "n", which are the open keys utilized for encryption. The examination means to adjust an RSA algorithm dependent on the key age of new estimation of the open key to used or expanded the protection from factorization attack [10].

4 RSA algorithm

In order to produce multiple keys open and private keys in RSA calculation, it is important to check the calculation time of different keys; however, the security is extra contrasted with the standard calculation RSA. By utilizing two open keys and two private keys in optimized or

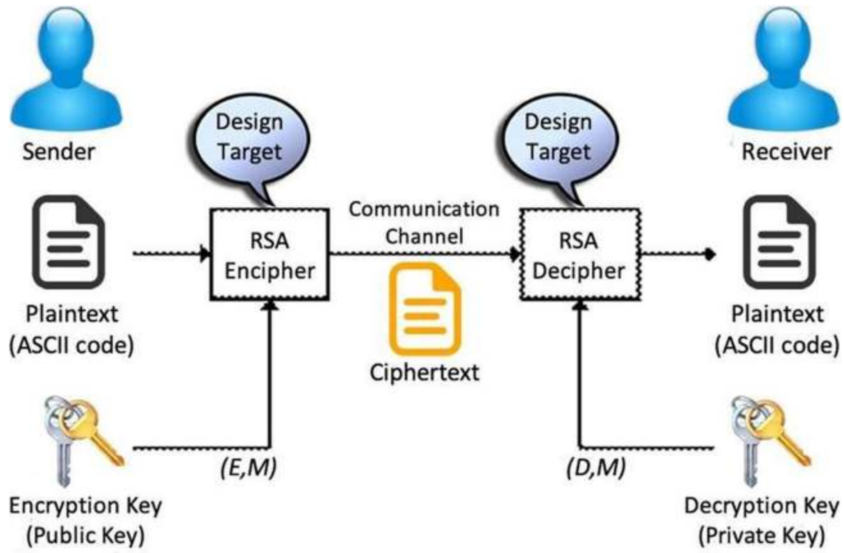


Fig. 4 Structure of RSA algorithm

enhanced RSA calculation, which will utilize four prime numbers and get open/private keys, likewise, utilizing two open keys for encoding and two private keys for unscrambling. In Fig. 4, the workflow or structure of the RSA algorithm can be seen. It is more defensive against attack. There are three stages: Generation of Key, encoding, and decoding.

4.1 Structure of RSA algorithm

In the wake of creating open and private keys, data that must be sent is scrambled with the open key. Encryption and decryption forms are done as follows:

- The figure content C is found by the condition C'=M^e and n', where M is the first message.
- The message M can be found from the figure content C by the condition M' = C^d mod n'.
- A content encoded with the open key must be settled with the private key.

Algorithm 1 The structure of RSA algorithm as follows.

```
Input Values: p and q
Compute:
    n = p x q
   (n) = (p-1) (q-1)
Select Integer values: e [(gcd (), e) - 1; 1 < e < $\phi$ (n)]
Compute: d de $mod\ \phi$ (n) = 1
    C = Cg 1 $mod$ (z)
Encryption: M < n C = M ($mod$ n)
Decryption: CM = C($mod$ n)
```

If the two prime numbers are taken simultaneously, it upgrades security. However, it requires the execution of Exponentiation by squaring calculation and square and duplicate calculation for viable encryption and unscrambling. For effortlessness, the program is planned with moderately little prime numbers.

5 Optimised RSA algorithm

A plan is being introduced where the data is packed utilizing compression algorithms and afterward scrambled by utilizing RSA calculation. The encoded message is then sent employing the Internet to the beneficiary. After accepting the message, it must be unscrambled first. At that point, the unscrambled message must be decompressed to get the first data. Then again, RSA works gradually when the size of its bit increments afterward 1024 bits, so as to improve the speed on RSA decryption side utilizing CRT (Chinese Remainder Theorem) by which the plan is semantically protected. The main idea of the proposed research upgrades the exhibition by utilizing Chinese Remainder Theorem and expand the security by utilizing two open keys in the decrypting. By proposing a CRT algorithm with RSA algorithm, a combination will help make fast communication and compress large data. The proposed model for the Optimized RSA algorithm is in Fig. 5.

