# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**CRYPTOGRAPHY AND NETWORK SECURITY**
**AAT REPORT**
**on**

# RSA Algorithm

***Submitted by***


**Varad Vithal KJ(1BM18CS122)**
**Soundarya Lakshmi Anand(1BM19CS407)**

***Under the Guidance of***
**Prof. Dr. Nandhini Vineeth**
**Assistant Professor, BMSCE**


***in partial fulfillment for the award of the degree of***
**BACHELOR OF ENGINEERING**
***in***
**COMPUTER SCIENCE AND ENGINEERING**


**B. M. S. COLLEGE OF ENGINEERING**
**(Autonomous Institution under VTU)**
**BENGALURU-560019**


**Mar-2021 to Jun-2021**

**B. M. S. College of Engineering,**
**Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
**Department of Computer Science and Engineering**

## CERTIFICATE

This is to certify that the AAT work entitled "**RSA Algorithm**" is carried out by **Varad Vithal KJ(1BM18CS122), and Soundarya Lakshmi Anand (1BM19CS407)** who are bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2021. The report has been approved as it satisfies the academic requirements in respect of **Cryptography and Network Security (20CS6PCCNS)** work prescribed for the said degree.

Signature of the Guide
Prof. Dr. Nandhini Vineeth
Assistant Professor
BMSCE, Bengaluru

Signature of the HOD
Dr. Umadevi V
Associate Prof. & Head, Dept. of CSE
BMSCE, Bengaluru

**External Viva**

| Name of the Examiner(s) | Signature with date |
| --- | --- |
| 1.______________________________ | _________________________ |
| 2. ______________________________ | __________________________ |

## B. M. S. COLLEGE OF ENGINEERING

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### *DECLARATION*

We, Varad Vithal KJ (1BM18CS122) and Soundarya Lakshmi Anand (1BM19CS407), students of 6th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, hereby declare that, this AAT entitled "RSA Algorithm" has been carried out by us under the guidance of Prof. Dr. Nandhini Vineeth, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester Mar-2021-Jun-2021

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

Varad Vithal KJ(1BM18CS122)

Soundarya Lakshmi Anand (1BM19CS407)

# Chapter 1

# Introduction

Describe the problem statement (algorithm chosen, attacks, etc.,).

The algorithm chosen for this demonstration is the RSA Algorithm.RSA (Rivest–Shamir–Adleman) is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography, because one of the keys can be given to anyone. The other key must be kept private. The algorithm is based on the fact that finding the factors of a large composite number is difficult: when the factors are prime numbers, the problem is called prime factorization. It is also a key pair (public and private key) generator.

The attack vectors possible on the RSA Algorithm are -

1.Plaintext Attack

2.Chosen cipher Attack

3.Factorization Attack

**Motivation**

Any positive integer greater than 1 can be uniquely factorized into its prime factorization form, but the fact is that it is not easy to do so. The intractability of this factoring problem surprisingly has an ingenious application in cryptography, in fact, the security of the first, most famous and widely used public-key cryptography RSA relies exactly on the intractability of the intractability of the integer factorization problem.

**Various aspects of the algorithm chosen**

A one-way function that is easy to compute; finding a function that reverses it, or computing this function is very difficult.

RSA uses a concept called discrete logarithm. This works much like the normal logarithm: The difference is that only whole numbers are used, and in general, a modulus operation is involved. As an example Ax=b, modulo n. The discrete logarithm is about finding the smallest x that satisfies the equation, when a b and n are provided

