**PRODUCT KEY ACTIVATION USING ASYMMETRIC CRYPTOGRAPHY**

***A Project Report Submitted by***

Varun Baldev (4NM16CS171)

Vishrut Sharma (4NM16CS179)

Vaibhav K (4NM16CS169)

**UNDER THE GUIDANCE OF**

### Mr. Raghunandan K R

**Department of Computer Science and Engineering**

***In partial fulfillment of the requirements for the award of the Degree of***

***Bachelor of Engineering in Computer Science & Engineering***

***From***

***Visvesvaraya Technological University, Belagavi***



(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC 

08258 - 281039 – 281263, Fax: 08258 – 281265

:

B.E.CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021



**April 2020**



(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

: 08258 - 281039 – 281263, Fax: 08258 – 281265

**Department of Computer Science and Engineering**

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

*Certified that the project work entitled*

*"****Product key activation using asymmetric cryptography****”*

*is a bonafide work carried out by*

***Vaibhav Kalgutkar(4NM16CS169) Varun Baldev(4NM16CS171)***

***Vishrut Sharma(4NM16CS179)***

*in partial fulfilment of the requirements for the award of* ***Bachelor of Engineering Degree in Computer Science and Engineering***

*prescribed by* ***Visvesvaraya Technological University, Belagavi*** *during the year 2018-2019.*

*It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.*

*The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.*

**Signature of Guide Signature of HOD Signature of Principal**

**Semester End Viva Voce Examination**

**Name of the Examiners Signature with Date**

*1.*

*2.*

# ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We express our deep sense of gratitude and indebtedness to our guide **Mr Raghunandan K R,** Assistant Professor Gd II, Department of Computer Science and Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course for our project.

We express our gratitude to the project co-ordinator **Mr. Ranjan Kumar HS** for his constant support.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

### Vishrut S Sharma

### Varun Baldev

### Vaibhav M Kalgutkar

### 

**ABSTRACT**

This paper proposes an improved key verification system to make sure that the product can be used on one system only. Every system has a unique MAC address and that's what we use as the project's foundation. The main contribution of this work is the effective use of a system's MAC address which in turn boosts the security of the project. The basic procedure consists of retrieving a MAC address from the user's system. Now, a serial key is derived by performing specific operations on the MAC address which is then converted into a simpler, user-friendly product key while encryption and decryption takes place during the transmission of keys between client and server. Looking into the future, this work can lead to better and consistent use of MAC address in key generation techniques which is something that hasn't been explored enough.

**Keywords: Asymmetric Cryptography, Serial key, Product key, MAC address**

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **Contents** |  | **Page** |
| Title Page |  | i |
| Certificate Acknowledgement Abstract  Table of contents  List of figures |  | ii iii iv v  vii |

CHAPTER 1: INTRODUCTION 1-8

Overview 1-4

Problem Statement 5

Study Area 6

Objective 6

Motivation 7

Organization of the Report 7-8

CHAPTER 2: LITERATURE SURVEY 9-16

History of Cryptography 9-10

Private Key Cryptography 10

Public Key Cryptography 10-11

RSA Cryptosystem 11-13

RSA Attacks 13-15

Dual RSA 16

Enhancement of RSA 16

CHAPTER 3: SYSTEM ANALYSIS AND REQUIREMENTS 17-21

System Analysis 17-19

Requirements Analysis 20

Functional Requirements 20

Non-Functional Requirements 21

CHAPTER 5: SYSTEM DESIGN 24

High Level Design Architecture 24

CHAPTER 6: SYSTEM IMPLEMENTATION 25-27

Software Approach 25-27

CHAPTER 7: SYSTEM TESTING 28-30

Introduction 28

Unit Testing 28-29

Integration Testing 29-30

CHAPTER 8: RESULTS & DISCUSSIONS 31-35

Simulation 31-32

Graph 33-34

Discussion 35

CHAPTER 9: CONCLUSION AND FUTURE WORK 36

Conclusion 36

Future Work 36

REFERENCES 37-41

vi

# LIST OF FIGURES

Figure 5.1 Block diagram for the proposed system 24

Figure 6.1 Diagram for steps involved in the system 26

vii

# CHAPTER 1 INTRODUCTION

**OVERVIEW**

The problem of illegal product key distribution has caused the rates of piracy to increase exponentially which is why it has become extremely important to tackle this problem. The main objective of this work is to make sure that the product activation key is authentic and exclusive to the user and to a particular system. This is done to ensure that the possibility of redistribution or sharing of a product key is completely eradicated. Asymmetric cryptography techniques are applied to ensure the aforementioned by retrieving the MAC address of the system using which the product has been bought, performing specific mathematical operations on it and then encrypting it so that the key can be activated only on one system only. The idea being to convert MAC address into serial key on the client side which is then encrypted, sent to server side where it is decrypted using a private key. This serial key is then converted to the product key which is sent to the rightful client using the email ID specified at the time during which the product was bought.

**Product Key Activation**

Product key Activation is a verification strategy required by some software to make sure that the product is being redeemed by its rightful owner. Key verification restricts free distribution of duplicated or reproduced software. Un-activated software cannot be used by a customer until the product key specified by the client is verified and validated by the product setup. Activation permits the software to unlock and allow complete utilization of all its tools and functionalities. An activation can keep going on forever, or it can have a time period limit, requiring a renewal or re-activation for proceeding with use.

Normally, the software distributor sends the client a unique serial number. At the point, when the client installs the product it requests the client to get a few more add-ons that the application demands. The application acquires permissions that apply to that client’s permit and a failure to do so might result in a lockdown of the software, rendering it useless. Some activation systems also make use of communication and verification of a key over the internet.

**MAC Address**

It is a unique identifier allotted to a network interface controller (NIC) which allows its use as a physical address during communications inside a network. This utilization is normal in most IEEE 802 networking advancements, including Ethernet, Bluetooth and Wi-Fi. As normally represented, MAC addresses consist of six groups of two hexadecimal digits without a separator, or isolated by hyphens, colons. MAC addresses are allocated by the distributors and generally can be referred to as a physical address.

**Cryptography**

It is a process of verifying information through utilization of codes, algorithms and mathematical calculations with the goal that only those people for whom the information is proposed can access it.

The four pillars of cryptography are defined as follows:-

Integrity: The information cannot be changed in the storage or in the transition between the receiver and the sender without any addition to the information being recognized.

Confidentiality: Information must be delivered to the individual for whom it is intended and none of the other third party or unauthorized users should be able to access that information.

Non-repudiation**:**  
It refers to the ability to ensure that a party or a organization cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

Authentication**:**  
The identity and the origin of the receiver and the sender is pre defined.

**Types Of Cryptography:**

**Symmetric Key Cryptography:** It is an encryption method where message sender and receiver use a common key to encrypt and decrypt messages. Symmetric key systems are quicker and simpler yet the problem is that the transmission of the messages between the sender and the receiver must operate in a secure environment.

Drawbacks of Symmetric Key Cryptography: When somebody gets a hold the symmetric

key, they can decrypt everything encrypted with that key. At the point when you’re utilizing symmetric key cryptography techniques for two-way communication, this implies that information from both sides of the discussion get’s compromised.

**Asymmetric Key Cryptography:** In this type, information is encrypted and decrypted using a two types of keys. For encryption a public key is used, and a private key is used for decryption. There’s a different public key and a private key. Regardless of whether anyone knows the public key, only the rightful receiver can decipher it because only he knows the private key.