The proposed framework is endeavoring to redesign the execution of the RSA cryptosystem through a system that has to improve the speed on the RSA encryption side by using the Chinese Remainder Theorem (CRT) and upgrade the security of the data by using two key factors rather than a single key factor. Using the sporadic whole number at whatever point encoded a comparable message more than one time it will look different. The general idea of this strategy is to advance the execution of the figuring and make it progressively secure and decrease the unraveling

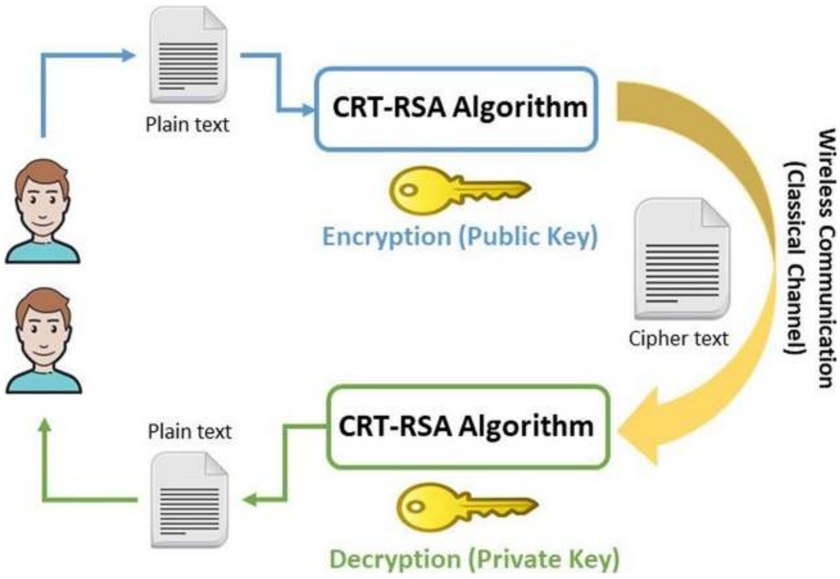


Fig. 5 Proposed optimized RSA algorithm model

time simultaneously. By using four prime numbers, and two figure content compositions for each message, the testing of estimation become increasingly unpredictable. RSA is a square figure where the plain-content and figure content are the whole numbers some place in the scope of 0 and n-1. Some steps for the encryption/decryption process are as follows:

Algorithm 2 Steps for encryption/decryption are as follows.

```
prime numbers: a, b, c, and d
def n = ab, and z = cd        {Calculate Estimation (n, z)}
$\phi(n)$ = (a-1) (b-1); $\phi(z)$ = (c-1) (d-1)        {Calculate}
while 1 < p < n and 1 < g < z do
    gcd (p, $\phi(n)$) = 1        {Encryption}
    def gcd(n, z)        {Factor of n and z}
    gcd (p, $\phi$ (z)) = 1
end while
for d: (xd = 1 mod ($\phi(n)$)) do
    for T: (tg = 1 mod ($\phi(z)$)) do
        xa = x mod (a-1); xb = x mod (b-1)    {Calculate private key}
        xc = x mod (c-1), xd = x mod (s-1)
        Return
    end for
end for
Open key KU = (p,n),(g,z)    {Cipher-text to Plain-text}
private key VK = t,z, xa, xb, xc, xd
Return
```

In steps for encryption and decryption, a, b, c, and d are prime numbers. For factorization, n and z integer values, whereas x denote for the cipher-text. During all steps of prime numbers are generated p and q and then calculate the modulus n. similarly, e for encryption of plain-text and d for the decryption of the cipher-text.

In RSA, with the aid of using the multi keys' method used prime numbers to generate more than one public keys and personal key, which this method offers more excellent protection compared with RSA set of rules and RSA CRT. In RSA, using the multi keys method, we used public and personal keys. This makes an individual more secure because he is not constantly attacked or robbed using unauthorized parties and enhancing protection and performance in information sharing over the network. However, much less pace evaluates to RSA set of rules and CRT-RSA. Since the usual RSA used high numbers to generate one public key and one non-public key to encrypt and decrypt, this makes it much less steady, which it is miles without problems decomposed. RSA takes more time with the aid of using multi keys to encrypt and decrypt than CRT-RSA using the use CRT in the decryption of RSA set of rules it calls for much less processing time and a smaller quantity of reminiscence for the very last decoded result as compared with RSA with the aid of using multi keys.

5.1 Procedure of encryption

Under the encryption process, messages are encrypted with a code referred to as a public key, shared openly. Due to the RSA algorithm's few awesome mathematical residences, as soon as a message has been encrypted with the public keys. To encode the message, M steps are as per the following:

Algorithm 3 To encrypt the Cipher-text C1 as follows.

```
Ensure: Encryption of Cipher-text C1        {Initialization}
if M[0-n1] then
    public key: (p) C1 = Mp mod (n) )
    Open key: (g) C = Cg 1 mod (z)
    Return
end if
```

After generating more than one public and personal key inside the procedure of key generation, now encrypted the message with the general public keys. Thus, the encryption procedure made times, the reliability is has become extra compared to the same old RSA algorithm. We will take the message (M) and the primary public key (e), then make the procedure of encryption and discover C1 = Me mod (n). Using C1 and the second one, the public key (g) could be determined the cipher textual content in the encryption procedure: C = Cg1 mod (z).

