Secure File Storage in Cloud Computing using Modified Cryptography Algorithm

Kumar Sparsh – 19BCB0025
Siddharth Garg – 18BCB0038
Simbothula Varun Kumar – 19BCI0050
Gowthami A. T. – 19BCI0250

A report submitted for the J component of

CSE1011- CRYPTOGRAPHY FUNDAMENTALS

Supervisor: Dr. D. RUBY

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
Vellore institute of Technology, Vellore

May 30, 2021

Declaration

This report has been prepared on the basis of our own work. Wherever other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 10200 words

Students' Name: Siddharth Garg, Kumar Sparsh, Simbothula Varun Kumar, Gowthami A. T.

Date of Submission: 28-May-2021

Signature:

Table of Contents

• Abstract

Our project investigates the security issues identified with the file cloud storage. By doing a thorough literature review, we found out about the current ways to deal with securing the cloud file storage framework, its applications and their disadvantages. Identifying the research gap, we noticed that even though RSA algorithms were used in the literature papers, the security can still be increased and it is not that much fast. To ensure the security of clients' information in cloud information server, we have proposed a modified RSA algorithm with multiple keys and CRT to ensure confidentiality of data coupled with hashing through SHA-512 to maintain integrity.

In this project, we have made a secure data exchange app where files are encrypted using RSA-CRT algorithm and hashed later. On successful implementation of our project, we observed that the proposed technique is more secure as compared to original RSA algorithm and RSA-CRT. And it enhanced the algorithm performance for decryption because it employed the CRT for decryption, thus the proposed technique proved to be faster than RSA with multi keys.

• Introduction

The algorithm of RSA is an asymmetric cryptography technique, this is working on two keys i.e., public key and private key. The proposed model in our project takes four prime numbers for modified RSA. Instead of sending one public key directly, send two public keys to the receiver. But there is problem of the speed, so that in RSA decryption used Chinese remainder theorem to enhancement the speed of RSA decryption.

• Problem Statement

In Cloud Storage, we share data among many clients, server and people. Thus, the ***security of information present in cloud is not guaranteed*** since it is easy for an intruder to access and demolish the first type of information. So, there is requirement of some plainly key which help us to do cross breed encryption and protect the data. The algorithm of RSA is an asymmetric cryptography technique, this is working on two keys i.e., public key and private key. The propos--ed model takes four prime numbers in RSA. Instead of sending one publ-ic key directly, send two public keys to the receiver. But there is proble-m of the speed, so that in RSA decryption used Chinese remainder theor -em to enhancement the speed of RSA decryption.

• Objective

a. **Ensure Confidentiality of data on Cloud**

Since Cloud today is accessed by a large number and variety of users, confidentiality is constantly under threat. Cloud computing is capable of handling intense computing tasks and mass data storage, so traditional security mechanisms is not enough for cloud due to heavy communication or computation overhead.

Method keeping the shortcoming of RSA in mind would help achieving this objective using RSA-CRT.

b. Ensure Availability of data on Cloud

The project targets at deploying method which can encrypt and decrypt fast. A faster encryption and decryption methodology would ensure a better user experience since the rate of processing data (i.e., availability of services) would be high.

CRT when combined with the RSA algorithm ensures it.

Using CAPTCHA in the project also increases availability by reducing DOS attack.

c. Ensure Integrity of data on Cloud

Encryption and decryption process which ensures that data are not modified by the 3rd party. It will be achieved by using ***hash function*** in the project using SHA-512.

d. Ensure proper authorization of data and services on Cloud

Cloud is a virtual data store where 3rd party (Cloud employees and managers) are trusted with user's data. It is vulnerable to internal attacks (e.g. irritated employees). Hence our project would ***ensure proper login into the platform*** using RSA-CRT and hybrid hash function.

A cryptographic hash function would take a message of arbitrary length and creates a message digest of fixed length, which is unique for each message.

Using MD5, we save password so no one can see it, even in database.

• Literature Survey

i. Yang, K., & Jia, X. (2012). An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE transactions on parallel and distributed systems, 24(9), 1717-1726

In this paper, it proposed a proficient and intrinsically secure dynamic reviewing convention. It secures the information protection against the

reviewer by consolidating the cryptography strategy with the bi-linearity property of bi-linear paring, as opposed to utilizing the veil method. In this manner, their multi-cloud clump reviewing convention doesn't require any extra coordinator. Their cluster reviewing convention can likewise bolster the clump examining for numerous proprietors. Moreover, their evaluating conspire brings about less correspondence cost and less calculation cost of the evaluator by moving the registering loads of evaluating from the inspector to the worker, which enormously improves the evaluating execution and can be applied to enormous scope distributed storage frameworks.

ii. Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017). Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, 387, 103-115

This paper concentrated on the issue of the cloud information stockpiling and planned to give a methodology that could stay away from the cloud administrators arriving at client' delicate information. Tending to this objective, they proposed a novel methodology entitled as Security-Aware Effective Distributed Storage (SA-EDS) model. In this model, they utilized their proposed calculations, including Alternative Data Distribution (AD2), Secure Efficient Data Distributions (SED2) and Efficient Data Conflation (EDCon) calculations. Their exploratory assessments had demonstrated that their proposed plan could viably shield significant dangers from cloud-side. The calculation time was shorter than current dynamic methodologies. Future work would address making sure about information duplications so as to increment the degree of information accessibility since any of datacentre's down will cause the disappointment of information recoveries.

iii. Bindu, B. S., & Yadaiah, B. (2011). Secure data storage in cloud computing. International Journal of Research in Computer Science, 1(1), 63-73.

In this paper, they contemplated the issue of information security in cloud servers. To ensure the accuracy of clients' information in cloud information server, they proposed a viable and adaptable plan with unequivocal unique information support, including square change, delete, and join. They use erasure-correcting code in the record dissemination planning to give repetition

equality vectors and assurance the information reliability. Their plan achieves the joining of capacity rightness protection and information defilement has been recognized during the capacity accuracy check over the circulated workers. Their plan is exceptionally productive and tough to Byzantine disappointment, noxious information alteration assault, and even worker intriguing assaults. They accept that information stockpiling security in Cloud Computing, a zone loaded with difficulties and of prevailing essentials, is still in its early stages to be distinguished. They imagine a few potential bearings for future examination on this territory. It permits Third Parity Auditor to review the cloud information stockpiling without requesting clients' time, likelihood.

iv. Sookhak, M. (2015). Dynamic remote data auditing for securing big data storage in cloud computing

This paper examines the issue of extra handling time in the current information evaluating techniques and discovers extension to build up a far-off information reviewing strategy that can be utilized to check the uprightness of the redistributed information in distributed computing. This plan has the ability to safely bolster dynamic information update procedure on the square level, for example, embed, erase, adjust, and add activities. To distinguish the holes and remarkable issues in the zone of information stockpiling trustworthiness of distributed computing, it proposed a topical scientific classification based on the best in class information evaluating techniques to meet the necessities. Subjective examination is utilized to analyse the current strategies and feature the points of interest and impediments of them and open issues and difficulties of information inspecting plans in cloud and portable distributed computing condition that have not been tended to yet were distinguished and featured. The current information inspecting approaches were actualized in the genuine distributed computing condition, and the benchmark test was utilized to assess such strategies dependent on the calculation and correspondence cost on the customer and worker side. In addition, the effect of dynamic information update tasks was breaking down on the current information approaches in the genuine condition. It examined the impact of dynamic information update procedure for the enormous scope document size. At last, the effect of regular information refreshes was assessed for various size of the documents. Another far off information evaluating

strategy was proposed based on mathematical mark procedure to satisfy the target of productive answer for checking the honesty of the redistributed information in distributed computing. The proposed conspire addresses the issue of extra calculation and correspondence cost for cloud information stockpiling framework. The D&CT information structure likewise enables their strategy to be pertinent for huge scope information with least preparing time on the customer. The proposed information examining plan is executed in the genuine condition by utilizing java and C++ language to address the target of assessing DRDA technique. The presentation of the DRDA plot was approved by utilizing the benchmark test in the copying condition and broke down the DRDA conspire by utilizing unmistakable boundaries, for example, length of mark, document size, and likelihood of identification. The various situations likewise characterized to assess the proposed strategy. Moreover, it broke down the quality of the security based on mathematic to approve and verification the security of the DRDA technique. The outcomes indicated that the D&CT information structure lessens the handling season of dynamic information update activities by diminishing the quantity of moving and furthermore exhibited that the D&CT information structure significantly decline the preparing season of dynamic information update for enormous scope re-appropriated record in distributed computing.

v. ***Shimbre, N., & Deshpande, P. (2015, February). Enhancing distributed data storage security for cloud computing using TPA and AES algorithm. IEEE.***

This paper, talks about the record circulation and SHA-1 strategy. At the point when document is conveyed then information is likewise isolated into numerous workers. So here the need of information security emerges. Each square of record contains its own hash code, utilizing hash code which will improve client verification process; just approved individual can get to the information. Here, the information is encoded utilizing propelled encryption standard, so information is effectively and safely put away on cloud. Outsider reviewer is utilized for open inspecting. This paper talks about the treatment of some security issues like Fast mistake limitation, information honesty, information security. The proposed plan permits clients to review the information with lightweight correspondence and calculation cost. Examination shows that proposed framework is profoundly effective against

noxious information adjustment assault and worker plotting assault. Execution and broad security examination shows that proposed frameworks are provably secure and exceptionally productive. They show that their plan is profoundly productive for worker conspiring assault and vindictive information adjustment assault with least calculation overhead. Execution investigation and broad security shows that the proposed plot is provably secure and profoundly productive.

vi. Behl, A. (2011, December). Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation. In 2011 World Congress on Information and Communication Technologies (pp. 217-222). IEEE.

This paper investigates the security issues identified with the cloud. The paper likewise talks about the current security ways to deal with secure the cloud framework and applications and their disadvantages. At long last, we investigate some key examination difficulties of actualizing new cloud-mindful security arrangements that can give any semblance of pre-emptive assurance for complex and ever powerful Cloud framework, trailed by end where they attempt to involve the entire exploration and attempt to plan a security system which will empower the Cloud suppliers and clients the same to battle against regularly developing security dangers.