Advantage of Asymmetric on Symmetric: In this case if an attacker gets your private key then he or she can decrypt the messages sent to you but he or she can’t access the set of messages sent by you as it uses a different key pair.

Drawbacks of Symmetric Key Cryptography: If an attacker gets the key which is used to encrypt and decrypt he or she can access all the messages as this method utilizes only one key. At the point when you are utilizing symmetric encryption for two-way interchanges, this represents that both ways of the discussion get’s exposed.

**Public key infrastructure**:

One way to deal with such attacks includes the utilization of a public key infrastructure(PKI). A number of approaches, and methodologies expected to make, oversee, circulate, use and deny computerized authentications and oversee public-key encryption. Nonetheless, it has its drawbacks.

Internet browsers, for example, are provided with an extensive rundown of "self-marked identity certificates" from PKI suppliers – these are utilized to check the *bona fides* of the testament authority and afterward, in a subsequent advance step, the authentications of potential communicators.

An assailant who could subvert any single one of those authentication specialists into giving a certificate for a bogus public key could then mount a "man-in-the-middle" assault as effectively as though the certificate scheme were not utilized at all.

In a substitute situation once in a while talked about, an external user who infiltrated an organization's servers and got into its store of certificates and keys (public and private) would have the option to spoof, disguise, decrypt, and produce exchanges at will.

Notwithstanding its hypothetical and potential issues, this methodology is broadly utilized. Models incorporate TLS and its ancestor SSL, which are generally used to give security to internet browser exchanges (for instance, to safely send credit card details to an online store).

Beside the protection from the assault of a specific key pair, the security of the confirmation pecking order must be viewed as while conveying public key frameworks.

At the point when a private key is utilized for certificate creation higher in the PKI server pecking order is undermined, or unintentionally revealed, and then a "man-in-the-middle attack" is conceivable, making any subordinate authentication entirely uncertain.

**Collatz conjecture:**

It is a procedure of conjecture that works as follows: begin with any positive number n. At that point each term is retrieved as follows from the previous term: if the previous term is even, the next term is one half of the previous term. If the previous term is odd, then the following term is 3 times the previous term plus 1. The assumption is that, regardless of the estimation of n, the sequence will consistently arrive at 1.

## PROBLEM STATEMENT

A product key, also known as a software key, is a specific software-based key for a computer program. It certifies that the copy of the program is original. The current system only has this opportunity for the English medium students explaining their syllabus.  [Activation](https://en.wikipedia.org/wiki/Product_activation) is sometimes done offline by entering the key, or with software like [Windows 8.1](https://en.wikipedia.org/wiki/Windows_8.1), online activation is required to prevent multiple people using the same key. Computer games use product keys to verify that the game has not been [copied without authorization](https://en.wikipedia.org/wiki/Copyright_infringement_of_software).

Likewise, one is not allowed to play online with two identical product keys at the same time. With the emergence of high-quality hackers, security in today's society is no more optional but instead it's a major requirement. Illegally distributed and generated product keys are available and being used casually as a result of weak encryption algorithms and open source software. This is why, the main aim of our project is to provide a solution to this crisis by making use of complex and secure algorithms to generate product keys.

## STUDY AREA

Learning field is changing rapidly due to different methods and technologies that are introduced so often. It is important to have a safer, more secure method of generating product keys. Hackers and malicious users are constantly re-inventing themselves which gives us higher responsibility to stay ahead of the curve and provide better, efficient software. Security is the primary objective, although the needs of the user must not be neglected and lost during the process. That is, the process can be complex but the execution must be simple so that it can be used easily by most clients.

Reduction in costs, Flexibility in algorithms, Complexity of the operations, Ease of development, regular modification and updating of the current system, Tracking the progress with immediate feedbacks are some of the are immediate fields of study.

## OBJECTIVE

The main objective of this project is to make sure that the product activation key is authentic and exclusive to the user and to a particular system. This is done to ensure that the possibility of redistribution or sharing of a product key is completely eradicated.

Application of asymmetric key cryptography to ensure the aforementioned by acquiring the MAC address of the system using which the product has been bought and then encrypting it so that the key can be activated only on one system.

Some of the primary objectives are: -

* Product keys are very important for selling any software products.
* Since not many people will not have access to a CD tray nowadays the product will need to be sold online.
* To make sure the person who paid gets the product the product key is sent to the consumer securely.
* Product key authentication is vital to prevent product counterfeiting, which is the usage of the same product on a number of machines greater than as permitted by the company.

## MOTIVATION

Standard key generation, where product keys are generated mathematically, is not completely effective in stopping copyright infringement of software, as these keys can be distributed.

In addition, with improved communication from the rise of the internet, more sophisticated attacks on keys such as cracks (removing the need for a key) and product key generators have become common. Because of this, software publishers are increasingly turning to alternative methods of verifying that keys are both valid and uncompromised.

One method, product validation, assigns a product key based on a unique feature of the purchaser's computer hardware (such as its MAC address), which cannot be as easily duplicated since it depends on the user's hardware.

## ORGANIZATION OF THE CHAPTERS

The project report has been organized under nine chapters, which are as follows:

**Chapter I:** Introduces to the main principles of the project. It represents information about the aim and methodology of the same.

**Chapter II:** It includes literature survey of related works.

**Chapter III:** Discusses the system requirements that are needed for the project. These include functional requirements, non-functional requirements, user requirements and hardware requirements.

**Chapter IV:** Includes the system design details which includes flowchart, sequence diagram.

**Chapter V:** Includes the implementation details of the project, application is explained in detail. It also deals with software approach.

**Chapter VI:** Deals with system testing concepts and the various test cases for the project.

**Chapter VII:** Includes the screenshots of the application and the database.

**Chapter VIII:** Discuss the results of the project.

**Chapter IX** outlines conclusions and future work that can be done

# CHAPTER 2

# LITERATURE SURVEY

# History of Cryptography

Edge C. et.al [1] word cryptography is derived from the Greek words kryptos, meaning hidden and grafein meaning to write. Mohammed Abdalbasit et.al [2] states that Cryptography is a method to accomplish the privacy of messages. The term has a particular meaning in Greek: "secret writing". These days, in any case, the protection of privacy of people and associations is achieved through cryptography at a significant level, ensuring that data sent is secure such that the approved receiver can access this data. Hromkovic Jurav et.al [3] address the fact that in cryptography the message can be sent from the sender and only those receivers who are given the permission to access it can read it. Haunts S. et.al [4] states that in olden times the messages which had to be transmitted from one location to other were made unreadable by using some mode of conversion such as mathematical or any other. This paper also discusses about the evolution of cryptography from the olden times to this modern era. O’Regan G. [5] discusses cryptography, which is a primary application of number theory. After the Second World War, the code breaking work undertaken at Bletchley Park in England is discussed and the fundamentals of cryptography, including private and public-key cryptosystems, are addressed. O’Regan G.[6] also discusses about the various mathematical computation used in the conversion of plaintext into cipher text in this paper. This paper also discusses about the history and evolution of computations used in the conversion of the plaintext from cipher text. Zapechnikov S et.al [7] divides the history of cryptography into the following

* Pre-scientific period (from ancient times to the end of the 17th century)
* Classical science period, in which the scientific basis of cryptography was formed (18th–19th centuries)
* Transition phase from classical to modern science (the end of the 19th century – the beginning of the 20th century)
* Modern cryptography (the second half of the 20th century – the beginning of the 21st century).