5.2 Procedure of decryption

Algorithm 4 To decrypt the Cipher-text C2 as follows.

```
Input: $Input value$ : $C2$, $private keys(t, z)$
Output: Output value: m1 = Ct mod (z). return message
if $Compute Ca = C1 mod a, Cb = C1 mod b$ then
    Cc = C1 mod c, Cd = C1 mod d
    Then: ma = Caixa mod a, MB = Cbxb mod b,
    mc = Ccxc mod c, MD = Cdxd mod d
    Return output
end if
Return error
Combine Ma, Mb, mc, and MD, we get unique plain-text messages.
```

In the decryption system, the authentic message is decrypted with the aid of using non-public keys d, t. The cipher-text could be decrypted with the primary non-public key (d) with this formula: m1 = C t mod (z), then can get the authentic message with second non-public key (t) with this formula: M = MD 1 mod (n).

6 Multiple keys in CRT-RSA

CRT-RSA provides better security when compared to the standard RSA. This calculation utilized four prime numbers to create numerous open keys and many private keys.

In this CRT-RSA algorithm, two open and private keys, which makes it more secure since it will not be attacked or looted by unapproved individuals and enhancing safety and efficiency in information distribution over the system, however low- speed contrasts with RSA calculation. Meanwhile, the classical RSA utilized two prime numbers to create one open and private key to do encryption. Decryption provides a little less security, which effectively deteriorates. Here, in Fig. 6, the flow chart for optimizing the RSA algorithm is shown.

In CRT-RSA, applying the multi-key method and utilizing two open and private keys makes it more secure since it cannot be attacked or burgled by unapproved individuals and refining security and efficiency information distribution over the system. However, low-speed contrasts with RSA calculation and CRT-RSA. Hence, the classical RSA utilized two prime numbers to produce one open key and one private key to encode and decode, provide little less security, which effectively deteriorates. RSA, by using multi keys, set aside some effort to encode and decode much more than classical RSA by utilizing CRT to decode RSA calculation, consumes or takes less handling time, and littler measure of memory for final decoded result contrasted and RSA by multi keys.

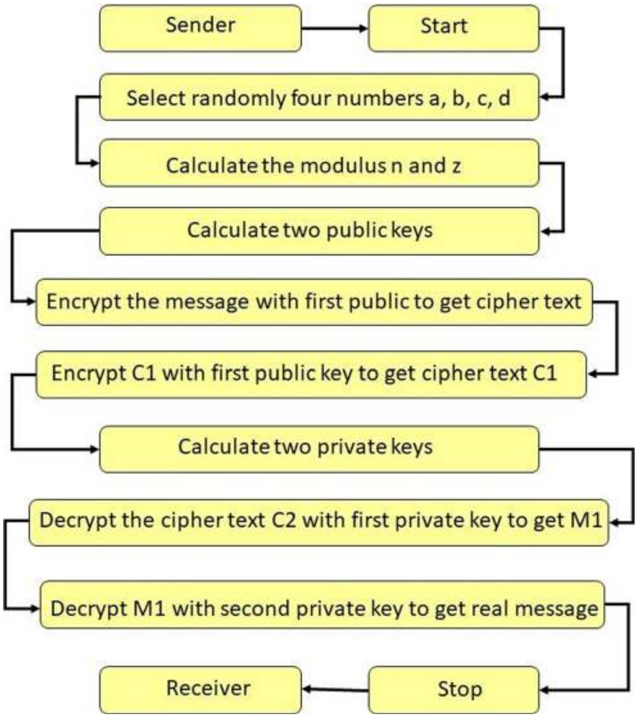

Sender
Start
Select randomly four numbers a, b, c, d
Calculate the modulus n and z
Calculate two public keys
Encrypt the message with first public to get cipher text
Encrypt C1 with first public key to get cipher text C1
Calculate two private keys
Decrypt the cipher text C2 with first private key to get M1
Decrypt M1 with second private key to get real message
Receiver
Stop

Fig. 6 Proposed optimized RSA algorithm

The following is the step-by-step procedure for the working of the proposed CRT-RSA algorithm in the form of Pseudocode by using Python language: Select input values a, b, c, d (by using RSA built-in library), then encrypt the key, a, b, c, d prime numbers generated, then decrypt the key (by using CRT-RSA), And then finally print the output.

6.1 Optimised homomorphic CRT-RSA algorithm

Our proposed HE-CRT-RSA algorithm is currently actualized with homomorphic encryption/decryption calculation for better and greater security. This calculation depends on keeping up harmony among the security and speed of a calculation. Currently, stacks are taken, such as the first stack holds the data about Sequence Counter (CS), and the other stack keeps up the record for the information on which encryption must be performed. The decoding process is the opposite of the encryption process. The correct Sequence Counters (CS) are recognized from the succession Counter table for the decrypting strategy.

6.1.1 Encryption process CRT-RSA using homomorphic technique

Fundamental favorable position of the altered calculation is no key common over the system while keeping up the information scrambled over the system. Absolute security of the information regardless of which organization used to get to the information. No information respectability was lost.