# Chapter 2

## Methodology

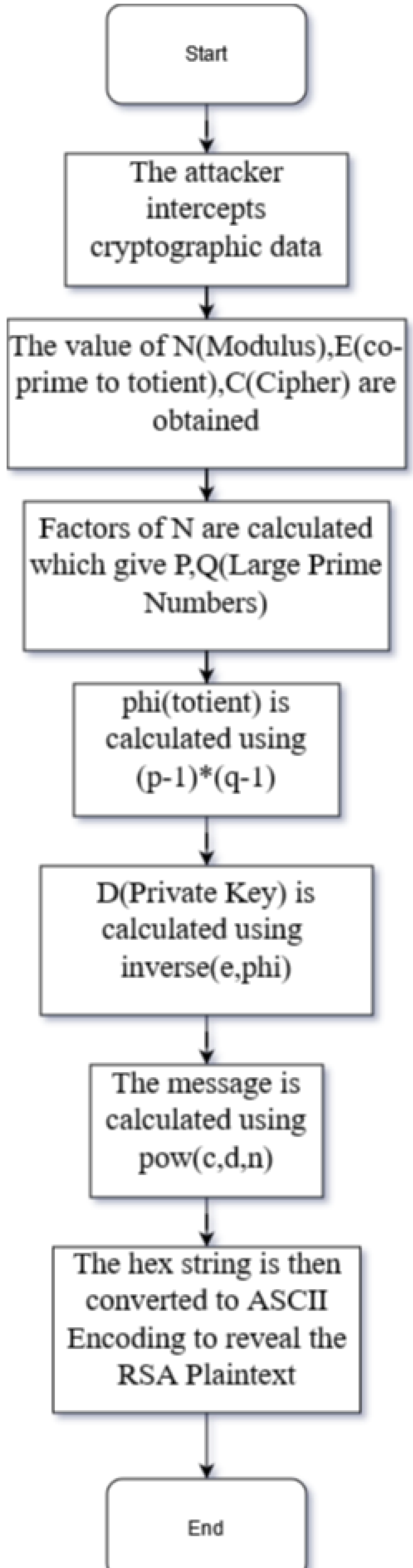
Briefly explain the various steps used in the complete AAT implementation procedure.

Also, draw a flow diagram to show the work flow.

In the following demonstration,Factorization attack is simulated on an intercepted RSA transmission between two parties.In Factorization Attack, the attacker impersonates the key owners, and with the help of the stolen cryptographic data, they decrypt sensitive data, bypassing the security of the system. This attack occurs on An RSA cryptographic library which is used to generate RSA Key. By doing this, Attackers can have the private keys of n number of security tokens, smartcards, Motherboard Chipsets by having a target's public key.These Encryption key used in some of high-security Standard Platforms such as national identity cards, software- and application-signing, and trusted platform modules protecting government and corporate computers.

The steps involved in this demonstration are as follows:

1. The attacker intercepts cryptographic data
2. The value of N(Modulus),E(co-prime to totient),C(Cipher) are obtained
3. Factors of N are calculated which give P,Q(Large Prime Numbers)
4. phi(totient) is calculated using (p-1)*(q-1)
5. D(Private Key) is calculated using inverse(e,phi) {inverse():part of the Crypto library}
6. The message is calculated using pow(c,d,n) which returns a hex string
7. The hex string is then converted to ASCII Encoding to reveal the RSA Plaintext


Start
The attacker intercepts cryptographic data
The value of N(Modulus),E(co-prime to totient),C(Cipher) are obtained
Factors of N are calculated which give P,Q(Large Prime Numbers)
phi(totient) is calculated using (p-1)*(q-1)
D(Private Key) is calculated using inverse(e,phi)
The message is calculated using pow(c,d,n)
The hex string is then converted to ASCII Encoding to reveal the RSA Plaintext
End