Borda M [8] states that in ancient times also information transfer from one point to another was very important. For example in Spartan military a message was tied to a stick and then it was rolled by a ribbon, the person with the same stick could only read the message.

Saurabh Sharma et.al [9] states the new innovations that took place in the field of cryptography. Public key cryptography was such a innovation that took place. It solved many security issues but also it didn’t bring a solution to the key management problem.

**Private Key Cryptography**

Yan S.Y [10] states that private key cryptography is a type of cryptography that came into working 5000 years ago, it is a type of cryptography which uses the same key for encryption and decryption of the message between the sender and receiver.

**Public Key Cryptography**

Delfs Hans et.al [11] proposed that public key cryptography is an encryption method where message sender and receiver utilize two different keys to encrypt and decrypt messages. In a public-key encryption scheme, the communication partners do not share a secret key. Each user has a pair of keys: a secret key known only to him and a public key known to everyone. Public key cryptography is much more secure and safe than others. Boyd Colin et.al [12] proposes the use of the public key cryptography in the authentication and the key transport advantages carried by the public key cryptography systems. Every cryptography system will have a complexity factor Barak Boaz [13] surveyed all the computational foundations for the public key cryptography. The computational assumptions made are also discussed in this paper. There are different public key cryptography algorithms Ahmad Jasmin et.al [14] catogorizes the algorithms into Discrete Logarithm, Integer Factorization, Coding Theory, Elliptic Curve, Lattices, Digital Signature and Hybrid algorithms. This paper also presents the advantages and drawbacks of the algorithms.

The public key cryptography is very safe and secure but the problem is that it is slower as compared to private key algorithm. G.B. Agnew et.al [15] proposed a public key cryptography system which deals with increasing the speed of the cryptography system .In this paper a VLSI implementation is created for making an exponentiation for cryptographic of approximately 300 kilobits per second.

**Applications of Public Key Cryptography**

K.Shim [16] surveys the use of public key cryptosystem in the Wireless Sensor Networks(WSN’s). This paper discusses about the implementation results of public-key cryptographic primitives in terms of execution time, energy consumption andresource occupation.

M. Malik et.al [17] focuses on Key bootstrapping protocols based on public key cryptography in the Internet of Things. This paper discusses about the use of public key cryptography in the secure transmission of the messages between the devices, mobiles,

TV’s etc. in an Internet of Things environment.

Ahmad F et.al [18] proposed that the implementation of various forgery techniques will counterfeit the printed domain documents. This paper suggests verification of the authenticity of printed documents and their origins using digital signature based on print-scan-resistant hashing of images and cryptography with public key.

**RSA cryptosystem**

Rubinstein Slzedo S. [19] states RSA cryptosystem consists of mainly two parts namely key generation and encryption. The RSA cryptosystem is set up by first multiplying two very large prime numbers and then the product n is made public. The product n is part of the public key, and the factors of n are kept secret and are used as the secret key. The main idea because of which this algorithm works is that the factors of n cannot be recovered from n. The security of RSA depends on the tremendous difficulty of factoring. Nicholas Tuzzio et.al [20] states the RSA algorithm has two different parts Key generation and encryption. The Key generation consists of RSA cryptosystem has two different pair of keys, one is the public key and the other is the private key. The key generation consists of three steps. Large distinct primes p and q are chosen randomly, and which thus computes n = p\*q. Then an integer is chosen, 1< e < φ(n),which is prime to φ(n). Then (n, e) is published as the public key. Then d is computed with ed≡1 mod φ (n). (n, d) is his secret key.

The encryption of the messages is done by the public key which is generated and the decryption is done by the secret key which is generated after the key generation process .The encryption is done by n which is the product of two large distinct prime numbers which are randomly selected.

The security factor provided by the RSA cryptosystem depends on the prime numbers used, Childs L.N et.al [21] construction of an RSA cryptosystem involves large prime numbers, s**o** this paper is dedicated to seeing how many large primes there are, and how to classify large primes with high certainty.

# To improve the security of the RSA cryptosystem Das S.B et.al [22] proposes a new approach to asymmetric cryptography, i.e. modified algorithm RSA. This proposed approach reduces RSA algorithm involving complex calculation. A novel algorithm was proposed to find the value of both the public key (k) and the private key . In addition, the ciphertext (N) and plaintext (M) are identified using a newly developed formula. They have been doing some important work in the field of RSA algorithm.

# Properties of RSA

# Katzenbeisser S a.e.t [23] discusses the most important properties of RSA cryptosystem they are The RSA algorithm is very efficient for the encryption of the messages and this algorithm is also less efficient for the decryption of the messages. To improve the security of the messages the keys sometimes have to be lengthy. The RSA algorithm is also widely used among many users and is renowned for it’s security features.

**Applications of RSA**

The following are the different applications of the RSA cryptosystem used in the transmission of data

# Aniruddha Bhattacharya et.al [24] proposes a Secure Hybrid RSA (SHRSA), The decryption of SHRSA is much safer and more effective than the RSA and Chinese Remainder Theorem (CRT)-RSA. It provides users of the messaging scheme with a complete privacy. The communication protocol for the SHRSA cipher solves several RSA-related problems. This also uses much less memory than RSA and CRT-RSA, and far less CPU. So high security, decryption efficiency with less memory and features of CPU occupancy make this secure message communication protocol more applicable to the Internet.

# Liu B et.al [25] examines function of asymmetric encryption technology and thorough presentation of the RSA algorithm theory is also studied. Montgomery's quick modular multiplication algorithm was adopted as the central part of the RSA encryption algorithm, despite the complexity of division in hardware.

# M. Fueyo et.al [26] concentrates on the RSA-based environment, and consider blacklists L = LB as case of non-membership proofs. Current solutions for this setting are focused on the use of universal dynamic accumulators; the underlying zero-knowledge proofs are bit complicated, and hence their efficiency; but independent of the size of the blacklist L, it seems improvable.

# Zou L et.al [27] proposed a hybrid encryption algorithm combining AES and RSA algorithm is proposed in this paper to solve the above-mentioned problems to solve the file encryption performance and security issues. The experimental results show that the RSA and AES hybrid encryption algorithm can not only encrypt files, but also have the efficiency and security benefits of the algorithm.

# Shaheen B et.al [28] proposes different security algorithms were discussed in this paper in order to analyze the efficiency of the algorithms and to find out the encryption algorithm is better for data protection in the cloud computing platform. The algorithms mostly concerned in this paper are RSA Algorithm and RSA Encryption Algorithm Modification.

**RSA Attacks**

# The attacks on the RSA cryptosystem are performed to retrieve the message from the transmission medium by decoding the encryptions of the RSA cryptosystem.Yan S.Y. [29] states the computational preliminaries used in the attacks on RSA. Dubey M.K et.al [30] presents a brief summary on RSA, discuss various flaws and cryptanalytic attacks including genetic algorithm applicability and some countermeasures to overcome from some flaws and cryptanalytic attacks. The analysis study shows that RSA is the most common stable asymmetric cryptosystem, and that its strength will remain intact until quantum computers are available. Different attacks on RSA and the variations of these attacks are listed below

[Abderrahmane Nitaj](https://www.semanticscholar.org/author/Abderrahmane-Nitaj/2597818) a.e.t [31] proposed an attack in which let N = pq be an RSA module, i.e. the sum of two major unknown primes of the same bit size. In this paper, they define an assault on RSA in the presence of two or three exponents ei with the same modulus N and satisfying equation eixi−ϕ(N)yi=zi where unknown parameters are ϕ(N)=(p−1)(q−1) and xi, yi, zi.