The process of creating key and encryption of text is almost similar to the process of classical RSA except that the procedure for the generation of private key is in the form of i.e., a, b, mod n and m for message.

Algorithm 5 Process to encrypt the text as follows.

```
input values: a, b, da, db, and m for message
Output: M (message)
if d = da mod a - 1 and d = db mod b - 1 then
    d - 1 = (da - 1) mod (a - 1);          {to find solution}
    d - 1 = (db - 1) mod (b - 1);          {d is integer value}
    M = (Ma b (b -1 mod a)                 {CRT Algorithm}
      + Mb a (a -1 mod b)) mod (a x b)
    return M
end if
```

Whereas: By including the element of homomorphic encryption, speeds up, security, and throughput. The work flow of the Proposed Homomorphic RSA encryption Procedure shown in Fig. 7.

6.1.2 Decryption process of CRT-RSA using homomorphic encryption

Decryption process is actually the converse of encryption process. The best possible sequence counters (CS) are distinguished from the SC table for decrypting methodology. The work flow of the Proposed Homomorphic RSA decryption Procedure is shown in Fig. 8.

Information cannot be decrypted by any procedural technique for no data will be made accessible about the key. This adjusted CRT-RSA calculation gives greater security as a contrast with RSA calculation by performing mathematical procedures on encoded information.

7 Result and discussion

For the practical implementation of HE-CRT-RSA, an online python interface was used for the test and experiments. In this proposed research, Homomorphic encryption techniques was used along with CRT-RSA, where this method helps to multiply Cipher-Text (as value) and get the result when decrypted as shown in Fig. 9.

In this experiment, a CRT-RSA was used to multiply two value, and get the result of by decrypting them. For encryption: $C = M\ (\mathrm{mod}\ N)$ $C = M\ (\mathrm{mod}\ N)$. Find decrypting key and compute: $M = C\ (\mathrm{mod}\ N)$ $M = C\ (\mathrm{mod}\ N)$. If two values are selected then a cipher-text can be formed as: $C1 = a\ (\mathrm{mod}\ N)$ $C1= a\ (\mathrm{mod}\ N)$ and $C2 = b\ (\mathrm{mod}\ N)$ $C2 = b\ (\mathrm{mod}\ N)$. When the Cipher-text is multiplied, the result becomes $C3 = C1\times\ C2 = a\ (\mathrm{mod}\ N)\times\ b\ (\mathrm{mod}\ N) = a\times\ b\ (\mathrm{mod}\ N)$

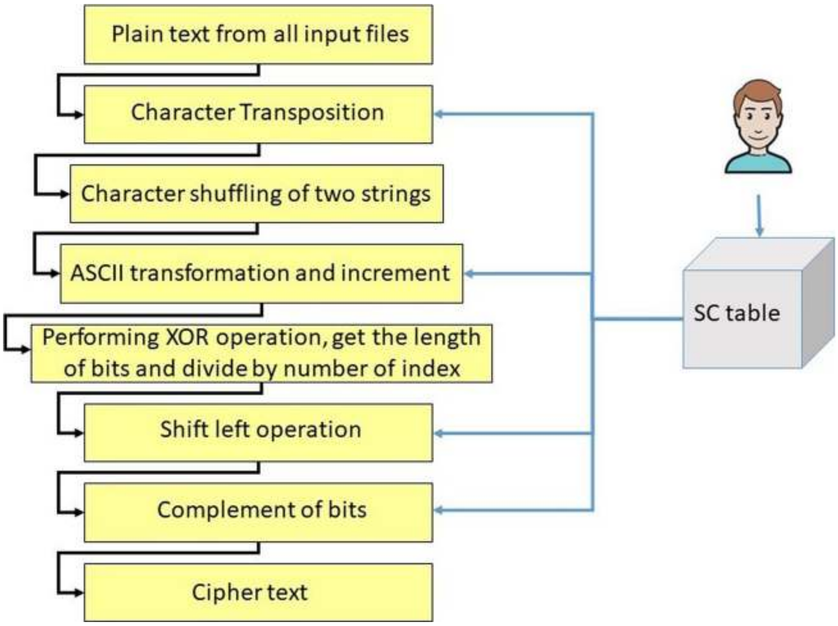

Plain text from all input files
Character Transposition
Character shuffling of two strings
ASCII transformation and increment
Performing XOR operation, get the length of bits and divide by number of index
Shift left operation
Complement of bits
Cipher text
SC table