# Chapter 3

# Results and Discussion

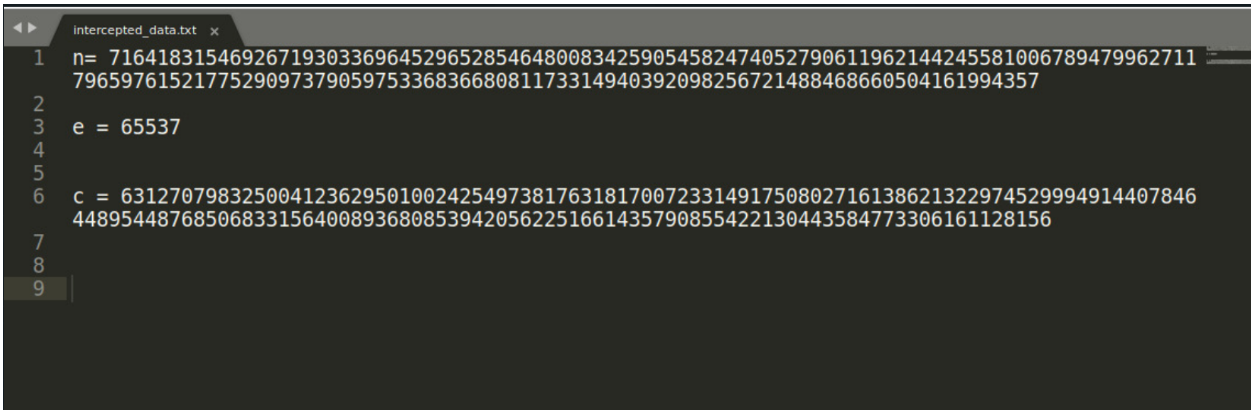



Fig1.0:intercepted RSA Data

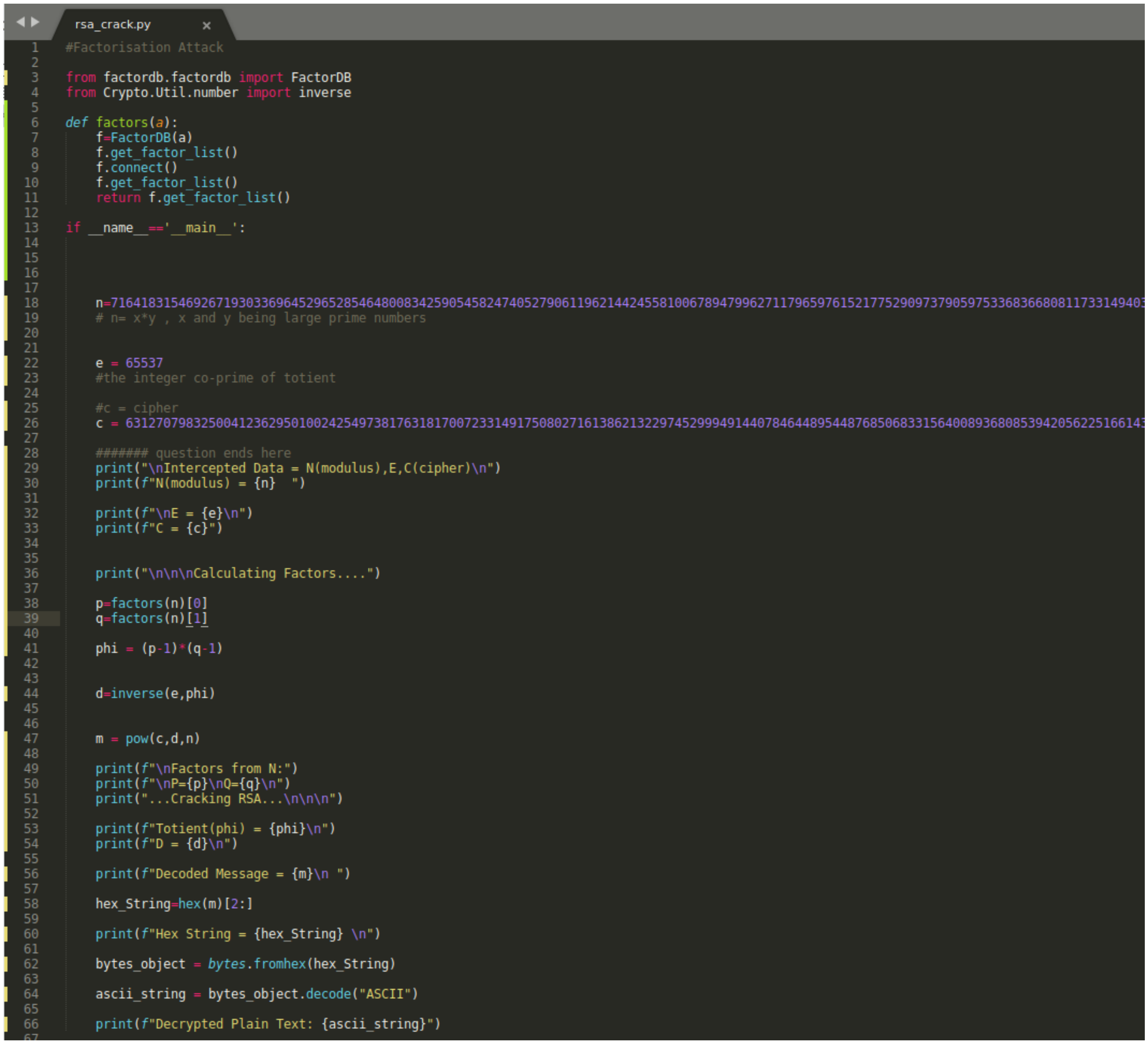



Fig1.2: Cracking RSA using Python and FactorDB

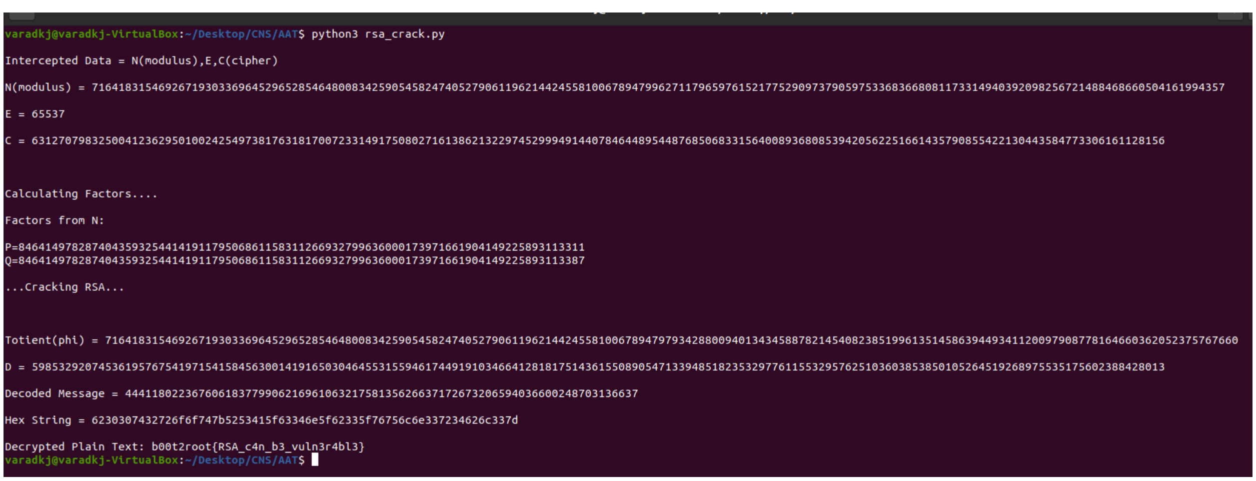


Fig1.3:Decoded RSA Plaintext

# Chapter 4

## Conclusion and Future Work

We studied the factorization and the impact on the security of the RSA cryptosystem.We proposed a novel extension to our previously established methods of semiprime factorization using a sum of squares approach for cryptanalytic attacks on RSA modulus N = p1 p2 with N being a large semiprime, constituting two primes p1 and p2. We gradually developed our proposed technique by providing illustrations of semi-prime factorization for small and large numbers. We arrived at the conjecture that a composite consisting of p unique primes ≡ 1 mod 4, has 2 p−1 sums of squares. We provided the detailed steps involved in the implementation of our enhanced semi-prime factorization algorithm. We then applied our proposed factorization algorithm to attack 768-bit RSA successfully.

## References:


- Research and implementation of RSA algorithm for encryption and decryption - https://ieeexplore.ieee.org/document/6021216

- Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 1978, 21, 120–126. - https://dl.acm.org/doi/10.1145/359340.359342
- Yan, S.Y. Factoring Based Cryptography. In Cyber Cryptography: Applicable Cryptography for Cyberspace Security; Springer: Berlin/Heidelberg, Germany, 2018; pp. 217–286 - https://link.springer.com/chapter/10.1007%2F978-3-319-72536-9_5
- HSCTF - RSA Cryptography (Reverse Search Algorithm) https://youtu.be/Ovi33rfaLLk