There are various security challenges in the haze of which, this paper has attempted to address the most widely recognized and basic ones. A protected cloud is unthinkable except if the virtual condition viz. foundation, VM, interfaces, organize transmissions are secure. Cloud condition request much over the customary security arrangements, which don't plan well to the virtualized situations, in view of the mind boggling and dynamic nature of the distributed computing. As a venturing stone, cloud suppliers and clients should cooperate on characterizing the prerequisites and the points of interest. It is 220 2011 World Congress on Information and Communication Technologies certain that new virtualization-mindful security arrangements ought to be actualized to guarantee the pre-emptive security to the general framework. The cloud security arrangements ought to have the knowledge to act naturally safeguarding and be able to give constant checking, discovery and avoidance of known and obscure dangers.

vii. ***Usman, M., Jan, M. A., & He, X. (2017). Cryptography-based secure data storage and sharing using HEVC and public clouds. Information Sciences, 387, 90-102.***

Scrambling mystery information in compacted video transfers is a moderately new examination region which is drawing in the consideration of scientists. This is basically because of protection and security issues worried about the open mists. In this article, a made sure about conspire has been introduced which shrouds the mystery information in HEVC encoded video transfer, i.e., in packed area. The proposed plot comprises of three significant stages, which are video encoding, information encryption, and unscrambling with/without interpreting. The proposed conspire attempts to keep up the first video transfer size after encryption without influencing the visual nature of video information. In this way, it creates a perfect stage for constant video applications. The mystery information is appropriated in encoded video transfer so it is hard for programmers to remove whole mystery information. This is because of the way that programmers don't have the foggiest idea about the specific areas and examples of the concealing plan, regardless of whether they take the mystery key. Another significant bit of leeway is that their proposed conspire completely bolsters the encoding and unravelling structure of HEVC standard. The video transfer with encoded mystery information can undoubtedly be decoded without getting undermined or indicating any indication of extra concealed data. Test results have demonstrated that the proposed plot keeps up the visual quality with a slight trade off on expanding the size of the encoded video transfer.

viii. ***Garg, P., & Sharma, V. (2014, February). An efficient and secure data storage in Mobile Cloud Computing through RSA and Hash function. IEEE***

At the point when an asset obliged cell phone stores its information on the cloud, there is consistently a major worry of whether the cloud specialist organization stores the documents accurately or not. Security is the principle worry in portable distributed computing. The proposed instrument gives a security component to making sure about the information in portable distributed computing with the assistance of RSA calculation and hash work. This exploration paper has proposed an instrument to give classification and

honesty to the information put away in portable cloud. The proposed plot utilizes RSA calculation with other encryption decoding procedures to make sure about the information in such a way that no spillage of information on cloud could be performed. In this plan encryption is utilized to give security to the information while in communicate. Since the scrambled record is put away on the cloud, so client can accept that his information is secure. In the plan record, just in scrambled structure is moved over the channel, which lessens the issue of data divulgence. No, third individual or gate crasher can get the document since that individual do not knows the key of information proprietor. There is consistently an extension for development in each field of work, so here too. One of the suspicions made in all the models of security is that the TP A is nonpartisan. All the calculations and confirmations are offloaded to TP A so there is a need to accomplish some work for making TP A safer. Future work could be investigating the utilizations of other systems applied in secure capacity administrations of portable cloud condition. Some work should likewise be possible to diminish the overhead of versatile terminal.

ix. **Rasha Samir Abdeldaym, Hatem Mohamed Abd Elkader, RedaHussein, Rasha Samir Abdeldaym. Modified RSA Algorithm Using Two Public Key and Chinese Remainder Theorem**

The proposed model in the paper takes 4 prime number in RSA and instead of using one public key, 2 public key is sent to the receiver. The problem of speed is sorted out by using RSA with Chinese Remainder Theorem. Chinese Remainder Theorem, CRT, is a theorem in mathematics which can be used in the field of cryptography. Its application is computing, which is very important in regard to calculations of algorithmic and computations modular. The Chinese remainder theorem (CRT) is to determine a single integer from its remainders from a set of modulos. It has also got applications in digital signal processing. CRT allows for RSA algorithm implementation very efficiently. The fact that if the same message is encrypted using random key more than one time makes the ciphertext look different every time is used here. The comparison in the paper clearly states the all-time encryption and decryption of RSA-CRT is nearly half to that of RSA. For a 640-bit length plaintext, RSA-CRT to RSA time taken is 26:42. The paper shows in depth research about how it claims RSA-CRT an enhanced algorithm than RSA.

x. ***Pant, V. K., Prakash, J., & Asthana, A. (2015, October). Three step data security model for cloud computing based on RSA and steganography. IEEE***

Information and Data security is a most significant issue of cloud processing and IT industry. In this paper they utilize some strategy to make sure about information in cloud or web. I trust this work help to secure information structure outcast or programmer, who annihilate the significant information. In future I need to work to improve the working of these calculation in term of heartiness or concealing limit and utilize other secure calculation or technique to ensure data (information) on cloud Modem zone of data innovation are completely founded on online help or web administrations. This paper examined security issues in distributed computing frameworks and how they can be forestalled, here they use cryptography and steganography strategy together to make sure about information. RSA calculation is safer than other calculation. They incorporate RSA calculation with other calculation to give greater security to information. In steganography process they get scrambled picture, which appears to be identical to unique picture by natural eye. In the event that they examination the picture double codes then the distinctions would be seen. Else they are incapable to recognize the first picture. The methodology they have use in this paper, will assist with making a solid structure for security of information in distributed computing field or web.

xi. ***Somani, U., Lakhani, K., & Mundra, M. (2010, October). Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. IEEE***

Among the numerous IT powerhouses driven by patterns in distributed computing has not dicey. It gives nearly everybody has brought uplifting news. For ventures, cloud processing is deserving of thought and attempt to fabricate business frameworks as a path for organizations along these lines can without a doubt realize lower costs, higher benefits and more decision; for huge scope industry, After the monetary unrest will be the expense of framework for huge scope pressure appears to be likely; designers, when in the face of distributed computing, through the PaaS model can adequately

improve their own limit, Therefore, the effect of cloud processing on the ISV is the biggest of the numerous jobs; for architects and engineers are concerned. There is the approach of cloud registering will undoubtedly birth various new openings. The mists will develop in size as before long as accessible transfer speed and the relating administration model develop enough, distributed computing will bring a progressive change in the Internet. Cloud registering reported a minimal effort supercomputing administration to give the plausibility, while there are an enormous number of producers behind, there is no uncertainty that distributed computing has a splendid future.

xii. ***Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. International journal of engineering research and applications, 3(4), 1922-1926***

In this paper encryption calculations have been proposed to make cloud information secure, helpless and offered worry to security issues, challenges and furthermore examinations have been made between AES, DES, Blowfish and RSA calculations to locate the best one security calculation, which must be utilized in distributed computing for making cloud information secure and not to be hacked by assailants. Encryption calculations assume a significant job in information security on cloud and by examination of various boundaries utilized in calculations, it has been discovered that AES calculation utilizes least an ideal opportunity to execute cloud information. Blowfish calculation has least memory prerequisite. DES calculation expends least encryption time. RSA devours longest memory size and encryption time. By doing execution for all calculations in IDE apparatus and JDK 1.7, the ideal yield for the information on distributed computing has been accomplished. In the present time request of cloud is expanding so the security of the cloud and client is on top concern. Henceforth, proposed calculations are useful for the present prerequisite. In future a few correlations with various methodologies and results to show adequacy of proposed structure can be given.

xiii. ***Ruj, S., Stojmenovic, M., & Nayak, A. (2012, May). Privacy preserving access control with authentication for securing data in clouds. In 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 556-563). IEEE***

This paper presents a security safeguarding access control plot which not just gives fine-grained get to control yet in addition confirms clients who store data in the cloud. The cloud anyway doesn't have the foggiest idea about the personality of the client who stores data, however just confirm the client's accreditations. Key appropriation is done in a decentralized manner. One restriction is that the cloud knows the entrance strategy for each record put away in the cloud. In future, they might want to ensure the security of client traits as well. In this paper, they propose another security saving validated access control conspire for making sure about information in mists. In the proposed plot, the cloud checks the validity of the client without realizing the client's personality before putting away data. Their plan likewise has the additional element of access control where just legitimate clients can unscramble the put away data. The plot forestalls replay assaults and supports creation, adjustment, also, perusing information put away in the cloud. In addition, their validation also, get to control plot is decentralized and vigorous, in contrast to other get to control plans intended for mists which are brought together. The correspondence, calculation, and capacity overheads are tantamount to brought together methodologies.

xiv. Subashini, S., & Kavitha, V. (2011, October). A metadata-based storage model for securing data in cloud environment. IEEE

In this paper they examined the issues in security in information capacity in cloud condition. To guarantee that the information is secure during the put away period of the existence pattern of the information, they proposed a metadata-based model utilizing which the information dwelling at server farm are burglarized of their qualities and the values are incidentally developed during runtime and destroyed once its usage scope is completed. This makes the information priceless regardless of whether an interloper gains admittance to this information. In spite of the fact that this model will require some quantifiable exertion to be actualized continuously, it gives essential answer for a situation like the Cloud which is indicating an antagonistic potential to turn into the cutting-edge undertaking condition. Executing such a model during the prior periods of the development of the framework will be generally simpler as for actualizing it after part of information take exile in the cloud. This model in blend with their multi-level security model for making sure

about information over transmission will give legitimate cross bars in the wires of malevolent clients.

xv. ***Wang, C., Ren, K., Lou, W., & Li, J. (2010). Toward publicly auditable secure cloud data storage services. IEEE network, 24(4), 19-24.***

Cloud computing has been imagined as the next-gen of big business IT. As opposed to conventional venture IT arrangements, where the IT administrations are under appropriate physical, coherent, and faculty controls, distributed computing moves the application programming and databases to workers in enormous server farms on the Internet, where the administration of the information and administrations are not completely dependable. This, one of a kind trait, raises numerous new security challenges in regions, for example, programming and information security, recuperation, and protection, just as lawful issues in zones, for example, administrative consistence and examining, all of which have not been surely known. In this article they center around cloud information stockpiling security. They first present a organize engineering for viably portraying, creating, also, assessing secure information stockpiling issues. They at that point propose a lot of methodically and cryptographically attractive properties for open examining administrations of reliable cloud information stockpiling security to turn into a reality. Through inside and out investigation, some current information stockpiling security building squares are inspected. The advantages and disadvantages of their pragmatic ramifications in the setting of distributed computing are summed up. Further testing issues for open examining administrations that should be engaged on, are talked about as well. They accept security in cloud computing, a region brimming with difficulties and of vital significance, is still in its outset presently yet will pull in colossal measures of research exertion for a long time to come.

xvi. ***Zhou, L., Varadharajan, V., & Hitchens, M. (2013). Achieving secure role-based access control on encrypted data in cloud storage. IEEE transactions on information forensics and security, 8(12), 1947-1960***

In this paper, first it proposes RBE plot that accomplishes proficient client disavowal. At that point they introduced a RBAC based distributed storage design which permits an association to store information safely in an open cloud, while keeping up the delicate data identified with the association's structure in a private cloud. At that point they have built up a safe distributed

storage framework engineering and have indicated that the framework has a few unrivalled attributes, for example, steady size ciphertext and decoding key. From their trials, they see that both encryption and decoding calculations are effective on the customer side, and decoding time at the cloud can be decreased by having different processors, which is normal in a cloud condition. They accept that the proposed framework has the potential to be valuable in business circumstances as it catches handy access arrangements dependent on jobs in an adaptable way what's more, gives secure information stockpiling in the cloud authorizing these get to strategies.

• Research Gap

Even though RSA algorithms were used in the above papers, the security can still be increased and it is not that much fast.

So, our approach for making it better is by using RSA with CRT and multiple keys with hash functions.