Fig. 7 Proposed homomorphic RSA encryption procedure

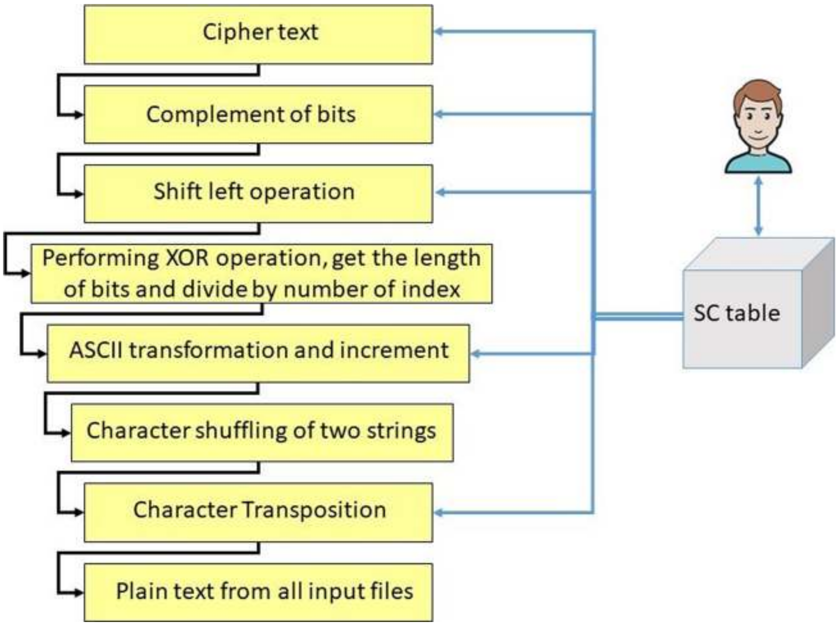
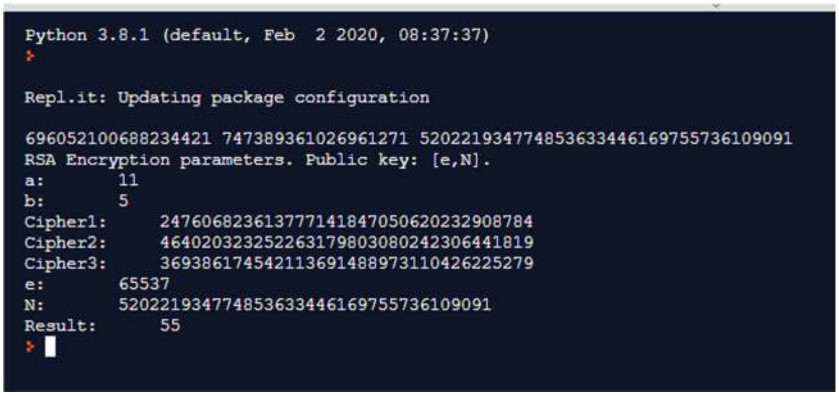

Cipher text
Complement of bits
Shift left operation
Performing XOR operation, get the length of bits and divide by number of index
ASCII transformation and increment
Character shuffling of two strings
Character Transposition
Plain text from all input files
SC table

Fig. 8 Proposed homomorphic RSA decryption procedure



```
Python 3.8.1 (default, Feb  2 2020, 08:37:37)
>

Repl.it: Updating package configuration

696052100688234421 747389361026961271 520221934774853633446169755736109091
RSA Encryption parameters. Public key: [e,N].
a:       11
b:       5
Cipher1:      247606823613777141847050620232908784
Cipher2:      464020323252263179803080242306441819
Cipher3:      369386174542113691488973110426225279
e:       65537
N:       520221934774853633446169755736109091
Result:       55
>
```