Atushi Takayasu a.e.t [32] tries to formulate general attack scenarios to capture several existing ones and propose attacks for the scenarios. These attacks capture general exposure scenarios, the findings can be used as a tool kit. The protection of some potential RSA variants can be investigated without understanding the method of Coppersmith.

Jean- s´Ebastein Coro a.e.t[33] proposes that against plain RSA encryption a chosen-ciphertext attack was described by Desmedt and Odlyzko at Crypto ’85 . The technique can also be applied to RSA signatures and enables an existential forgery under a chosen-message attack. Until a change in the technique in this attack the potential of this attack remained untapped. This change in the technique made it effective against two very popular RSA signature standards, namely iso/iec 9796-1 and iso/iec 9796-2. Following these attacks, iso/iec 9796-1 was withdrawn and iso/iec 9796-2 amended.

Atushi Takayasu a.e.t [34] proposes enhanced partial main exposure attacks with multiple exponent pairs on RSA. Our results solve the drawbacks set out above fully. These attacks are the first results for large exponents and our attacks for n=1 lead to the strongest attacks on the standard RSA which are currently known. These results are superior to previous results for small hidden exponents when n=1 and 2, and when n=3 and d1,…,dn>N3(n−1)/(3n+1).

Stelvio Cimato a.e.t [35] analyzes the feasibility of partial key exposure when taking a countermeasure for side-channel attacks. They consider in particular the technique of exponent blinding, which consists of randomizing the private exponent at each execution. This review is addressed to both RSA and CRT-RSA, including theoretical evidence and experimental tests.

Andrej Dujella et.al [36] suggests a new version of Wiener's attack using results on Diophantine approximations of the form − p / q < c / q 2, and the "meet-in - the-middle" version to test candidates (from the form rq m+1 + sq m) for the hidden exponent. This reduces the Attack's run-time complexity to O(D log D) (with O(D) space complexity).

Werner Schinder et.al [37] introducestwo new attack algorithms with CRT on RSA which significantly improve the effectiveness of the attack. In particular, attacks on length R=64 blinding factors have undoubtedly become realistic, and even R=96 can be overcome for small error levels *ϵb*

Liqiang Peng et.al [38] proposes partial key exposure attacks for both the standard RSA with N = pq module and the Takagi's version of RSA with N = p 2 q module when most significant bits (MSBs) or least significant bits of q are revealed. Compared to previous research, this theoretical study and experimental findings indicate a major increase in reducing the number of known private key bits to factor N.

Jain M et.al [39] proposes Over the popular RSA framework, powerful non-pairing schemes have been built to bridge the practical gap between actual public key ciphers and the utility of IBE within it. But in these proposals the security risks to RSA were maintained. This paper introduces an IBE scheme based on RSA that overcomes RSA security problems and requires linear computations, and allows key management easy.

Blinding by the exponent has been commonly believed to avoid any timing attack on RSA. Werner Schindler et.al [40] extends the aforementioned attacks to RSA with CRT at the expense of slightly more timing calculations when Montgomery's multiplication algorithm and exponent blindness are applied. Simulation experiments will be carried out which will confirm the theoretical results. The countermeasures are successful.

# Alexander May et.al [41] proposes RSA-type schemes with modulus N = p r q for r ≥ 2 are considered. They are implementing two new attacks for small secret exponent d. All methods are applications of the Coppersmith method to solve modular polynomial univariate equations. They explicitly derive partial key disclosure attacks from these latest attacks, i.e. attacks where the hidden exponent is not actually low but when the attacker is aware of a fraction of the secret key bits.

Vineet Kumar et.al [42] presentsparallel approach to the common factor attack on RSA moduli obtained from the Internet by mining TLS and SSH certificates. They propose a data-parallel routine to efficiently exploit the batch-wise GCD algorithm in a resource-constrained environment, and install the common factor attack on TLS and SSH certificates to get a fairly accurate collection of vulnerable RSA moduli.

# Zheng M et.al [43] proposes an implicit related factorization problem on the RSA cryptosystem. Informally, they investigate under what condition, given the implicit knowledge of relevant private keys, RSA moduli can be effectively factored in polynomial time. A generic attack is subsequently proposed using a heuristic lattice construction when more instances of RSA are given. In addition, they carry out numerical tests to test the validity of the proposed attacks.

Oded Yacobi et.al [44] shows that given e cryptograms c i (a i x+b i) e mod N, i=0,1, ... e–1, one can determine x in O(e) Z N -operations depending on the cryptograms, after a pre-processing which depends only on the constants for some known constants. The pre-processing complexity is O(elog 2 e) Z N -operations, and can be computed in several instances.

# Dual RSA

# Dual RSA states the generation of two distinct pair of keys. The same Encryption and the same Decryption Key is used in the RSA Key pair. That is why this type of system is known as dual RSA. It has advantages such as it reduces amount of storage required for the key.

# Applications of Dual RSA are

# J.Anith et.al [45] uses the dual RSA in storing the data securely and also intrusion detection. Steganography techniques are used after dual encryption to boost the storage security of the proposed system. Their proposed framework is implemented in the Java working platform with the support of Cloud simulator.

# Thakur S.S et.al [46] proposed that Dual RSA Based Secure Biometric System for Finger Vein Recognition in which the Dual RSA cryptosystem is used in the transmission of the information of the Finger Vein of the registered user from one system to another in a secure manner.

# A.Goel et.al [47] proposed that if the user can factor modulus (n) into its prime numbers in traditional RSA algorithms then they can generate private key. So to eliminate this weakness the users can use dual modulus and significantly improve device security.

**Enhancement in RSA**

The enhancements made in the RSA cryptosystem to improve the security and authentication of the message transmitted. Muhammad N et.al [48] proposes the uses of identity in the enhancement of RSA Key Generation. This algorithm uses the users identity to replace the numbers that are used as a public key in the RSA algorithm.This enhancement of the RSA algorithm can be applied in network security. Dutta S.C et.al [49] proposes an improvement in RSA algorithm for the increased security enhancement in mobile ad hoc network and Roy D et.al [50] proposes a modified RSA algorithm called MRSA with three prime numbers j, k, and l for VANETs to enhance vehicle communication safety. RSA algorithm is used with one more prime number, i.e. three prime numbers are used instead of two prime numbers, to improve the duration of the brute force attack.

# CHAPTER 3 SYSTEM ANALYSIS AND REQUIREMENTS

## SYSTEM ANALYSIS

### Relevance of Platform

In this project we use the IDLE python development tool. IDLE is Python’s Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously.

IDLE has the following features:

* coded in 100% pure Python, using the tkinter GUI toolkit
* cross-platform: works mostly the same on Windows, Unix, and Mac OS
* Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
* multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
* search within any window, replace within editor windows, and search through multiple files (grep)
* debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
* configuration, browsers, and other dialogs

IDLE is very easy to use and simple software. Due to the ease of use we decided to use IDLE as our IDE to create our project. One of the features that make it easy to use is automatic indentation. After a block-opening statement, the next line is indented by 4 spaces (in the Python Shell window by one tab). After certain keywords (break, return etc.) the next line is dedented. In leading indentation, Backspace deletes up to 4 spaces if they are there. Tab inserts spaces (in the Python Shell window one tab), number depends on Indent width.