Sr No	Title of the Paper	Author and Year of Publishing	Technique Used	Advantage	Research Gap
1.	**Modified RSA Algorithm Using Two Public Key and Chinese Remainder Theorem**	Rasha Samir Abdeldaym, Hatem Mohamed Abd Elkader, Reda Hussein (2019)	The proposed model in the paper takes 4 prime number in RSA and **instead of using one public key, 2 public key** is sent to the receiver. The problem of speed is sorted out by using RSA with Chinese Remainder Theorem..	CRT allows for RSA algorithm implementation very efficiently. The comparison in the paper clearly states the all time encryption and decryption of RSA-CRT is nearly half to that of RSA.	The paper shows in depth research about how it claims RSA-CRT an enhanced algorithm than RSA. Disadvantage of the proposed methodology is **integrity issue.**

2.	**Intelligent cryptography approach for secure distributed big data storage in cloud computing**	Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017)	They proposed a novel methodology entitled as Security-Aware Effective Distributed Storage (SA-EDS) model. In this model, they utilized their proposed calculations, including Alternative Data Distribution (AD2), Secure Efficient Data Distributions (SED2) and Efficient Data Conflation (EDCon) calculations	Their exploratory assessments had demonstrated that their proposed plan could viably shield significant dangers from cloud-side. The calculation time was shorter than current dynamic methodologies	**Man in the middle attack** still could not be completely avoided using this algorithm.
3.	**Cryptography -based secure data storage and sharing using HEVC and public clouds**	Usman, M., Jan, M. A., & He, X. (2017)	In this article, a made sure about conspire has been introduced which shrouds the mystery information in HEVC encoded video transfer, i.e., in packed area. The proposed plot comprises of three significant stages, which are **video encoding, information encryption, and unscrambling with/without interpreting**.	The simulation results clearly show that the proposed scheme outperforms AES-256 by decreasing the processing time up to 4.76% and increasing the data size up to 0.72% approximately. The proposed scheme can readily be applied to real-time cloud media streaming.	**Confidentiality** was not properly assured in the algorithm.
4.	**Enhancing Distributed Data Storage Security for Cloud Computing Using TPA and AES algorithm**	Shimbre, N., & Deshpande, P. (2015, February).	This paper, talks about the record circulation and **SHA1 strategy** At the point when document is conveyed then information is likewise isolated into numerous workers. So here the need of information security emerges. Each square of record contains its own **hash code**,	Utilizing hash code which will improve client verification process; just approved individual can get to the information. Here, the information is encoded utilizing propelled encryption standard, so information is effectively and safely put away on cloud.	**AES with TPA could no more be the best technique.**

5.	**Three Step Data Security Model for Cloud Computing based on RSA and Steganography Techniques**	Pant, V. K., Prakash, J., & Asthana, A. (2015, October).	This paper examined security issues in distributed computing frameworks and how they can be forestalled, here they use **cryptography and steganography strategy** together to make sure about information. RSA calculation is safer than other calculation. They incorporate **RSA calculation with other calculation** to give greater security to information. In steganography process they get scrambled picture, which appears to be identical to unique picture by natural eye.	The methodology they have use in this paper, will assist with making a solid structure for security of information in distributed computing field or web.	**Recent threat to traditional RSA is Shor's Algorithm.**
6.	**Dynamic remote data auditing for securing big data storage in cloud computing**	Sookhak, M. (2015).	The proposed information examining plan is executed in the genuine condition by utilizing java and C++ to address the target of assessing DRDA technique. The presentation of the DRDA plot was approved by utilizing the benchmark test in the copying condition and broke down the DRDA conspire **by utilizing unmistakable boundaries, for example, length of mark, document size, and likelihood of identification.**	It is found that by employing the proposed RDA method the computational and communication costs of data integrity is reduced. D&CT data structure reduces the computation cost of data update for normal and large-scale files markedly.	**Disadvantage is the algorithm could not secure data efficiently though integrity was ensured.**
7.	**An Efficient and Secure Data Storage in Mobile Cloud Computing through RSA and Hash Function**	Garg, P., & Sharma, V. (2014, February).	The proposed instrument gives a security component to making sure about the information in portable distributed computing with the assistance of **RSA calculation and hash work**. The proposed plot utilizes RSA calculation with other encryption decoding procedures to make sure about the information in such a way that no spillage of information on cloud could be performed.	This exploration paper has proposed an instrument to give classification and honesty to the information put away in portable cloud.	**Slow speed and newly emerged threats like Shor's algorithm are disadvantages.**

8.	**Secure User Data in Cloud Computing Using Encryption Algorithms**	Arora, R., Parashar, A., & Transforming, C. C. I. (2013).	In this paper **encryption calculations have been proposed to make cloud information secure – very easy way to learn which encryption to use according to expected project outcome,** helpless and offered worry to security issues, challenges and furthermore examinations have been made between AES, DES, Blowfish and RSA calculations to locate the best one security calculation, which must be utilized in distributed computing for making cloud information secure and not to be hacked by assailants.	AES calculation utilizes least an ideal opportunity to execute cloud information. Blowfish calculation has least memory prerequisite. DES calculation expends least encryption time.	**RSA devours longest memory size and encryption time**
9.	**Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage**	Zhou, L., Varadharajan, V., & Hitchens, M. (2013)	In this paper, first it proposes RBE plot that accomplishes proficient client disavowal. At that point they introduced a **RBAC based distributed storage** design which permits an association to store information safely in an open cloud, while keeping up the delicate data identified with the association's structure in a private cloud.	At that point they have built up a safe distributed storage framework engineering and have indicated that the framework has a few unrivaled attributes, for example, steady size ciphertext and decoding key.	**RBAC serves the purpose only if the encryption mechanism used is upto mark. It depends on algorithm in use, just RBAC does not ensure integrity of data.**
10.	**Privacy Preserving Access Control with Authentication for Securing Data in Clouds**	Ruj, S., Stojmenovic, M., & Nayak, A. (2012, May).	This paper presents a security safeguarding access control plot which not just gives fine-grained get to control yet in addition confirms clients who store data in the cloud. In the proposed plot, **it checks the validness of the client without realizing the client's personality before putting away data**	**The plot forestalls replay assaults and supports creation, adjustment, also, perusing information put away in the cloud**	**Internal threats like irritated employees minimizing security have not been discussed.**

• Novelty

- ***Using RSA-CRT algorithm along with hash function to ensure integrity***
- ***Generally, RSA application uses 1 public key, 2 public keyed technique used in our project.***
- ***RSA with multi-keys is slow, RSA-CRT with multi-keys ensures a better encryption and decryption speed.***

• Techniques to be Used & Experimental Setup

The modified cryptosystem technique uses a combination of:

a) RSA using Multi-Keys and Chinese Remainder Theorem

b) Hash function using SHA-512

a) CRT:

CRT, Chinese Remainder Theorem, is the one of the main theorems of mathematics. It is can be used in the cryptography field. CRT continues to present itself in new contexts and open vistas for new applications types. Original field of this application is the computing, continues to be important as regards some aspects of algorithmic and computations modular. The Chinese remainder theorem (CRT) is to determine a single integer from its remainders from a set of moduli. It has applications in various areas, such as digital signal processing and cryptography. CRT allows for implementation the RSA algorithm efficiently.

RSA with multiple keys:

Generated multiple keys (two public and two private keys) in RSA algorithm. In this algorithm the computation time is more because of multiple keys, but the security is more compared to the standard algorithm (RSA). We are using two

public and private keys in modified RSA algorithm, in which we will be used four prime number and get public key and private key, also using two public keys for encrypting and two private keys for decrypting.

Why RSA with multiple keys and CRT?

The RSA cryptosystem takes great computational cost. In many RSA applications, user uses a small public key to speed up the encryption operation. However, the decryption operation has to take more computational cost to perform modular exponentiation by this case, but It provides high security and it is easy to implement, RSA is an asymmetric key algorithm (public key). The standard RSA used two prime numbers to generate one public key and one private key to encrypt and decrypt, which makes it less secure which it is easily decomposed.

The proposed model for RSA cryptosystem contains four prime numbers and by using two public keys instead of sending one public key directly, so that if an attacker has an opportunity of hacking and getting the component of public key, they cannot get the private key value by brute force search. On the other hand, RSA works quite slowly when its bit size increases after 1024 bits, so that to improve the speed on RSA decryption side used the Chinese remainder theorem (CRT) by which the scheme is semantically secure also. The objective of this paper enhancement the performance by using Chinese and increased the security by using two public keys in the encryption.

In the field of cryptosystem, the algorithm is used for functionality for modular computation. The size of the exponent decryption, d and the modulus, n is very important because the complexity of the decryption in RSA depends directly on it. The exponent decryption specifies the numbers of multiplication modular, there are necessary to perform the exponentiation. The modulus, n play an important role in determined the size of the intermediate results. A way to reduce the size of both d and n is by using the Chinese Remainder theorem.

The RSA decryption and signature operation can be speeded up by using the CRT, where the factors of the modulus N (i.e., P and Q) are assumed to be known. By CRT, the computation of M can be partitioned into four parts:

$$mp = Cp^{dp} \bmod p$$

$$mq = Cq^{dq} \bmod q$$

[22]

$$mr = Cr^{dr} \bmod r$$

$$ms = Cs^{ds} \bmod s$$

Where

$$Cp = C1 \bmod p$$

$$Cq = C1 \bmod q$$

$$Cr = C1 \bmod r$$

$$Cs = C1 \bmod s$$

And

$$C1 = C^{t} \bmod (z). \quad \text{[using the first private key (t,z)]}$$

This reduces computation time since dp, dq, dr, ds< D and Cp,Cq,Cr,Cs< C. In fact, their sizes are about half the original sizes. In the ideal case, as both the sizes of d and n are reduced, we can have a speedup of about 4 times. But the proposed technique increased the security.

Table 3: Comparison between standard RSA, RSA-CRT, RSA by multi keys and proposed technique with results

Size in bits	All time encryption and decryption of RSA in MS	All time encryption and decryption of RSA-CRT in MS	All time encryption and decryption of RSA by multi keys in MS	All time encryption and decryption of proposed technique in MS
640	42	26	84	66
1040	48	29	93	71
1136	63	32	113	85

The ***proposed technique is more secure as compared to original RSA algorithm and RSA-CRT***. And it ***enhanced the performance the algorithm in decryption because it used the CRT*** in decryption, thus the proposed technique faster than RSA by multi keys. ***It reduces the cost of computation***. Although it takes long time to perform it as compared to original RSA.

RSA using Multi-Keys & Chinese Remainder Theorem

Take a document from the cloud, which is to be sent to another person. The document will be chomped by a Hash function into a few lines and will have a referral pair of words such as “message digest”. Using our software, we will encrypt the message using our private key which gives the digital signature. RSA with multiple keys and CRT will then be used to encrypt the signature with the receiver’s public key. The receiver can decrypt the cipher text to plain text using their private key and our public key to verify the signature.

The proposed algorithm is trying to modify RSA cryptosystem by improving its speed by using Chinese remainder theorem and its security by taking 4 prime numbers instead of 2 and two public key pairs instead of one.

The procedure for generating the key: -

1. We generate four large prime numbers p, q, r and s.
2. We calculate the value of (n, z), i.e.

$$n = pq$$

$$z = rs.$$

3. We find the value of $\varphi(n)$ and $\varphi(z)$

$$\varphi(n)= (p - 1)(q - 1)$$

$$\varphi(z) = (r - 1)(s - 1).$$

4. We choose random integers e, g such that

$$1 < e < n \text{ and } GCD(e, \varphi(n)) = 1$$

$$1 < g < z \text{ and } GCD(g, \varphi(z)) = 1.$$

5. We calculate the value of d and t such that

$$ed = 1 \bmod (\varphi(n)).$$

$$tg = 1 \bmod (\varphi(z)).$$

6. We calculate the value of dp,dq,dr and ds

$$dp = d \bmod (p - 1)$$

$$dq = d \bmod (q - 1)$$

$$dr = d \bmod (r - 1)$$

$$ds = d \bmod (s - 1).$$

NOW,

$$\text{The Public key } KU =< (e, n),(g, z) >$$

$$\text{The Private key } KV =< t, z, dp, dq, dr, ds>.$$

Encryption for Proposed Technique:

For encryption of the message M, the following steps are as followed:

1. We convert the message M in integer form, in the range [0 to n − 1].

2. We calculate ciphertext C1 using first public key i.e e by

$$C1 = M^e \bmod (n)$$

3. We calculate Ciphertext C using second public key g and C1 by

$$C = C1^{g} \bmod (z).$$

4. We send the ciphertext C to the receiver.

Decryption for Proposed Technique:

For decryption of ciphertext C, we follow these steps:

1. First we find C1 using the first private key (t, z)

$$C1 = C^{t} \bmod (z).$$

2. We do following Calculations

$$Cp = C1 \bmod p$$

$$Cq = C1 \bmod q$$

$$Cr = C1 \bmod r$$

$$Cs = C1 \bmod s$$

3. Then we calculate:

$$mp = Cp^{dp} \bmod p$$

$$mq = Cq^{dq} \bmod q$$

$$mr = Cr^{dr} \bmod r$$

$$ms = Cs^{ds} \bmod s$$

4. Now after combining mp, mq, mr and ms,

   we get back our original plaintext message M.

b) Hashing

Hashing is an algorithm that calculates a fixed-size bit string value from a file. A file basically contains blocks of data. Hashing transforms this data into a far shorter fixed-length value or key which represents the original string. The hash value can be considered the distilled summary of everything within that file.