Fig. 9 Result of HE-CRT-RSA algorithm by using small integers of prime numbers

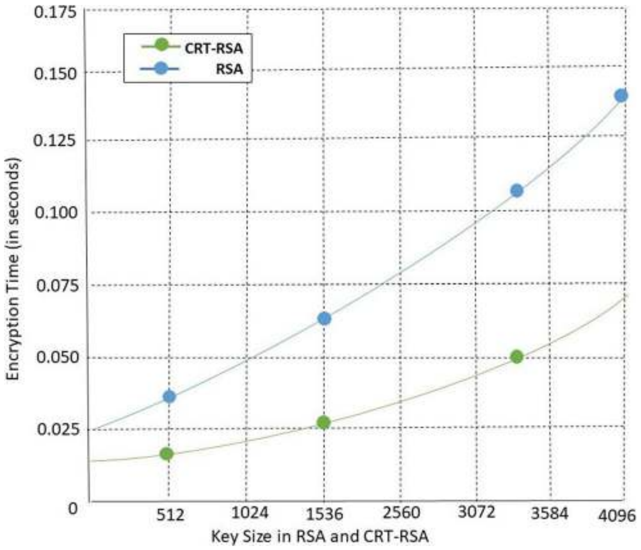


Fig. 10 Encryption time in RSA and HE-CRT-RSA

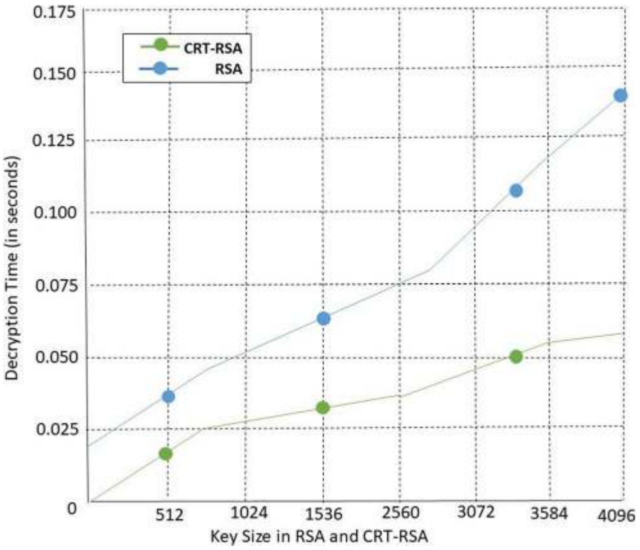


Fig. 11 Decryption time in RSA and HE-CRT-RSA

Table 1 CRT-RSA using multiple keys

Input size	Encryption time (ms)	Decryption time (ms)
512	0.019	0.020
1536	0.028	0.030
3345	0.050	0.050
4096	0.075	0.060

and $C3 = C1 \times C2 = a \pmod N \times b \pmod N = a \times b \pmod N$. Then $C3 = (ab) \pmod N$ $C3 = (ab) \pmod N$.

7.1 Comparison between existing RSA and proposed homomorphic CRT-RSA

Execution of RSA-CRT decoding is done from java-based code in numerous past looks. A few experiments are dependent on various situations, including diverse message lengths and distinctive key sizes for message decoding. These experiments are utilized to test the presentation and security elements of CRT-RSA.

Encryption time is a period that a calculation takes to encode the plain content. As key length builds, encryption time is taken by both calculation increments continuously. Figure 10 uncovers that RSA-CRT encryption time is very nearly multiple times quicker when contrasted with RSA encryption time.

Decrypting time is a period that a calculation takes to decrypt the Cipher-Text content. As key length builds, decrypting time taken by both calculation increments continuously. Figure 11 uncovers that CRT-RSA unscrambling time is very nearly multiple times quicker when contrasted with RSA decoding time.

Table 2 Comparison between classical RSA, CRT-RSA (with two keys) and HE-CRT-RSA (with multiple keys)

RSA	CRT-RSA	HE-CRT-RSA
Used single public key	Used single public key	Used multiple public key
Used single private key	Used single private key	Used multiple private key
Speed is slow	Speed is moderate	Speed is fast
Have less security	Have less security	Have strong security
Strongly affected from brutal attacks	Moderately affected from brutal attacks	Very less affected from brutal attacks
Required more time for encryption and decryption	Required less time for encryption and decryption	Required more but with time for encryption and decryption more accuracy rate

Table 3 Comparative analysis of different security algorithms with HE-CRT-RSA technique

Algorithms	Memory size (bytes)	Average encryption (bytes)	Average encoding (bytes)
AES	14.7	3.84024	256
DES	18.2	2.9477	27
3DES	20.7	2.9477	40
BlowFish	9.38	3.93891	128
HE-CRT-RSA	31.5	3.0958	44

Table 1 demonstrates the four unique sizes of documents and relating decoding execution time taken by straightforward RSA calculation and CRT-RSA utilizing Homomorphic encryption in KB/Seconds in a moment or two. By breaking down Table 1, we reason that the decoding time taken by CRT-RSA utilizing Homomorphic encryption is little in contrast with RSA. The decrypting time is taken by primary RSA and CRT-RSA utilizing Homomorphic encryption and four distinctive information records. By examining Table 1, we notice that the throughput time (millisecond) taken by CRT-RSA utilizing Homomorphic encryption is enormous compared to RSA. The Table 2 show the relation and comparison between the classical RSA algorithm, single Key CRT-RSA, and our proposed HE-CRT-RSA approach.

The proposed algorithm and techniques save the cloud from various malicious attacks and the third party. In Table 3, the comparison of different security algorithms with classical RSA technique. Comparison is based on the system memory, Bytes of encryption, and optimal Byte for encoding through the survey.

Table 4 shows the comparative analysis of classical RSA and proposed HE-CRT-RSA. After the practical implementation of both algorithms the decryption time in bytes along with key length shown. The results show the efficiency and fast application of the key length 1024, 2048, 3072 and 4096.

Table 4 Comparison of classical RSA and proposed HE-CRT-RSA

Key length	Time by simple RSA (ms)	Time by HE-CRT-RSA (ms)
1024	0.42	0.412
2048	4.54	3.64
3072	20.56	19.42
4096	37.35	33.36

The proposed method is progressively secure when contrasted with classical RSA calculation and CRT-RSA. Also, it improved the working, the calculation in unscrambling because it utilized the CRT in decoding, in this manner the proposed strategy quicker than RSA by multi keys. It diminishes the expense of calculation. Even though it requires some investment to perform it when contrasted with unique RSA.