Another feature is completions. Completions are supplied for functions, classes, and attributes of classes, both built-in and user-defined. Completions are also provided for filenames. If there is only one possible completion for the characters entered, a Tab will supply that completion without opening the ACW. ‘Show Completions’ will force open a completions window; by default the C-space will open a completions window. In an empty string, this will contain the files in the current directory. On a blank line, it will contain the built-in and user-defined functions and classes in the current namespaces, plus any modules imported. If some characters have been entered, the ACW will attempt to be more specific.

### Relevance of Programming Language

**Python** is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.  
In Python, we don’t need to declare the type of variable because it is a dynamic typed language.

There are many features in Python, some of which are discussed below –

**Easy to code:**   
Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

**Object-Oriented Language:**   
One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation etc.

**GUI Programming Support:**   
Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

**High-Level Language:**   
Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

**Python is Portable language:**  
Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, UNIX and Mac then we do not need to change it, we can run this code on any platform.

**Python is Integrated language:**  
Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

**Interpreted Language:**  
Python is an Interpreted Language. Because python code is executed line by line at a time like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **byte code**.

**Dynamically Typed Language:**  
Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don’t need to specify the type of variable.

## REQUIREMENT ANALYSIS

### Scope and Boundary

This project has a lot of scope in the real world. This project will enable safe purchasing of software products or other data due to its strength of encryption. The main application of this project is to generate a product key and send it to the user that purchased the product. Some of the other applications of this project are: encrypting coordinates, encrypting aadhar card number or any other crucial data. The double encryption of the data allows for even more secure transmission and there is a very low chance for attackers to get their hands on the data being transmitted.

Owing to the evolution of education through the years, institutions have installed systems in classrooms that provide visual learning for better understanding of the students. This form of learning has proven to be more effective than traditional and hence more and more educational institutions are inculcating visual based learning. It also focuses on students who cannot physically be present in schools and tries to provide them the facilities at their homes. Accessing the lecturers that are provided online is possible on a system sitting at home rather than having to travel miles to reach schools and attend classes. E- learning tries to save the students’ time and they do not have to entirely spend their schedules in schools instead can divide their time for work and learning equally.

The users of our software mostly being students would want the platform to be more engaging and interactive. The platform should be visually pleasing so that the students find it more interesting to work on and makes their learning process fun and productive.

## FUNCTIONAL REQUIREMENTS

### Software Requirements:

Software Requirements depends on the type of software, expected user and type of system where the software is used.

* IDLE python development tool

## NON-FUNCTIONAL REQUIREMENTS:

## Reliability

The program must produce accurate product keys and the user must always receive accurate product keys according to their MAC address. The program must not access the users MAC address without the user’s explicit permission to do so. The program will only access the MAC address after the user has completed the process of purchasing the product.

**Availability**

As long as the product’s purchase page is online the program will be available to generate and send product keys to the user.

**Security**

This application makes sure that malicious users cannot access the MAC address or the product keys of legitimate users. First, the program converts the MAC address to a serial key which is done using the collatz conjecture method. Then we encrypt this serial key with RSA to make it more secure. Therefore, unauthorized users will not be able to access the information easily. To add to the security the product key generated is sent only to the email address of the user that purchased the product.

# CHAPTER 4

## HIGH LEVEL DESIGN ARCHITECHTURE

**SYSTEM DESIGN**

The initial stage involves the conversion of MAC address to serial key using the collatz conjecture sequence. Then this serial key is encrypted using RSA. The Cipher formed after this is sent from client side to the server side where it is decrypted. The product key is then generated by applying the RSA formula on the serial key. The difference here is that

the n value does not need to be a prime number, it can be any random number and it is significantly smaller than the serial key which means that we can control the size of the product key depending on the size of n. In our existing build we have specified n as a random 10 digit integer. This means that the product key will always be 10 digits or lower.

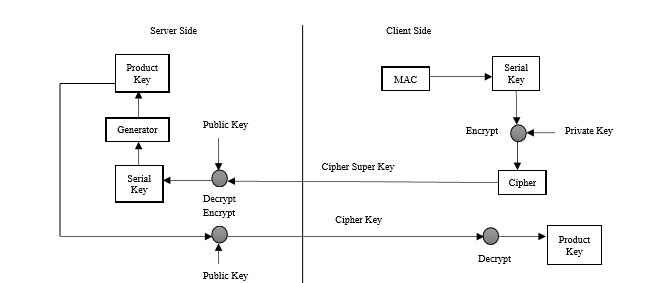


Figure 5.1: Block diagram of the proposed system

# CHAPTER 5 SYSTEM IMPLEMENTATION

## SOFTWARE APPROACH

The following steps are involved in the creation and verification of the product keys.

**STEP 1**: Converting MAC address to Serial Key

The MAC address of the system is retrieved by the program which is then converted to serial key using a sequence called the collatz conjecture. Here every bit of MAC address is applied to collatz conjecture to produce a serial key. If the number is even, the bit is divided by two. If the number is odd, triple it and add one. The value obtained after this step is the serial key generated.

**STEP 2**: Encrypting the Serial Key

Once the serial key is generated, we make use of the RSA algorithm to encrypt it utilizing the server’s public key. RSA algorithm is an asymmetric key cryptography algorithm which is used to encrypt and decrypt messages. The public key is known to everyone and is used to encrypt the messages. The messages encrypted by the public key can be decrypted by the private key.

**STEP 3**: Decrypting the Serial Key

After the encrypted serial key has been received by the server, it is then decrypted by utilizing the RSA algorithm i.e. the server’s private key is used. The serial key obtained is then to be converted into the product key.

**STEP 4**: Serial Key to Product Key Conversion

The serial key is converted into the product key by making use of the RSA formula. During the conversion process, the cipher text (serial key) received by the server is in-fact considered as the plain text (message). Only difference being, the value of ‘n’ used here is significantly smaller than the value of n used during the encryption process. This is done in order to reduce the size of the generated product key.

**STEP 5**: Generated product key is sent to the client via email

The product key generated is sent to the client using the email ID provided by them during the time of transaction.

**STEP 6**: Verification of Product Key received by the client

After the product key is received by the client, they enter it into the website. Authenticity of the entered product key is checked by the backend of the website, making sure that the product key is new and unique for each user. That is, making sure the product is being redeemed by the same system on which it was originally bought by checking the target system’s MAC address.

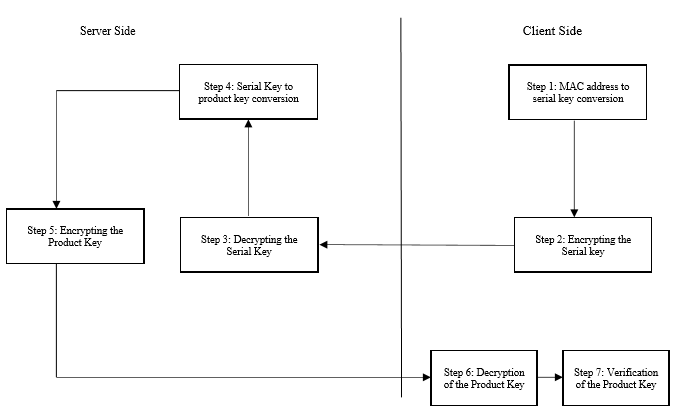


Figure 6.1 : Steps involved in Product Key Activation using Asymmetric key Cryptography

**Example**: The MAC address retrieved is 98:22:ef:70:51:90. This MAC address is then converted to serial key using a sequence called collatz conjecture. The serial key retrieved after this step is:

9281472211341752261340201051684212121147221134175226134020105161546237035106531608040201051672211341752261340201051605161928147221134175226134020105160.