A good hashing algorithm would exhibit a property called the avalanche effect, where the resulting hash output would change significantly or entirely even when a single bit or byte of data within a file is changed. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers. A hash is usually a hexadecimal string of several characters. Hashing is also a unidirectional process so you can never work backwards to get back the original data.

A good hash algorithm should be complex enough such that it does not produce the same hash value from two different inputs. If it does, this is known as a hash collision. A hash algorithm can only be considered good and acceptable if it can offer a very low chance of collision.

What are the benefits of Hashing?

One primary utilization of hashing is to look at two documents for uniformity. Without opening two record documents to think about them in exactly the same words, the determined hash estimations of these documents will permit the proprietor to know promptly on the off chance that they are extraordinary.

Hashing is likewise used to confirm the honesty of a document after it has been moved starting with one spot then onto the next, regularly in a record reinforcement program like SyncBack. To guarantee the moved document isn't undermined, a client can analyse the hash estimation of the two records. In the event that they are the equivalent, at that point the moved document is an indistinguishable duplicate.

In certain circumstances, an encoded record might be intended to never change the document size nor the last adjustment date and time (for instance, virtual drive holder documents). In such cases, it is difficult to tell initially if two comparative documents are unique or not, however the hash esteems would handily distinguish these records in the event that they are extraordinary.

Types of Hashing

There are many different types of hash algorithms such as RipeMD, Tiger, xxhash and more, but the most common type of hashing used for file integrity checks are MD5, SHA-2 and CRC32.

MD5 - An MD5 hash function encodes a string of information and encodes it into a 128-bit fingerprint. MD5 is often used as a checksum to verify data integrity. However, due to its age, MD5 is also known to suffer from extensive hash collision vulnerabilities, but it's still one of the most widely used algorithms in the world.

SHA-2 – SHA-2, developed by the National Security Agency (NSA), is a cryptographic hash function. SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256.

CRC32 – A cyclic redundancy check (CRC) is an error-detecting code often used for detection of accidental changes to data. Encoding the same data string using CRC32 will always result in the same hash output, thus CRC32 is sometimes used as a hash algorithm for file integrity checks. These days, CRC32 is rarely used outside of Zip files.

SHA2

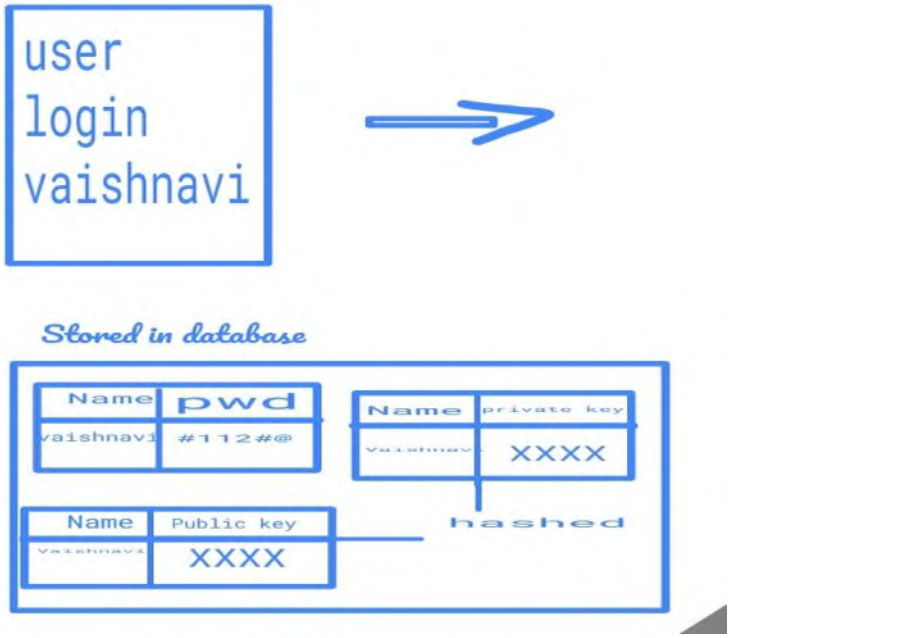
Because of the uncovered weaknesses of SHA-1, cryptographers adjusted the calculation to deliver SHA-2, which comprises of not one but rather two hash capacities known as SHA-256 and SHA-512, utilizing 32-and 64-piece words, separately. There are extra shortened forms of these hash capacities, known as

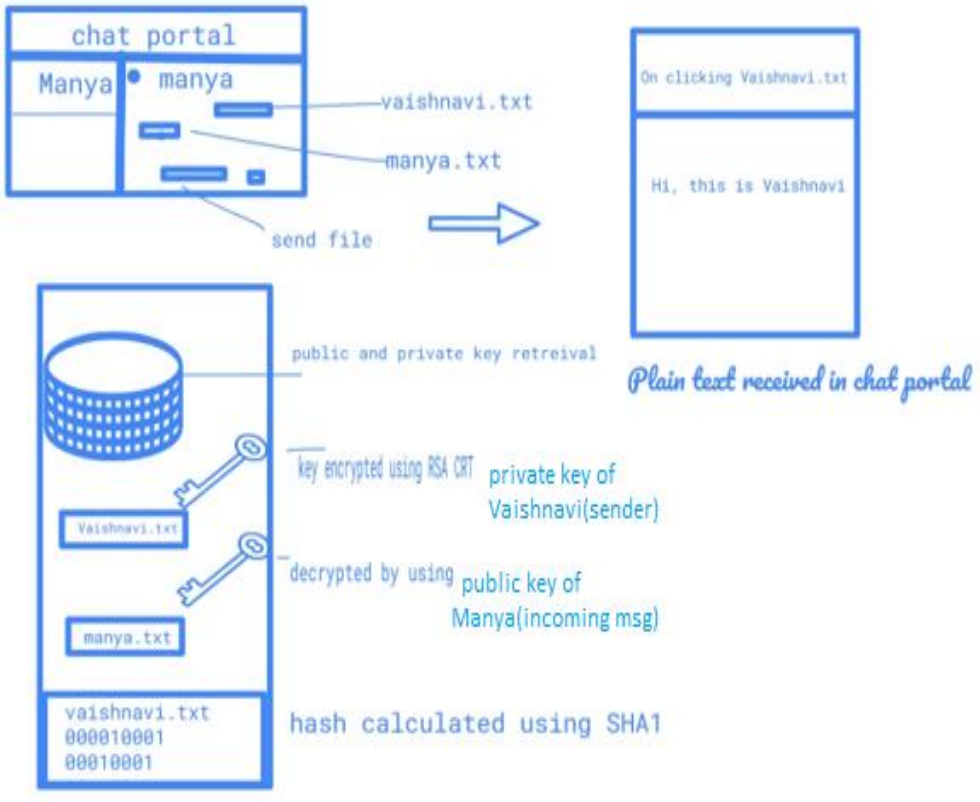
SHA-224, SHA-384, SHA-512/224, and SHA-512/256, which can be utilized for either part of the calculation.

SHA-1 and SHA-2 contrast in a few different ways; mostly, SHA-2 produces 224-or 256-sized reviews, though SHA-1 delivers a 160-piece digest; SHA-2 can likewise have square sizes that contain 1024 pieces, or 512 pieces, as SHA1.

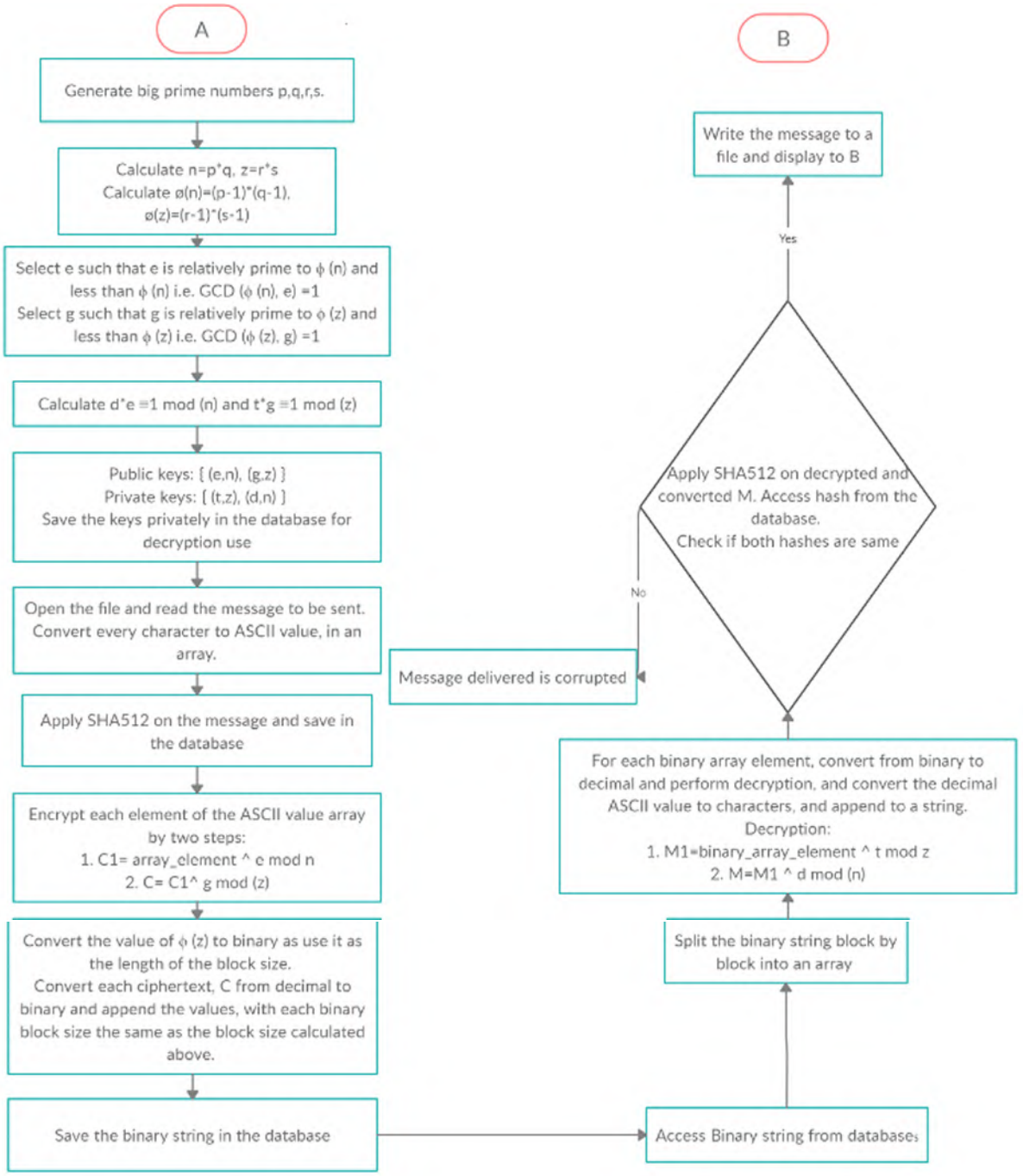
Beast power assaults on SHA-2 are not as successful as they are against SHA-1. An animal power looking for finding a message that relates to a given overview of length LL utilizing beast power would require 2^L2L assessments, which makes SHA-2 much more secure against these sorts of assaults.