8 Conclusion

A combination of the Chinese Remainder Theorem (CRT) algorithm has been proposed with the RSA algorithm to make fast communication and compress large data. The proposed framework is endeavoring to redesign the execution of the RSA cryptosystem that improves the speed on the RSA encryption side by using the CRT and upgrade the security of the data by using two key factors rather than a single key factor. In this proposed research, the CRT-RSA cryptographic algorithm is used to make fast, efficient, and secure wireless communication between two parties. Along with this, the proposed CRT-RSA algorithm is trying to merge with the homomorphic Technique for better and secure communication. This research objective includes a detailed overview of the most successful classical cryptographic Technique, where RSA is the best of all. The research proposed an RSA with the Chinese Remainder Theorem using multiple keys and large input values in a Matlab framework that helps optimize the cryptographic algorithm. Another contribution is the discussion of using homomorphic encryption/decryption with proposed CRT-RSA to take the security to the next level. Future work will discuss the practical implementation of the Advanced Homomorphic CRT-RSA algorithm and cloud issues and security threats.

Funding Open access funding provided by University of Gävle.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Abdeldaym RS, Abd Elkader HM, Hussein R (2019) Modified RSA algorithm using two public key and chinese remainder theorem. IJ Electron Inf Eng 10(1):51–64
2. Aggarwal D, Maurer U (2016) Breaking rsa generically is equivalent to factoring. IEEE Trans Inf Theory 62(11):6251–6259
3. Ahmad SA, Garko AB (2020) Hybrid cryptography algorithms in cloud computing: A review. In: 2019 15th international conference on electronics computer and computation (ICECCO), pp 1–6
4. B J S, Kumar VKRR, Nair A (2017) Comparative study on aes and rsa algorithm for medical images. In: 2017 international conference on communication and signal processing (ICCSP), pp 0501–0504
5. Basit A, Zafar M, Liu X, Javed AR, Jalil Z, Kifayat K (2020) A comprehensive survey of ai-enabled phishing attacks detection techniques. Telecommun Syst pp 1–16
6. Bhattacharya S, Kaluri R, Singh S, Alazab M, Tariq U et al (2020) A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu. Electronics 9(2):219
7. Biksham V, Vasumathi D (2017) Homomorphic encryption techniques for securing data in cloud computing: A survey. Int J Comput Appl 975:8887
8. Ch R, Srivastava G, Gadekallu TR, Maddikunta PKR, Bhattacharya S (2020) Security and privacy of uav data using blockchain technology. J Inf Secur Appl 55:102670
9. D Arivazhagan RK (2020) Wdevelop cloud security in cryptography techniques using des-3l algorithm method in cloud computing. Int J Sci Technol Res 9(1):252–255
10. Das D (2018) Secure cloud computing algorithm using homomorphic encryption and multi-party computation. In: 2018 international conference on information networking (ICOIN). IEEE, pp 391–396
11. Dhote C (2016) Homomorphic encryption for security of cloud data. Procedia Comput Sci 79:175–181
12. Ezzati KEMABHA (2019) Multiprime cloud-rsa: a fast homomorphic encryption scheme for data confidentiality protection in clouds. Int J Intell Enterp 6(2/3/4):217–229
13. Feng R, Wang Z, Li Z, Ma H, Chen R, Pu Z, Chen Z, Zeng X (2020) A hybrid cryptography scheme for nilm data security. Electronics 9(7):1128
14. Harba ESI (2017) Secure data encryption through a combination of AES, RSA and HMAC. Eng Technol Appl Sci Res 7(4):1781–1785
15. Hercigonja Z (2016) Comparative analysis of cryptographic algorithms. Int J Digit Technol Econ 1(2):127–134
16. Intila C, Gerardo B, Medina R (2019) A study of public key 'e' in rsa algorithm. In: The international conference on information technology and digital applications, IOP conf. series: materials science and engineering, vol 482, p 012016
17. Iwendi C, Jalil Z, Javed AR, Reddy T, Kaluri R, Srivastava G, Jo O (2020) Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks. IEEE Access 8:72650–72660
18. Jan SU, Ghani D, A Alshdadi A, Daud A et al (2020) Issues and challenges in cloud storage architecture: a survey. Alshdadi, Abdulrahman and Daud, Ali Issues and Challenges in Cloud Storage Architecture: A Survey (June 10, 2020)
19. Javed AR, Beg MO, Asim M, Baker T, Al-Bayatti AH (2020a) Alphalogger: Detecting motion-based side-channel attack using smartphone keystrokes. Ambient Intell Humaniz Comput pp 1–14
20. Javed AR, Usman M, Rehman SU, Khan MU, Haghighi MS (2020b) Anomaly detection in automated vehicles using

multistage attention-based convolutional neural network. IEEE Trans Intell Transp Syst