RSA algorithm is used to encrypt the serial key using the server’s public key. This encrypted serial key is then decrypted at the server’s side using the server’s private key. After the serial key is decrypted at the server’s side it is converted to product key using RSA's encryption algorithm (Using a smaller 'n' value to reduce the size of the key). The product key generated after this step is 8002281480558766684. This product key is then sent to the client via email.

# CHAPTER 6 SYSTEM TESTING

## INTRODUCTION

Software testing is a process used to identify the correctness, completeness and quality of the developed software. Testing is the process of questioning a product in order to evaluate it, where the questions are things the tester tries to do with the product and the product answers with its behavior in reaction to probing of the tester.

Testing phase is performed after coding to detect all the errors and provide quality assurance and ensure reliability of the software. Testing is vital to the success of the system. During testing, the software to be tested is executed with a set of test cases, and the behavior of the system for the test cases is evaluated to determine if the system is performing as expected. Clearly the success of testing in revealing errors depends critically on the test cases. [16]

## UNIT TESTING

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

The benefits of Unit Testing are:

* Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
* Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.
* Development is faster. If you do not have unit testing in place, you write your code and perform that fuzzy ‘developer test’ (You set some breakpoints, fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) But, if you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than ‘developer tests’. Development is faster in the long run too. The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing.
* The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Compare the cost (time, effort, destruction, humiliation) of a defect detected during acceptance testing or when the software is live.
* Debugging is easy. When a test fails, only the latest changes need to be debugged. With testing at higher levels, changes made over the span of several days/weeks/months need to be scanned.
* Codes are more reliable. I think there is no need to explain this to a sane person.

## INTEGRATION TESTING

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Definitions of integration testing are Integration Testing**:**

* Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

Component Integration Testing**:**

* Testing performed to expose defects in the interfaces and interaction between integrated components.

System Integration Testing**:**

* Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

# CHAPTER 7 RESULTS AND DISCUSSSIONS

The results generated by using the proposed methodology are as follows:-

**1. Simulation**



Image 1 shows the server's side simulation of the proposed methodology and the final product key which is generated on the server's side.

First, the MAC address of the target system is retrieved by the program and converted into a serial key using the collatz conjecture algorithm.

This serial key is then encrypted using RSA's encryption formula: -

Once encrypted, the cipher text is sent to the server and decrypted using RSA's decryption formula: -

Finally, the decrypted text is then used to generate a product key by using RSA's encryption formula again. The difference here being, the 'n' value used during this process is a lot smaller which results in the length of the product key being significantly smaller and user friendly.

*.*

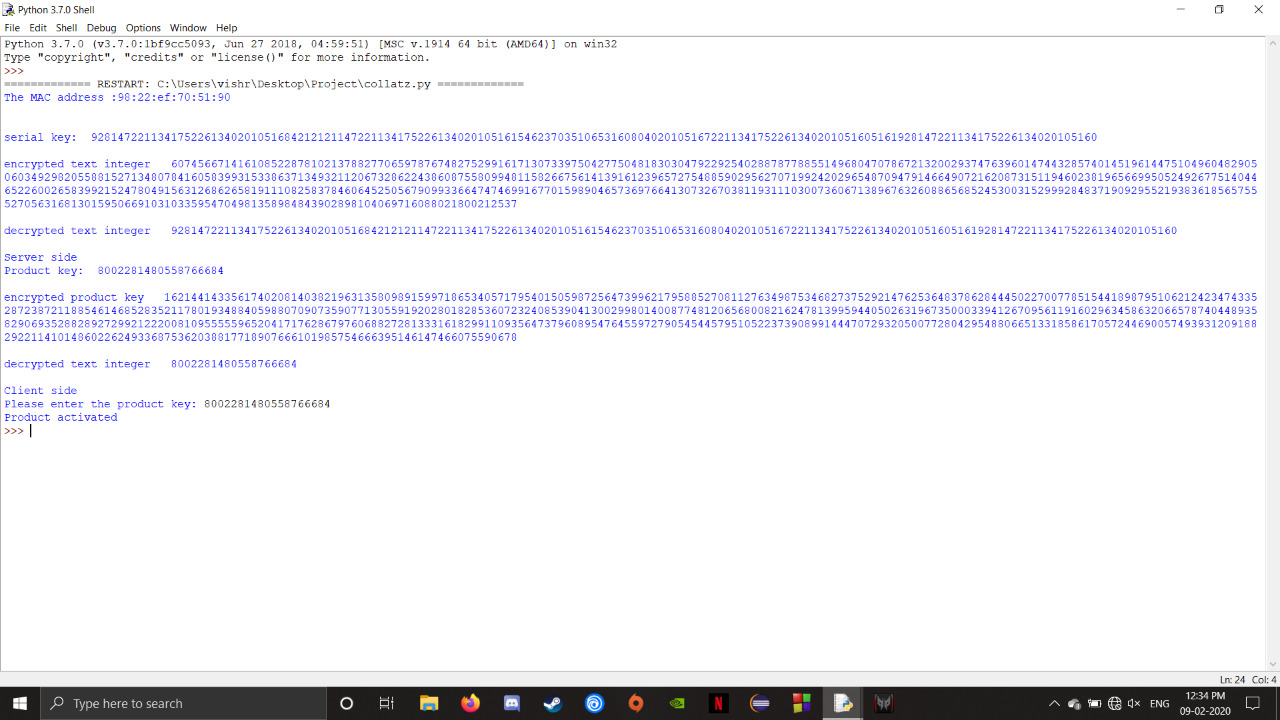


Image 2 represents the simulation on the client's end when he/she wishes to activate their product and enters it into the website to redeem it.

Here, the customer's job is very simple and stress free. All they have to do is enter the product key received by them(via email) and enter onto the website. Once done, the backend of the website will check if the product key is being entered from the system that was originally used to buy the product and also makes sure that a particular product key is used only once.

If the key entered is valid, the product is unlocked and can then be redeemed by the customer.

In case the product key is entered incorrectly three times by the user, this results in a lockdown of the product for that particular system as an extra layer of security.

**2.Comparison Graph as a measure of security**

Consider an example where:-

Original MAC address: 98:22:ef:70:51:90 and Product key:8002281480558766684

Now, if each bit of the MAC address is changed(One bit at a time) starting from the rightmost bit. The product key generated as a result of this is completely different to the product key generated from the original MAC address.

Consider,

Bit changed(right to left): Product Key:

1st (98:22:ef:70:51:91) 81468231738262776011

2nd (98:22:ef:70:51:a0) 55410159012135933914

3rd (98:22:ef:70:52:90) 32247282312834519690

4th (98:22:ef:70:61:90) 70613806947959177858

5th (98:22:ef:71:51:90) 6509906189986885886

6th (98:22:ef:80:51:90) 51982026067146377020

7th (98:22:ee:70:51:90) 65112653651591448890

8th (98:22:ff:70:51:90) 37965365883336520728

9th (98:23:ef:70:51:90) 57034371028485901328

10th (98:32:ef:70:51:90) 46217223073107650546

11th (99:22:ef:70:51:90) 362788305703657816

12th (a8:22:ef:70:51:90) 36834689788535930876

Graph shows the comparison between the original product key and the modified product key.