• Proposed Model / Algorithm / Framework

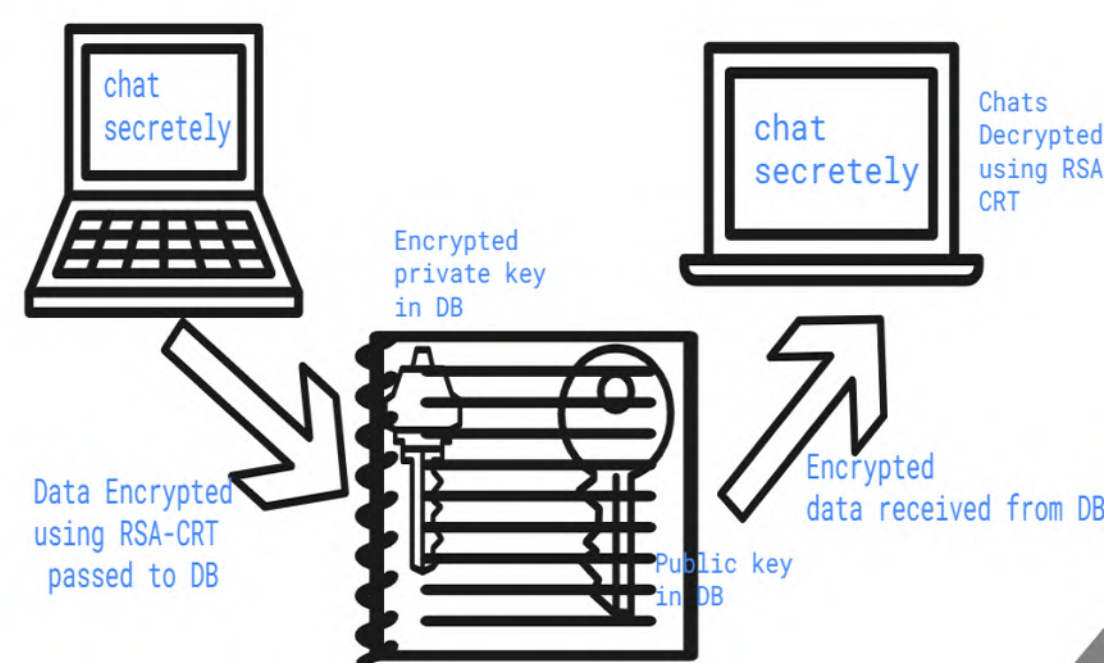




- **Algorithm Architecture**



- **Implementation:**



We have made a ***secure data exchange app where files are encrypted using RSA-CRT algorithm.***

For the first time, if a ***user signup for the portal***, a dynamic table is created in the database for the user which stores his/her password for the portal. A separate table each for ***storing private key and public key of every user***. All the ***password/keys of user are stored by applying cryptographic hash function MD5*** in database as storing plain text passwords in the database is no less than a sin.

When the user login successfully to the portal, he is redirected to his chat box (To do so, the contact's stored in the database is extracted dynamically). All the previous exchanged files appear in the chat box.

Now when the user sends a file? What happens?

When a file is sent, the file is uploaded in the database which serves as a Cloud to the user. Now, the file stored in the database (which will be used further for retrieval purpose by user) is actually ***large binary values (as a result of RSA-CRT encryption***). ***For implementing RSA-CRT, public key and private key is retrieved from database. SHA-1 is used for ensuring integrity of the data***. So

the third party has no access to the data sent and received between two users. It acts as Cloud since the user need not download the exchanged file, just by clicking on the file he can view the current and previously exchanged file.

• Overview of Algorithm in project

The algorithm implementation is described under "RSA USING MULTI-KEYS AND CHINESE REMAINDER THEOREM".

For encryption:

e, g, n and z (public keys are used), where

$n = pq$

$z = rs.$

(four large prime numbers -p, q, r and s)

The values are fetched from database.

It's a double encryption where firstly ciphertext C1 is generated:

$C1 = M^e \bmod (n)$

which is further encrypted

$C = C1^g \bmod (z)$

C is ciphertext stored in db.

For decryption:

Private key t, z, dp, dq,dr, ds is used.

We find C1 using

$C1 = C^t \bmod (z)$

and then C is calculated using dp,dq,dr,ds as already stated in the algorithm part.

The private key is retrieved from the db while users try to retrieve the files/data. To ensure integrity, SHA 1 hashing is used. If the message's hash calculated at the receiver's end is not same as that of sender, then the message is modified.

Common Attacks

Cryptography wouldn't be as fast created if not for the assaults that bargain their adequacy. One of the most well-known assaults is known as the preimage assault, where pre-figured tables of arrangements are utilized in an animal power way so as to split passwords. The arrangement against these sorts of assaults is to make a hash work that would take an assailant an extreme measure of assets, for example, a huge number of dollars or many years of work, to discover a message relating to a given hash esteem.

Most assaults infiltrating SHA-1 are crash assaults, where a non-sensical message creates a similar hash an incentive as the first message. By and large, this requires some serious energy relative to 2^{n/2}2 n/2 to finish, where nn is the length of the message. This is the explanation the message digests have expanded long from 160-piece processes in SHA-1 to 224-or 256-piece processes in SHA-2.

Different assaults exist that endeavour to misuse numerical properties so as to split hash capacities. Among these is the birthday assault, where higher probability of impacts is discovered when utilizing irregular assaults with a fixed number of letter mixes (see the categorize rule), or the rainbow table assault, where a pre-processed hash table is utilized to switch a hash work so as to split passwords**.**

Why using SHA-2?

Pragmatic assaults against the SHA-1 calculation are near being in reach of government offices, possibly empowering assailants to mimic secure sites. SHA-2 doesn't experience the ill effects of SHA-1's numerical shortcomings and offers hash capacities with digest lengths of 224-, 256-, 384-or 512-bits with SHA-256 and SHA-512 the most regularly utilized. Most of SSL authentications are utilizing 2048-piece keys, and now there is no compelling reason to consider a more drawn-out key length except if a declaration is being utilized for an all-encompassing timeframe, (for example, an in-house endorsement authority or an

OpenPGP essential key). Some equipment - including savvy cards and perusers - don't yet uphold keys greater than 2048-pieces, and longer keys utilize more CPU cycles during encryption and validation. The NIST conjectures that 2048-piece keys will be substantial up to about the year 2030.

SHA-2 authentications are viable with most refreshed present day Web programs, OS stages, mail customers and cell phones. For ventures running their own sites, website admins need to demand new SHA-2 declarations to supplant any testaments utilizing SHA-1 and lapsing after January 1, 2017, in any case their workers won't be trusted by Windows-based gadgets (all Windows gadgets will quit trusting SHA-1 endorsements after this date).

• Expected Outcome

The expected outcome is achieving CIA triad along with authentication on Cloud. To do so, we aimed at following:

1) A hybrid algorithm by combining RSA with CRT (Chinese Remainder Theorem) and keeping the vulnerability of RSA using Shor's algorithm in account.

A faster yet secure (ensuring CIA triad) methodology is the outcome of the project. Security threats are ever evolving. Cracking RSA with Shor's Algorithm is another threat that this security algorithm is facing today. A hybrid algorithm keeping all this in mind is our aimed outcome.

2) Hash function making the authentication and data integrity process more effective

Hash functions can be used for digital signatures or message authentication, pseudo-random number generation, key derivation, and cryptography protocols. The hash function algorithms would help compute a fixed length cryptography hash for a given data (message digest).

3) Deployed platform that gives a look and feel of the proposed method

Once the proposed method is ready it would be deployed using frontend and backend to see the effectiveness of the method and its comparison with method in use (RSA).

One of the most vital outcomes of the project would be:

4) **In depth knowledge about RSA, Shor's and RSA-CRT algorithm, hash function, to each of the project members** carried out with detailed discussion on its vulnerability and how to overcome it i.e., our practical experience with the course.

- # Sample Code

D: > Win Sem 2020-21 > CSE1011 - Cryptography Fundamentals - A2 - L13+L14 > Project > algorithm.php

```php
<?php
$p=101;echo $p." =p  ";echo"<br>";
$q=103;echo $q." =q  ";echo"<br>";
$r=107;echo $r." =r  ";echo"<br>";
$s=109;echo $s." =s  ";echo"<br>";
$n=$p*$q;echo $n." =n  ";echo"<br>";
$z=$r*$s;echo $z." =z  ";echo"<br>";
$phin=($p-1)*($q-1);
echo $phin." =phin ";
echo"<br>";$phiz=($r-1)*($s-1);
echo $phiz." =phiz  ";echo"<br>";
// Function to print primes for e n
$count=0;
$arrayp=array(0,1);
$isit=0;
// Driver Code
$nn = $phin;
for ($i = 2; $i<= $nn; $i++)
{
    if($nn<=1)
    $isit=0;
    for($i=2;$i<$nn;$i++)
    {
        if($nn%$i==0)
        $isit=1;
    }
    if ($isit==1)
    {
        $arrayp[$count]=$i;
        $count++;
    }
}
$cc=count($arrayp);
/*for($i=0;$i<$cc;$i++)
{
    echo $arrayp[$i];
    echo("<br>");
}*/
for($i=0;$i<$cc;$i++)
{
    if(gmp_gcd($arrayp[$i],$phin)==1)
    $e=$arrayp[$i];
}
//sdsd
echo $e." = e";
echo"<br>";
// Function to print primes for g z
$count=0;
$arrayp=array(0,1);
$isit=0;
// Driver Code
$ng = $phiz;
```

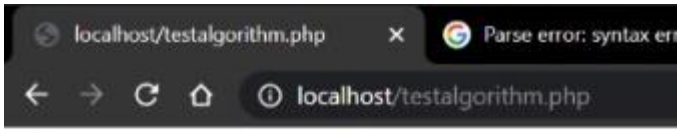
```php
for ($i = 2; $i<= $ng; $i++)
{
    if($n<=1)
    $isit=0;
    for($i=2;$i<$ng;$i++)
    {
        if($ng%$i==0)
        $isit=1;
    }
    if ($isit==1)
    {
        $arrayp[$count]=$i;
        $count++;
    }
}
$cc=count($arrayp);
/*for($i=0;$i<$cc;$i++)
{
    echo $arrayp[$i];
    echo("<br>");
}*/
for($i=0;$i<$cc;$i++)
{
    if(gmp_gcd($arrayp[$i],$phin)==1)
    $g=$arrayp[$i];
}
//sdsd
echo $g." = g ";
echo"<br>";
$d=gmp_invert($e,$phin);
echo $d." =d ";
echo"<br>";
$t=gmp_invert($g,$phiz);
echo $t." =t ";echo"<br>";
$dp=$d % ($p-1);$dq=$d % ($q-1);
$dr=$d % ($r-1);$ds=$d % ($s-1);
//public keys: (e,n),(g,z)
//private key:(t,z,dp,dq,dr,ds)
//variable M is message in integer form
$M=113;echo $M." is the messages to be sent.<br> ";
//$count=strlen($M);
//encryption
$C1= pow($M,$e) % $n;
$C= pow($C1,$g) % $z;
//$C=pow($M,$e)%$n;
echo $C." =c ";
echo"<br>";
$prikeyar1=array($t,$z);
$prikeyar2=array($d,$n);
$pubkeyar1=array($e,$n);
$pubkeyar2=array($g,$z);
$privkey=join($prikeyar1);
$pubkey=join($pubkeyar1);
echo" public key1: (".$e.",".$n.")<br>";
echo" public key2: (".$g.",".$z.")<br>";
echo" private key1: (".$t.",".$z.")<br>";
echo" public key2: (".$d.",".$n.")<br>";
//SHA
//hash ( 'sha512' , $C [, bool $raw_output = FALSE ] ) : string
$hashed = hash('sha512', $C);
$privkey1=openssl_pkey_get_private("file://C:\Users\HP\Downloads\openssl-toolkit\server.pem");
openssl_sign($C,$signature,$privkey1);
//decryption
$pubkey1=openssl_pkey_get_public("file://C:\Users\HP\Downloads\openssl-toolkit\server.crt");
$ok=openssl_verify($C,$signature,$pubkey1);
if ($ok==1)
{
echo"Signature is good<br>";
}
elseif($ok==0)
{
echo"Signature is bad<br>";
}
else
{
echo"ugly, eroor checking signature<br>";
}
$C1= pow($C,$t) % $z;
$Cp= $C1 % $p;
$Cq= $C1 % $q;
$Cr= $C1 % $r;
$Cs= $C1 % $s;
$mp= pow($Cp,($d*$p)) % $p;
$mq= pow($Cq,($d*$q)) % $q;
$mr= pow($Cr,($d*$r)) % $r;
$ms= pow($Cs,($d*$s)) % $s;
```

```
//$message=pow($C,$d)%$n;
//echo $message." mm ";
$hashed2=hash('sha512',$C);
if($hashed==$hashed2)
{
echo"Hashes are the same";
    echo"<br>";
}
echo $mp." is the message received.";
?>
```