21. Khan SA, Aggarwal R, Kulkarni S (2019) Enhanced homomorphic encryption scheme with pso for encryption of cloud data. In: 2019 5th international conference on advanced computing & communication systems (ICACCS). IEEE, pp 395-400
22. Mathur P, Gupta AK, Vashishtha P (2019) Comparative study of cryptography for cloud computing for data security. Recent Adv Comput Sci Commun 12:1–00
23. Mittal M, Iwendi C, Khan S, Rehman Javed A (2020) Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using levenberg-marquardt neural network and gated recurrent unit for intrusion detection system. Trans Emerg Telecommun Technol pp e3997
24. Obaid TS (2020a) Study a public key in RSA algorithm. Eur J Eng Res Sci 5(4):395–398
25. Obaid TS (2020b) Study a public key in RSA algorithm. Eur J Eng Res Sci 5(4):395–398
26. Omar G, Abood SKG (2018) A survey on cryptography algorithms, vol 8. IJSRP
27. Parameshachari B, Kiran RP, Rashmi P, Supriya M, Rajashekarappa PandurangaH (2019) Controlled partial image encryption based on lsic and chaotic map. In: ICCSP, pp 60–63
28. Parameshachari B, Panduranga H, liberata Ullo S et al (2020) Analysis and computation of encryption technique to enhance security of medical images. In: IOP conference series: materials science and engineering, vol 925. IOP Publishing, p 012028
29. R Wardoyo ES, Sari AK (2018) Symmetric key distribution model using rsa-crt method. 2018 third international conference on informatics and computing (ICIC) pp 1–9
30. Ramya KR, Josephine BM, Sai JN, Chandra NJ, Basha SS (2020) An improved homomorphic based encryption and decryption process on cloud texual data. J Crit Rev 7(7):609–615
31. Rehman Javed A, Jalil Z, Atif Moqurrab S, Abbas S, Liu X (2020) Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles. Trans Emerg Telecommun Technol pp e4088
32. Rivest R, Robshaw M, Sidney R, Yin Y (2016) The rc6 block cipher v1 1, August 20, 1998
33. RM SP, Maddikunta PKR, Parimala M, Koppu S, Reddy T, Chowdhary CL, Alazab M (2020) An effective feature engineering for dnn using hybrid pca-gwo for intrusion detection in iomt architecture. Comput Commun
34. Santhosh Kumar BJAN, K RRV (2017) Hybridization of rsa and aes algorithms for authentication and confidentiality of medical images. 2017 international conference on communication and signal processing (ICCSP)
35. Sendhil R, Amuthan A (2020) A descriptive study on homomorphic encryption schemes for enhancing security in fog computing. In: 2020 international conference on smart electronics and communication (ICOSEC). IEEE, pp 738-743
36. Shabbir M, Shabbir A, Iwendi C, Javed AR, Rizwan M, Herencsar N, Lin JCW (2021) Enhancing security of health information using modular encryption standard in mobile cloud computing. IEEE Access 9:8820–8834
37. Subramani P, Rajendran GB, Sengupta J, Pérez de Prado R, Divakarachari PB (2020) A block bi-diagonalization-based precoding for indoor multiple-input-multiple-output-visible light communication system. Energies 13(13):3466
38. Tarigan SY, Ginting DS, Gaol ML, Sitompul KL (2017) The combination of rsa and block chiper algorithms to maintain message authentication. In: Journal of physics: conference series, vol 930. IOP Publishing, p 012009
39. Tebaa M, El Hajji S, El Ghazi A (2012) Homomorphic encryption method applied to cloud computing. In: 2012 national days of network security and systems. IEEE, pp 86–89
40. Vyakaranal S, Kengond S (2018) Performance analysis of symmetric key cryptographic algorithms. In: 2018 international conference on communication and signal processing (ICCSP). IEEE, pp 0411-0415

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Rabia Abid[1] · **Celestine Iwendi**[2] · **Abdul Rehman Javed**[3] · **Muhammad Rizwan**[1] · **Zunera Jalil**[3] · **Joseph Henry Anajemba**[4] · **Cresantus Biamba**[5]

Rabia Abid
rabiaba576@gmail.com

Celestine Iwendi
celestine.iwendi@ieee.org

Abdul Rehman Javed
abdulrehman.cs@au.edu.pk

Muhammad Rizwan
Muhammad.rizwan@kinnaird.edu.pk

Zunera Jalil
zunera.jalil@mail.au.edu.pk

Joseph Henry Anajemba
herinopallazo@ieee.org

[1] Department of Computer Science Kinnaird College for Women, University Lahore, Lahore, Pakistan

[2] School of Creative Technologies, University of Bolton, A676 Deane Rd, Bolton BL3 5AB, UK

[3] Department of Cyber Security, Air University, Islamabad, Pakistan

[4] Department of Communication Engineering, Hohai University, 211100, Changzhou, China

[5] Department of Educational Sciences Faculty of Education and Business Studies, University of Gävle, Gävle, Sweden