As stated earlier, graph shows the difference in product key when each digit of the MAC address is changed(One bit at a time) starting from the rightmost bit. This means that every time a bit of the MAC address is changed, the product key generated as a result of the new MAC address is significantly different to the product key generated from the original MAC address.

This ensures that there are no patterns developing during the generation of a product key and that the product keys are as authentic as possible.

The graph is plotted according to the sizes of the newer product keys generated from the modified MAC address against the original MAC address which is retrieved initially.

This shows that there are no particular areas from which one can attack the MAC address in order to unethically generate a customised product key.

## DISCUSSIONS

The main purpose of our project is to boost the security of product keys being used to redeem products online. This is done by retrieving a system's MAC address and performing operations on it in order to generate a unique customer and system specific product key. For this purpose, we have based our entire project around the MAC address since it is unique and specific to a particular system and thus acts as a good identifier. The operations being performed on the MAC address are done using some of the well known algorithms such as collatz conjecture and the RSA algorithm is used to send the product key from the client to the server's side. Once the customer enters their product key into the website and it is validated, their product is then unlocked and can be redeemed by them at any time. In case a customer fails to enter the right product key after trying three times in a row, then the product is locked for that particular system and thus cannot be removed on that system anymore. This acts as an extra layer of security to prevent dictionary, brute force attacks or bots which may try to attack our system.

# CHAPTER 8 CONCLUSION AND FUTURE WORK

## CONCLUSION

Public key cryptography helps in the secure transmission of the message and adds greater levels of encryption which makes it difficult for the third party to access the messages.

The use of RSA algorithm in the encryption and decryption process between the client and the server during the transmission of messages also increases the security of the system. Adding different levels of encryption in the form of conversion of MAC Address to serial key then conversion of the serial key to product key makes it more difficult for the attacker to retrieve the message thus making the system more secure. This project because of its different levels of encryption makes it more secure in transmission of the key from the client side to the server side. Another advantage of the project is that it uses the MAC Address of the client side to form the serial key which is then converted into product key on the server side; this use of MAC Address is an advantage because the MAC address is different for different systems thus improving the authenticity of the system.

## FUTURE WORK

The system that we have currently developed has multiple features that benefit the product by adding the layers of security and also authenticity. Further we want to focus more on making the system more secure and adding more levels of encryption to the system. The generation of the product key also can be improved by using more than two types of algorithms in conversion from MAC Address to Serial Key and then to Product Key which will make the generation of the product key more complex.