- **Experimental Results Obtained**

Algorithmic Testing:

```
Enter path to store certificate files: C:/Users/HP/Downloads/openssl-toolkit

Generating a Key and CSR
Enter password for private key:
Confirm password:

Generating RSA private key, 2048 bit long modulus (2 primes)
.....
.....
.....+++++
.....+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AE
State or Province Name (full name) [Some-State]:Abu Dhabi
Locality Name (eg, city) []:Abu Dhabi
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MYOWN
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:Shruti
Email Address []:shrutivarsha2000@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1111
An optional company name []:MYOWN

server.key can be found at /c/Users/HP/Downloads/openssl-toolkit/server.key
server.csr can be found at /c/Users/HP/Downloads/openssl-toolkit/server.csr

Signing certificate.
Enter amount of days certificate will be valid for(ie. 730): 300
Server certificate created at /c/Users/HP/Downloads/openssl-toolkit/server.cr

Creating PEM...
Creating server.pem at /c/Users/HP/Downloads/openssl-toolkit/server.pem
```



17 =p
31 =q
17 =r
31 =s
527 =n
527 =z
480 =phin
480 =phiz
1 = e
1 = g
1 =d
1 =t
8 is the messages to be sent.
8 =c
public key1: (1,527)
public key2: (1,527)
private key1: (1,527)
public key2: (1,527)
Signature is goodHashes are the same
8 is the message received.

• Inference

a. <u>The algorithm successfully generated keys for the Modified RSA with CRT algorithm</u>

Keys as shown in the figure above are successfully generated using RSA-CRT. 4 prime nos. are used which makes the algorithm way more efficient than RSA ensuring data security.

b. <u>Encryption and Decryption using modified RSA and CRT was a success.</u>

Confidentiality is assured as we can see a successful encryption and decryption at our localhost. The message being sent is very securely put in DB and the correct message is as it is forwarded to the user at the other end.

c. <u>Hashing the encrypted message and obtaining a signature was achieved and verified.</u>

Signature for authentication purpose and our motto to achieve integrity is achieved by implementing this SHA algorithm along with RSA CRT to avoid attacks like MMT.

d. The message that was encrypted and sent was successfully retrieved at a faster rate. The message was the same.

Using RSA CRT makes algorithm faster thus ensuring the availability way more than existing algorithm. Slow processing time can also lead to Denial of services for users.

e. Without using built-in functions for RSA and key generation, the algorithm succeeded in execution.

It is self-made code without any reference from already existing code source. We as a team learnt a lot in this process. The experimental output also corresponds to our attempt of understanding and knowing the current scenario and how can we make it better.
The knowledge we gained is also one of our experimental output.

A look and feel of our experimental result:

1. Confidentiality Testing:

Encryption-

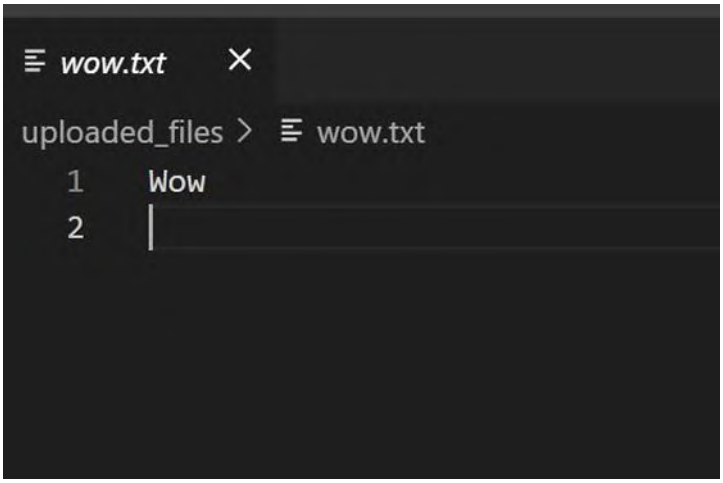


Figure 1 : Original plain text ("Wow")

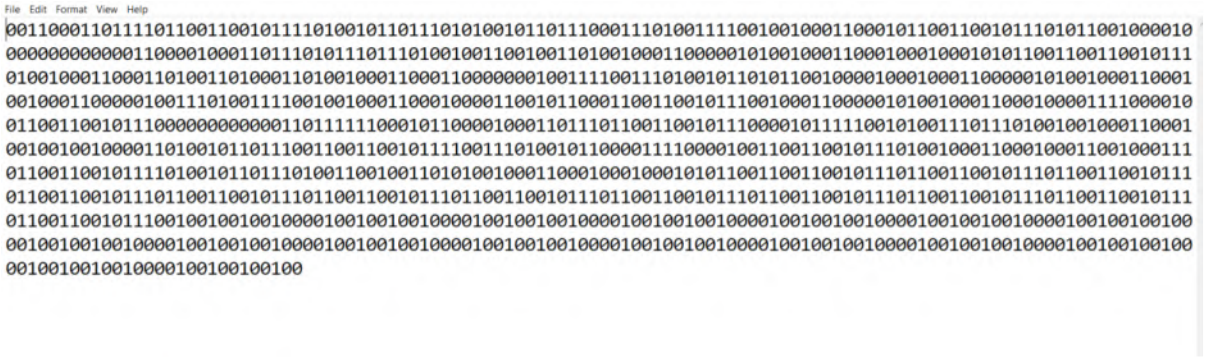


Figure 2 : Encrypted data for original plain text

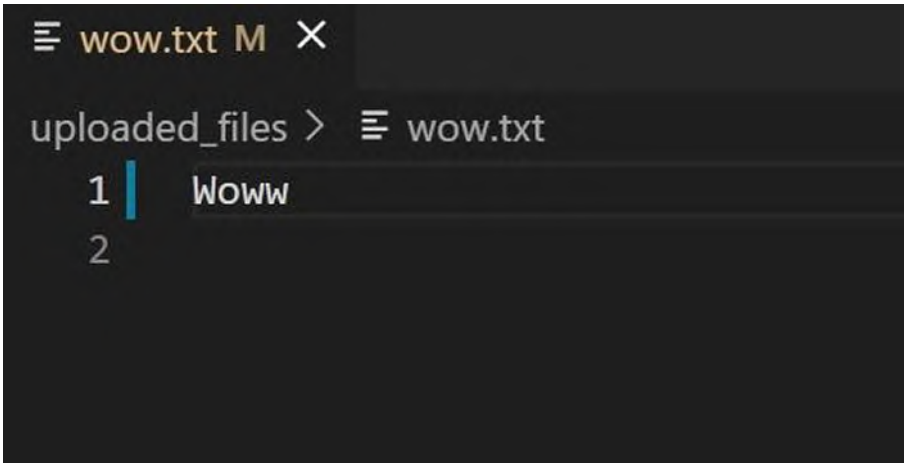


Figure 3 : Plaintext (after slight change)

Figure 4 : Encrypted data of slightly changed plaintext

We can see that just by a small change merely changing “Wow” with “Woww”, there is a huge change in blocks.

2. Authorization Testing:

Testing our login web portal:

2.1. Successful Login

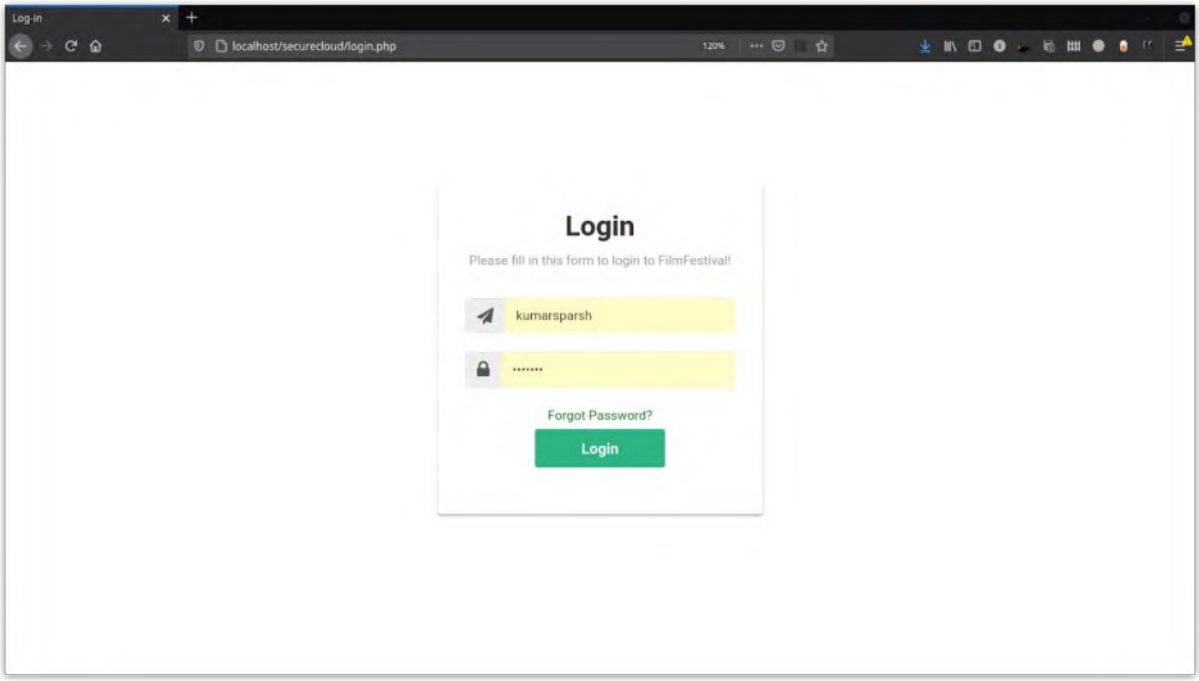


Figure 5 : User login interface

2.2. Unsuccessful Login

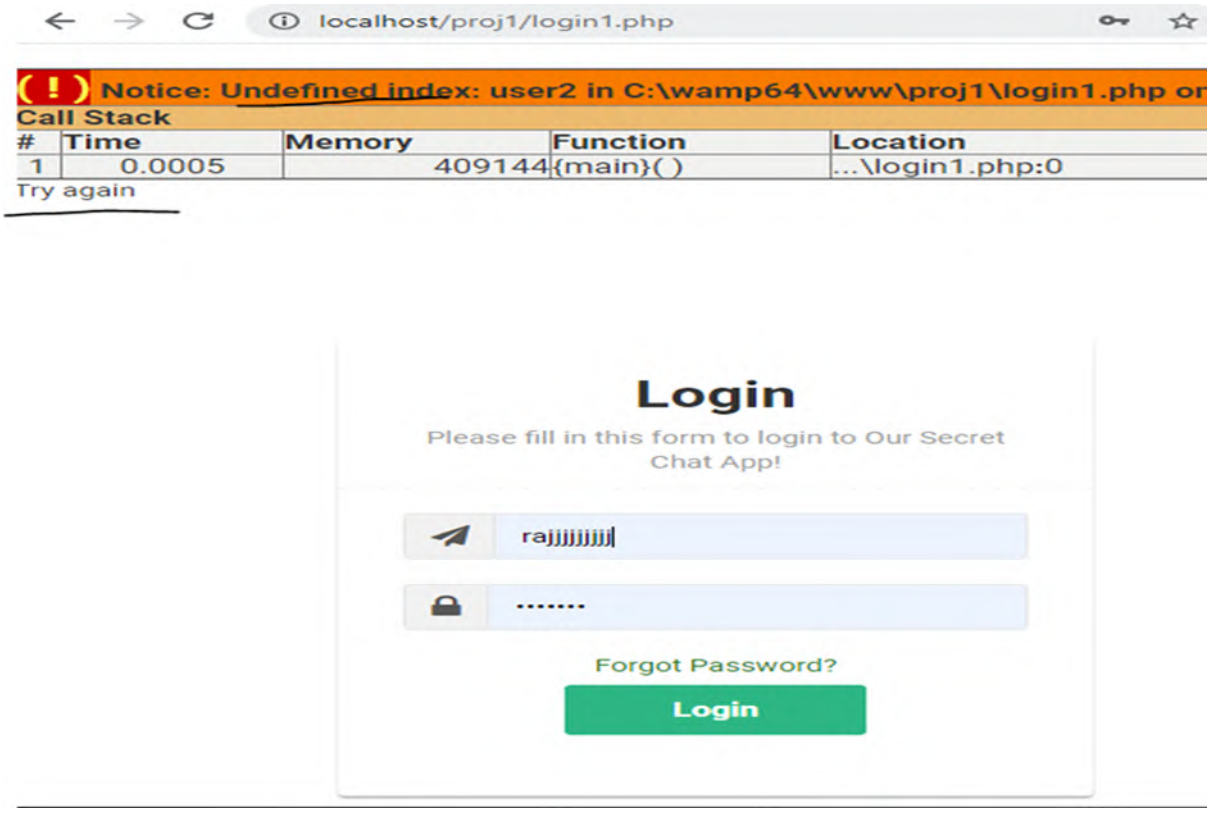


Figure 6 : Unauthenticated user trying to access

The user is not able to access since he is not registered to the portal.

3. Integrity Testing: Backend

Every password is stored in hashed manner, even the owner is unaware of what the keys are.

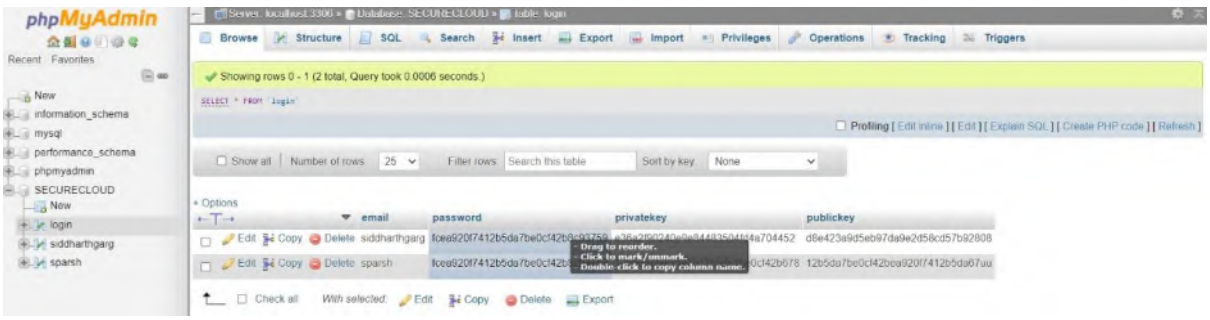


Figure 7 : Login database of registered chat portal user

Figure 8 : User info hash stored in database

In database, file is stored in binary format, i.e., **integrity maintained.**

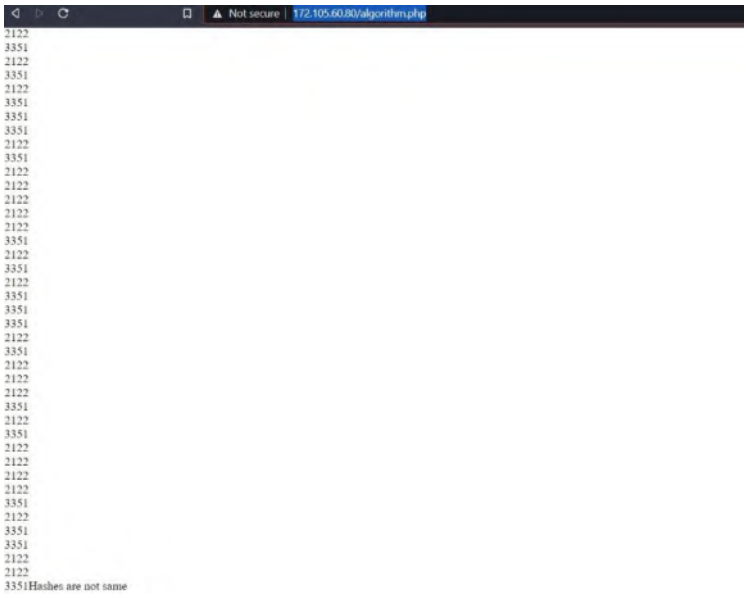


Figure 9 : Hash gets changed if data is tampered with

Hash won't be the same if data in the database gets tampered with.

Users can cross check the authenticity, though the scheme ensures no attacks, we purposefully changed data in mid in code to expose this functionality of our algorithm.

4. Availability Testing: Speed Factor

5 files are encrypted and decrypted using RSA and RSA-CRT. The time taken for encryption and decryption is as follows:

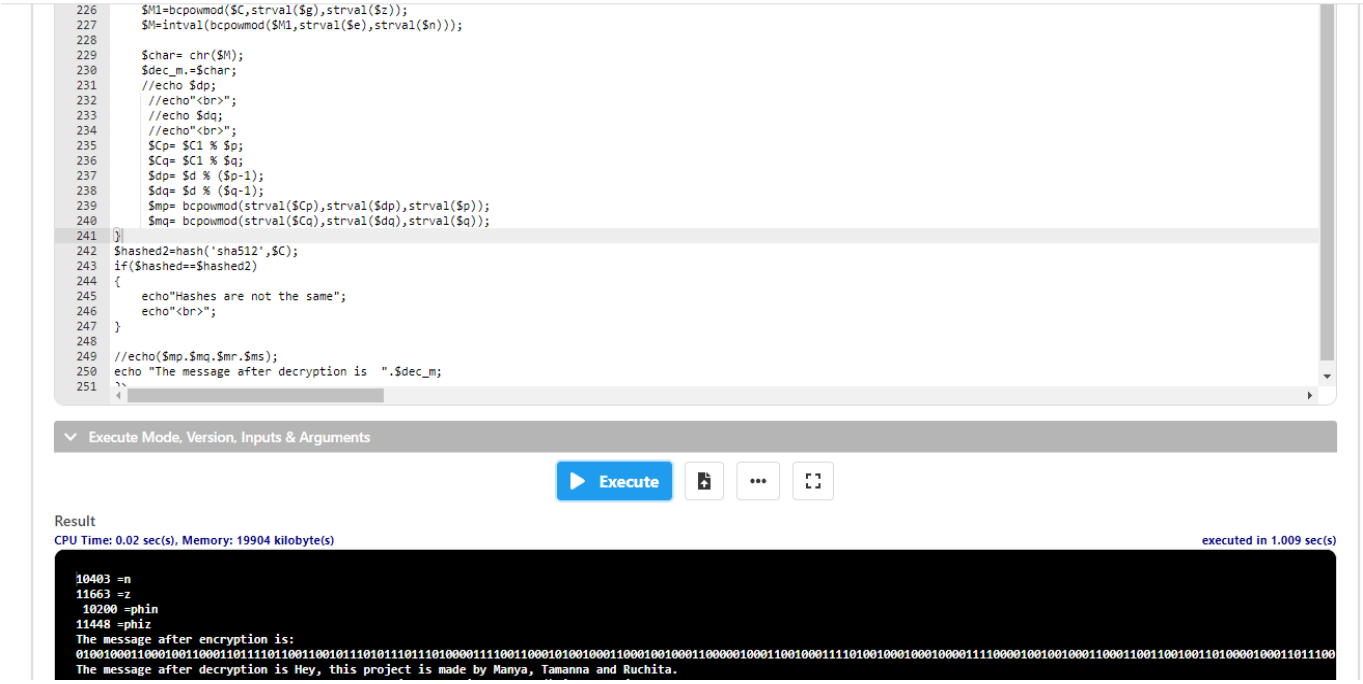
File	Encryption of RSA (nano second)	Encryption of RSA-CRT (nano second)
1	330,130	371,148
2	316,803	348,592
3	330,131	358,845
4	344,485	375,250
5	351,661	382,426

Figure 10 : Results of Comparison Time RSA encryption and RSA-CRT 1024-bits

File	Decryption of RSA (nano second)	Decryption of RSA-CRT (nano second)
1	103,534,963	29,710,330
2	102,998,756	29,400,697
3	103,437,564	29,477,594
4	103,300,180	29,538,084
5	103,515,483	29,730,836

Figure 11 : Results of Comparison Time Decryption RSA and RSA-CRT 1024-bits

Executing the code using compiler to get speed,



• Comparative Analysis

- **Complex but Secure:**

1. **Secure option for application needing faster communication:**

We can see that on average encryption of RSA-CRT is 367252.2 while that of RSA is 334642. Thus, the RSA-CRT takes 33610.2 nano seconds more than RSA while encrypting 1024-bit block. But while decryption, on average decryption of RSA-CRT is 29,571,508.2 while that of RSA is 103,357,389. Thus, the RSA-CRT takes 73,785,800.8 nano seconds less than RSA while decrypting 1024-bit block.

Seeing the overall mechanism of encryption and decryption RSA-CRT is 3 times faster than RSA by 73,752,270.6 nano seconds while applying the mechanism. This proves that despite of computational overhead, RSA- CRT is faster than RSA.

Though the computational overhead is high, time taken by this mechanism is less. Thus, it is a better option to be used in fast communication applications like transaction-oriented or business plans oriented.

2. **Key Generation Overhead but is less vulnerable to attacks.**

Attacks like Fault Injection or Side-Channel Analysis can't be mitigated using RSA but RSA-CRT can protect user's data from these attacks. RSA-CRT prevent the attacker from obtaining the signature when a fault has been induced during the computation. Not only this, it makes other attacks (with big prime numbers) like factoring large number or common modulus highly infeasible to be done in the exchanging data's lifetime.
Timing attacks on RSA-CRT is highly infeasible since in timing attack precise time of decryption the card takes can help an attacker find or discover the private decryption exponent d. But in RSA-CRT, the decryption time is 3 times lesser than RSA (which is safe since it's still in use).

3. **Multi-keys with RSA-CRT makes attacking highly infeasible in the data's lifetime.**

Calculating private keys (t, z, dp, dq, dr, ds) where t is obtained by $tg = 1 \bmod (\varphi(z))$. (g is random integers such that $\gcd(g, \varphi(z)) = 1$).

$dp = d \bmod (p - 1)$
$dq = d \bmod (q - 1)$
$dr = d \bmod (r - 1)$
$ds = d \bmod (s - 1)$.

is highly infeasible in the data's lifetime.

The private key calculation of RSA is just {d,n} which is comparatively more prone to be hacked in data's lifetime.