# REFERENCES

1. Edge C., O’Donnell D. (2016) Introduction to Cryptography. In: Enterprise Mac Security. Apress, Berkeley, CA
2. A Review Paper on Cryptography - Mohammed, Abdalbasit & Varol, Nurhayat. (2019). 1-6.10.1109/ISDFS.2019.8757514.
3. Hromkovič J. (2009) Cryptography, or How to Transform Drawbacks into Advantages. In:Algorithmic Adventures. Springer, Berlin, Heidelberg
4. Haunts S. (2019) A Brief History of Cryptography. In: Applied Cryptography in .NET and Azure Key Vault. Apress, Berkeley, CA
5. O’Regan G. (2020) Cryptography. In: Mathematics in Computing. Undergraduate Topics in Computer Science. Springer, Cham.
6. O’Regan G. (2016) Cryptography. In: Guide to Discrete Mathematics. Texts in Computer Science. Springer, Cham
7. Zapechnikov S., Tolstoy A., Nagibin S. (2015) History of Cryptography in Syllabus on Information Security Training. In: Bishop M., Miloslavskaya N., Theocharidou M. (eds) Information Security Education Across the Curriculum. WISE 2015. IFIP Advances in Information and Communication Technology, vol 453. Springer, Cham
8. Borda M. (2011) Cryptography Basics. In: Fundamentals in Information Theory and Coding. Springer, Berlin, Heidelberg
9. Sharma S., Mishra N.K. (2011) New Innovations in Cryptography and Its Applications. In: Mantri A.,Nandi S., Kumar G., Kumar S. (eds) High Performance Architecture and Grid Computing. HPAGC 2011. Communications in Computer and Information Science, vol 169. Springer, Berlin, Heidelberg
10. Yan S.Y. (2019) Secret-Key Cryptography. In: Cybercryptography: Applicable Cryptography for Cyberspace Security. Springer, Cham
11. Delfs H., Knebl H. (2015) Public-Key Cryptography. In: Introduction to Cryptography. Information Security and Cryptography. Springer, Berlin, Heidelberg
12. Boyd C., Mathuria A., Stebila D. (2020) Authentication and Key Transport Using Public Key Cryptography. In: Protocols for Authentication and Key Establishment. Information Security and Cryptography. Springer, Berlin, Heidelberg
13. Barak B. (2017) The Complexity of Public-Key Cryptography. In: Lindell Y. (eds) Tutorials on the Foundations of Cryptography. Information Security and Cryptography. Springer, Cham
14. Ahmad, Jasmin &amp; Din, Roshidi &amp; Ahmad, Mazida. (2018). Analysis Review on Public Key Cryptography Algorithms. Indonesian Journal of Electrical Engineering and Computer Science.
15. Agnew, G.B., Mullin, R.C., Onyszchuk, I.M. et al. An implementation for a fast public-key cryptosystem. J. Cryptology 3, 63–79 (1991).
16. K. Shim, A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks, in IEEE Communications Surveys &amp; Tutorials, vol. 18, no. 1, pp. 577-601, First quarter 2016.
17. M. Malik, M. Dutta and J. Granjal, A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things, in IEEE Access, vol. 7, pp. 27443-27464, 2019.
18. Ahmad F., Cheng LM. (2019) Paper Document Authentication Using Print-Scan Resistant Image Hashing and Public-Key Cryptography. In: Wang G., Feng J., Bhuiyan M., Lu R. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2019. Lecture Notes in Computer Science, vol 11611. Springer, Cham
19. Rubinstein-Salzedo S. (2018) The RSA Cryptosystem. In: Cryptography. Springer Undergraduate Mathematics Series. Springer, Cham.
20. Tuzzio N., Tehranipoor M. (2012) RSA: Implementation and Security. In: Tehranipoor M., Wang C. (eds) Introduction to Hardware Security and Trust. Springer, New York, NY.
21. Childs L.N. (2019) RSA Cryptography and Prime Numbers. In: Cryptology and Error Correction. Springer Undergraduate Texts in Mathematics and Technology. Springer, Cham.
22. Das S.B., Mishra S.K., Sahu A.K. (2020) A New Modified Version of Standard RSA. Cryptography Algorithm. In: Elçi A., Sa P., Modi C., Olague G., Sahoo M., Bakshi S. (eds) Smart Computing Paradigms: New Progresses and Challenges. Advances in Intelligent Systems and Computing, vol 767. Springer, Singapore.
23. Katzenbeisser S. (2001) Properties of the RSA Cryptosystem. In: Recent Advances in RSA Cryptography. Advances in Information Security, vol 3. Springer, Boston, MA
24. Bhattacharjya A., Zhong X., Wang J., Li X. (2020) A Secure Hybrid RSA (SHRSA)-based Lightweight and Efficient Personal Messaging Communication Protocol. In: Farsi M., Daneshkhah A., Hosseinian-Far A., Jahankhani H. (eds) Digital Twin Technologies and Smart Cities. Internet of Things (Technology, Communications and Computing). Springer, Cham.
25. Liu B. (2020) Research and Implementation of RSA IP Core Based on FPGA. In: Huang C.Chan YW., Yen N. (eds) Data Processing Techniques and Applications for Cyber-Physical Systems (DPTA 2019). Advances in Intelligent Systems and Computing, vol 1088. Springer, Singapore
26. M. Fueyo and J. Herranz, On the Efficiency of Revocation in RSA-Based Anonymous Systems, in IEEE Transactions on Information Forensics and Security, vol. 11, no. 8, pp.1771-1779, Aug. 2016.
27. Zou L., Ni M., Huang Y., Shi W., Li X. (2020) Hybrid Encryption Algorithm Based on AES and RSA in File Encryption. In: Hung J., Yen N., Chang JW. (eds) Frontier Computing. FC 2019. Lecture Notes in Electrical Engineering, vol 551. Springer, Singapore.
28. Shaheen B., Siddiqui F. (2020) Comparison Between RSA Algorithm and Modified RSA Algorithm Used in Cloud Computing. In: Smys S., Bestak R., Rocha Á. (eds) Inventive Computation Technologies. ICICIT 2019. Lecture Notes in Networks and Systems, vol 98. Springer, Cham
29. Yan S.Y. (2008) Private Exponent Attacks. In: Cryptanalytic Attacks on RSA. Springer, Boston, MA.
30. Dubey M.K., Ratan R., Verma N., Saxena P.K. (2014) Cryptanalytic Attacks and Countermeasures on RSA. In: Pant M., Deep K., Nagar A., Bansal J. (eds) Proceedings of the Third International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, vol 258. Springer, New Delhi
31. Nitaj, A. A new attack on RSA with two or three decryption exponents. J. Appl. Math. Comput. 42, 309–319 (2013).
32. Takayasu A., Kunihiro N. (2017) A Tool Kit for Partial Key Exposure Attacks on RSA. In: Handschuh H. (eds) Topics in Cryptology – CT-RSA 2017. CT-RSA 2017. Lecture Notes in Computer Science, vol 10159. Springer, Cham
33. Coron, J., Naccache, D., Desmedt, Y. et al. Index Calculation Attacks on RSA Signature and Encryption. Des Codes Crypt 38, 41–53 (2006).
34. Takayasu A., Kunihiro N. (2016) Partial Key Exposure Attacks on RSA with Multiple Exponent Pairs. In: Liu J., Steinfeld R. (eds) Information Security and Privacy. ACISP 2016. Lecture Notes in Computer Science, vol 9723. Springer, Cham.
35. Cimato S., Mella S., Susella R. (2016) Partial Key Exposure Attacks on RSA with Exponent Blinding. In: Obaidat M., Lorenz P. (eds) E-Business and Telecommunications. ICETE 2015. Communications in Computer and Information Science, vol 585. Springer, Cham.
36. Dujella A. A variant of Wiener’s attack on RSA. Computing 85, 77–83 (2009).
37. Jain M., Singh M. (2017) Identity Based Secure RSA Encryption System. In: Modi N., Verma P., Trivedi B. (eds) Proceedings of International Conference on Communication and Networks. Advances in Intelligent Systems and Computing, vol 508. Springer, Singapore.
38. Schindler W., Wiemers, A. Generic power attacks on RSA with CRT and exponent blinding: new results. J Cryptogr Eng 7, 255–272 (2017).
39. Peng L., Hu L., Huang Z., Xu J. (2015) Partial Prime Factor Exposure Attacks on RSA and Its Takagi’s Variant. In: Lopez J., Wu Y. (eds) Information Security Practice and Experience. ISPEC 2015. Lecture Notes in Computer Science, vol 9065. Springer, Cham.
40. Schindler W. (2015) Exclusive Exponent Blinding May Not Suffice to Prevent Timing Attacks on RSA. In: Güneysu T., Handschuh H. (eds) Cryptographic Hardware and Embedded Systems -- CHES 2015. CHES 2015. Lecture Notes in Computer Science, vol 9293. Springer, Berlin, Heidelberg.
41. May A. (2004) Secret Exponent Attacks on RSA-type Schemes with Moduli N=p r q. In: Bao F., Deng R., Zhou J. (eds) Public Key Cryptography – PKC 2004. PKC 2004. Lecture Notes in Computer Science, vol 2947. Springer, Berlin, Heidelberg.
42. Kumar V., Roy A., Sengupta S., Sen Gupta S. (2017) Parallelized Common Factor Attack on RSA. In: Shyamasundar R., Singh V., Vaidya J. (eds) Information Systems Security. ICISS 2017. Lecture Notes in Computer Science, vol 10717. Springer, Cham.
43. Zheng M., Hu H. (2019) Implicit Related-Key Factorization Problem on the RSA Cryptosystem. In: Mu Y., Deng R., Huang X. (eds) Cryptology and Network Security. CANS
44. Yacobi O., Yacobi Y. (2006) A New Related Message Attack on RSA. In: Goldreich O., Rosenberg A.L., Selman A.L. (eds) Theoretical Computer Science. Lecture Notes in Computer Science, vol 3895. Springer, Berlin, Heidelberg.
45. J. Anitha Ruth, H. Sirmathi and A. Meenakshi, &quot;Secure data storage and intrusion detection in the cloud using MANN and dual encryption through various attacks,&quot; in IET Information Security, vol. 13, no. 4, pp. 321-329, 7 2019.
46. Thakur S.S., Srivastava R. (2020) Dual RSA Based Secure Biometric System for Finger Vein

In: Smys S., Bestak R., Rocha Á. (eds) Inventive Computation Technologies. ICICIT 2019. Lecture Notes in Networks and Systems, vol 98. Springer, Cham.

1. Manu and A. Goel, &quot;Encryption algorithm using dual modulus,&quot; 2017 3rd International Conference on Computational Intelligence &amp; Communication Technology (CICT), Ghaziabad, 2017, pp. 1-4.doi: 10.1109/CIACT.2017.7977331
2. Muhammad N., Zain J.M., Saman M.Y.M., Ramle M.F. (2015) Enhancement of RSA Key Generation Using Identity. In: Sulaiman H., Othman M., Othman M., Rahim Y., Pee N. (eds) Advanced Computer and Communication Engineering Technology. Lecture Notes in Electrical Engineering, vol 315. Springer, Cham.
3. Dutta S.C., Sudha Singh, Singh D.K. (2016) Enhancement of Mobile Ad Hoc Network Security Using Improved RSA Algorithm. In: Pant M., Deep K., Bansal J., Nagar A., Das K. (eds) Proceedings of Fifth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, vol 436. Springer, Singapore.
4. Roy D., Das P. (2018) A Modified RSA Cryptography Algorithm for Security Enhancement in Vehicular Ad Hoc Networks. In: Mandal J., Saha G., Kandar D., Maji A. (eds) Proceedings of the International Conference on Computing and Communication Systems. Lecture Notes in Networks and Systems, vol 24. Springer, Singapore.