4. **Reduction in the value of d and n can still yield same promising result in RSA-CRT with multi-keys.**

In the field of cryptosystem, many algorithms use functionality of modular computation. The size of the exponent decryption, d and the modulus, n is very important because the complexity of the decryption in RSA depends directly on it. The exponent decryption specifies the numbers of multiplication modular, there are necessary to perform the exponentiation. The modulus, n play an important role in determined the size of the intermediate results. A way to reduce the size of both d and n is by using the Chinese Remainder theorem since the exponentiation modular is performed on half the bit size of n

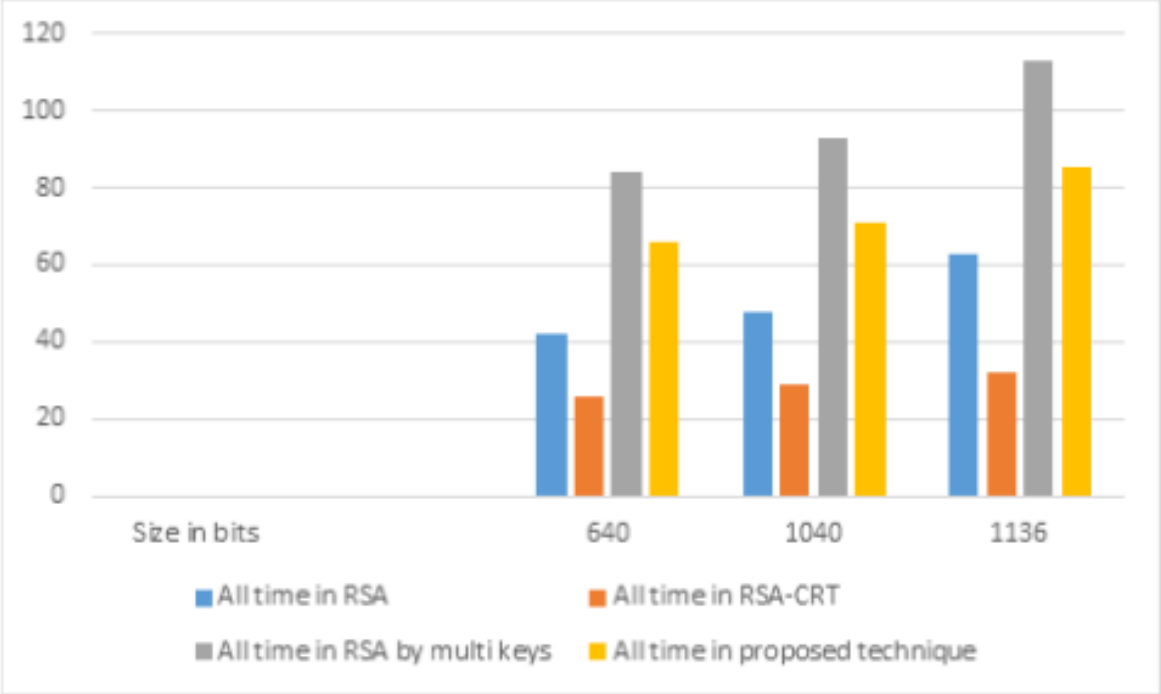


Figure 12 : Time comparison of all RSA algorithm variants

- **Shortcoming of RSA attack**

(i) Brute force attacks

Savage power assaults:

• The initial phase in breaking the private key is to locate the two prime numbers p and q that were increased together to create the modulus n.

• Determining φ(n) given n is comparable to considering n.

• With as of now realized calculations deciding d given e and n seems, by all accounts, to be in any event as tedious as the considering issue.

• Since we would brute be able to constrain break a private key by figuring N, the security of RSA relies upon the huge numbers.

(ii) **Timing attack:**

• A planning assault is fairly comparable to a robber speculating the mix of a safe by seeing what amount of time it requires for somebody to divert the dial from number to number.

• It has been seen that the RSA calculation sets aside various measures of effort to play out its crypto activities as indicated by the key's worth.

• So dependent on the time needed to apply the private key to some data, some gauge can be made of the private key.

• The centrality of this danger expands as indicated by how close an assailant can get to the cycle playing out the crypto activity.

- The timing attack is concerned with the time required to perform the cryptographic operations. This time to operate varies among the inputs as these operations may have unnecessary statements, conditional, branching statements etc. which have different time for execution.
- The attacker uses this difference in the time to make the exact guess of the key. The time samples are collected with different inputs and are fed into a statistical model which guesses the key with an extent to certainty.
- With the measurement of these amounts of time required, the attacker can easily find out the fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems

Other RSA weaknesses:

• While RSA normally experiences unsurprising weaknesses to animal power assaults at key recuperation, its curious numerical nature additionally makes it defenseless against different assaults.

• These incorporate assaults against message classification and assaults against public key age procedures.

• Fortunately, we can by and large shield against numerical assaults by utilizing RSA cautiously.

• The commonsense outcome is that RSA must be utilized sparingly.

Return of Coppersmith's Attack, or ROCA for short is a cryptographic shortcoming in age of RSA keys, that permits the private key of a critical pair to be recouped from the public key.

The key length goes that are viewed as basically factorizable are 512 to 704 pieces, 992 to 1216 pieces and 1984 to 2144 pieces. 4096-piece RSA key isn't easily factorizable now, yet it very well may be conceivable if the assault is improved.

Why we're using asymmetric encryption?

Asymmetric encryption is considered to be more secure than symmetric encryption as it uses two keys for the process. The public key used for encryption is available to everyone but the private key is not disclosed. This encryption method is used in everyday communication over the internet.

Since it uses different keys for both encoding and decoding and key for decoding is only known to owner of the data. That’s the reason any intruder can't decode the data with public key.

An important advantage of asymmetric ciphers over symmetric ciphers is that no secret channel is necessary for the exchange of the public key. The receiver needs only to be assured of the authenticity of the public key. Symmetric ciphers require a secret channel to send the secret key—generated at one side of the communication channel—to the other side.

Asymmetric ciphers also create lesser key-management problems than symmetric ciphers. Only 2n keys are needed for n entities to communicate securely with one another. In a system based on symmetric ciphers, you would need n(n ? 1)/2 secret keys. In a 5000-employee organization, for example, the companywide deployment of a symmetric crypto-based security solution would require more than 12 million keys. The deployment of an asymmetric solution would require only 10,000 keys.

- **Attacks of RSA that can be mitigated using RSA-CRT:**

Mitigation of attacks on RSA-CRT:

Low exponent attack

d is the private exponent in RSA.

Let's see it in detail:

- Given input, m, raise it to the d-th power modulo p and modulo q. The results intermediate is then combined through addition and multiplication with some constant predefined to compute the final result.
- Since the exponentiation modular is performed on half the bit size of n, the execution time is less than 4 times.
- In RSA,

  Decryption, $M = C^d \bmod n$

  Rely directly on size of d and n.

  But in RSA-CRT:

  We calculate:

$$mp = Cp^{dp} \bmod p$$

$$mq = Cq^{dq} \bmod q$$

$$mr = Cr^{dr} \bmod r$$

$$ms = Cs^{ds} \bmod s$$

Now after combining mp, mq, mr and ms, we get back our original plaintext message M.

- In RSA, the exponent decryption d determines the multiplications numbers of modular necessary to implement the exponentiation, and the modulus n decide the size of the result intermediate. Thus, reducing the size of both d and n is considered important advantage in the Chinese Remainder Theorem.

Timing Attack

The timing attack can be mitigated by incorporating **Montgomery Modular Multiplication**. Montgomery multiplication is modular multiplication that allows computing such multiplications faster. Instead of dividing the product and subtracting n multiple times, it adds multiples of n to cancel out the lower bits and then just discards the lower bits.

With RSA-CRT and multiple keys, CRT works similar to modular multiplication. It reduces the size of both d and n, so that the CRT technique improves the throughput rate up to 4 times in the best case where the factors of the modulus n (i.e., p and q) are assumed to be known. By CRT, the computation can be partitioned into four parts, this reduces computation time since (dp, dq, dr, ds) < d and (Cp, Cq, Cr, Cs) < C. In fact, their sizes are about less than half the original sizes.

References

- https://ieeexplore.ieee.org/abstract/document/6311398/

  Yang, K., & Jia, X. (2012). An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE transactions on parallel and distributed systems, 24(9), 1717-1726

- https://www.sciencedirect.com/science/article/pii/S0020025516307319

  Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017). Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, 387, 103-115

- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.391.1412&rep=rep1&type=pdf

  Bindu, B. S., &Yadaiah, B. (2011). Secure data storage in cloud computing. International Journal of Research in Computer Science, 1(1), 63-73.

- http://studentsrepo.um.edu.my/5850/

  Sookhak, M. (2015). Dynamic remote data auditing for securing big data storage in cloud computing (Doctoral dissertation, University of Malaya

- https://ieeexplore.ieee.org/abstract/document/7155804/

  Shimbre, N., & Deshpande, P. (2015, February). Enhancing distributed data storage security for cloud computing using TPA and AES algorithm. In 2015 International Conference on Computing Communication Control and Automation (pp. 35-39). IEEE.

- https://ieeexplore.ieee.org/abstract/document/6141247/

  Behl, A. (2011, December). Emerging security challenges in cloud computing: An insight to cloud security challenges and their mitigation. In 2011 World Congress on Information and Communication Technologies (pp. 217-222). IEEE.

- https://www.sciencedirect.com/science/article/pii/S0020025516306302

  Usman, M., Jan, M. A., & He, X. (2017). Cryptography-based secure data storage and sharing using HEVC and public clouds. Information Sciences, 387, 90-102.

- https://ieeexplore.ieee.org/abstract/document/6781303/

  Garg, P., & Sharma, V. (2014, February). An efficient and secure data storage in Mobile Cloud Computing through RSA and Hash function. In 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (pp. 334-339). IEEE

- https://ieeexplore.ieee.org/abstract/document/7380514/

  Pant, V. K., Prakash, J., & Asthana, A. (2015, October). Three step data security model for cloud computing based on RSA and steganography. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 490-494). IEEE

- https://www.researchgate.net/publication/334603666_Modified_RSA_Algorithm_Using_Two_Public_Key_and_Chinese_Remainder_Theorem

  Rasha Samir Abdeldaym, Hatem Mohamed Abd Elkader, Reda Hussein (Corresponding author: Rasha Samir Abdeldaym) Faculty of Computers and Information BenhaQalubia, El menufia Faculty of Computers and Information Shebein El kom, El menufia Faculty of Computers and Information KafrElshiekh (Email: rashasamir661@yahoo.com, hatem6803@yahoo.com, reda mabrouk@fci.kfs.edu.eg) (Received Oct. 29, 2018; revised and accepted Jan. 6, 2019)

- https://ieeexplore.ieee.org/abstract/document/5679895/

  Somani, U., Lakhani, K., & Mundra, M. (2010, October). Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. In 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010) (pp. 211-216). IEEE

- https://pdfs.semanticscholar.org/9799/a9f9bec6cf85715ca236035b5d89204b326a.pdf

  Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. International journal of engineering research and applications, 3(4), 1922-1926

- https://ieeexplore.ieee.org/abstract/document/6217466/

  Ruj, S., Stojmenovic, M., & Nayak, A. (2012, May). Privacy preserving access control with authentication for securing data in clouds. In 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 556-563). IEEE

- https://ieeexplore.ieee.org/abstract/document/6079468/

  Subashini, S., & Kavitha, V. (2011, October). A metadata based storage model for securing data in cloud environment. In 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (pp. 429-434). IEEE

- https://ieeexplore.ieee.org/abstract/document/5510914/

  Wang, C., Ren, K., Lou, W., & Li, J. (2010). Toward publicly auditable secure cloud data storage services. IEEE network, 24(4), 19-24.

- https://ieeexplore.ieee.org/abstract/document/6642048/

  Zhou, L., Varadharajan, V., & Hitchens, M. (2013). Achieving secure role-based access control on encrypted data in cloud storage. IEEE transactions on information forensics and security, 8(12), 1947-1960

- https://www.cse.wustl.edu/~jain/cse567-06/ftp/encryption_perf/

  Tamimi, A. A. (2008). Performance analysis of data encryption algorithms. Retrieved October, 1.