A Minor Project Report

On

**Kernel Process grabbing and Decryption**

Submitted in partial fulfilment of requirements for the award of the

Degree of

**Bachelor of Technology**

In

**Information Technology**

Under the guidance of

**(Cryptography Teacher)**

**Submitted By**

**Himanshu Dixit**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Bhagwan Mahaveer College of Engineering and Management Barotha Sonipat**

**November,2020**

**Protruding Certificate**

**Dated:**

**To Whomsoever It May Concern**

This is to Certify that Mr. Himanshu Dixit beholding the project on since last one year under the intuition of **Bhagwan Mahaveer Education Foundation**.

Their major development is to identify the threat of digital vulnerability across all upcoming platforms and neoteric software modules through patching and encryption techniques.

General processing involves identification of loop holes, abduction of resources, present threat on different scale.

In this all tenure of fatigue less work their team work is monotonous positively with equal exchange of ideas and problems.

We have found this is a splendid opportunity and thoughtful idea.

Their miserable project leadings are under the far sight vision of BMEF under covered by …………….. in her best.

**Name of the Supervisor –**

**Sign -**

**Date-**

**Abstract**

Late 19s only have some cellular GSM CDMA mobile phones with some limited features but registration process of cellular network is much more similar to nowadays i.e. 738 million.

Thus, after smart phone development the rate of cellular device increase as the 2% rate of total consumption and now, we have a massive pool of 5.2 billion smart phone users and 80 % are erranded to net services directly or indirectly.

Some popular services are payment gateways, authentication gateways, Add Ons, Bot services, web scripts, apps,mails and finally, not a late one security parameters.

Below presenting the data we got through different reliable sources, references staged at last.

By the help of these data set we helpfully able to statistical analysis of vulnerability to User ratio.

As we are developing nation with an economy of 3.5 trillion per year in which digital services plays a major role.

We can’t prohibit digitalization but the risk rate in online media and software realm is not a new deal.

As we can see the growth of vulnerability is succeeding the rate of involvement.

To provide a certain idea about new potential hazard with some flexible solutions we submitted an intended roll out on this new idea of Cryptography.

**Acknowledgement**

I would like to express my special thanks of gratitude to my Cryptography Teacher (), Advance Networking Teacher () and whole pool of BMCEM education group for their able guidance and support for completing my project.

I would also like to extend my gratitude to the HOD Ma’am () for providing me with all the facility that was required and also all the staff is co-operative in future references also.

**Date: Himanshu Dixit**

**20/10/2020 B-Tech (I.T) 7th Semester**

**CONTENTS**

**LIST Page No.**

**1. Certificate 2**

**2. Acknowledgement 3**

**3. Abstract 4**

**4. Table of Contents**

**5. List of figures 7**

**6. Chapters**

**i) Introduction 8**

**ii) Motivation 9**

**iii) Problem Statement 10**

**iv) Declared Problem 12**

**v) Aim**

* **Objective 13**
* **Cyber Attacks 14**
* **Types of Attacks 15**

**vi) Ransomware**

* **Phases of work 16**
* **Precaution and Removal 17**

**vii) Cryptography**

* **Types of Cryptography 19**
* **Crypto Algo 20**

**viii) AES encryption**

* **Working 21**
* **AES 128 vs AES 256 22**
* **AES vs RSA 23**
* **AES vs DES 24**

**ix) Tools**

* **Strace 25**
* **Pycrypto 26**
* **Shell 27**
* **Ltrace 28**

**x) Process Flow 29**

**xi) Reverse Engineering 36**

**7. Future Scope 38**

**8. References 39**

**9. Checklist 40**

**10. Declaration 41**

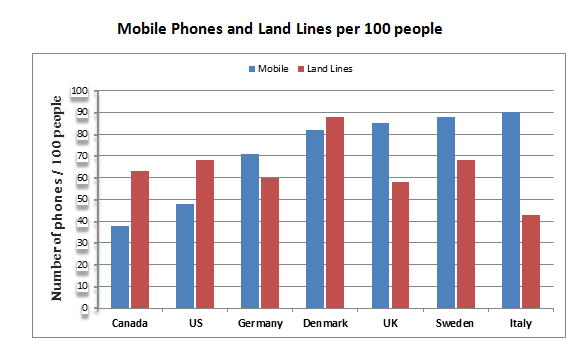
**LIST OF FIGURES**

* **Figure Number 1 🡪 Mobile Phones and landlines per 100 people**
* **Figure Number 2 🡪 World E-Commerce Statistics**
* **Figure Number 3 🡪 Number of spamming bots by country**
* **Figure Number 4 🡪 Ransomware growth rate**
* **Figure Number 5 🡪 Predicted Ransomware attacks 2015-2019**
* **Figure Number 6 🡪 Hacker Shocker Ransomware**
* **Figure Number 7 🡪 Top 10 ransomware families**
* **Figure Number 8 🡪 Types of Cyber Attacks**
* **Figure Number 9 🡪 Crypto Ransomware vs Locker Ransomware**
* **Figure Number 10 🡪 How Ransomware works**
* **Figure Number 11 🡪 Cryptography**
* **Figure Number 12🡪 Types of Cryptography**
* **Figure Number 13 🡪 AES Design**
* **Figure Number 14 🡪 PS AUX Command**
* **Figure Number 15 🡪 Ltrace**
* **Figure Number 16 🡪 Encryption Vs Decryption**
* **Figure Number 17 🡪 XOR Output**
* **Figure Number 18 🡪 Shifting Rows**
* **Figure Number 19 🡪 PID Screen**
* **Figure Number 20 🡪 Tracing Process**
* **Figure Number 21 🡪 Strace Binding**
* **Figure Number 22 🡪 Running Encryption Algorithm**
* **Figure Number 23 🡪 Encrypted File**
* **Figure Number 24 🡪 Analysing Process Memory Map**

**Introduction**

As we are a developing nation and taking a responsible step to make more use of digitalization, the more precautionary we have to be in digital world.

Below is a bar graph representation given for the cell connectivity in U.S.

We chose this data Because Indian Telecom data aren’t precise with practical things in 19’s.

**Fig no. 1**

As you can analyse around 2000 foreign country are outspreaded their feet in connectivity with a traumatic potential.

**Basic Goals of Connectivity** –

1. Reduction of Physical documents example – Tables and notary papers.
2. Fast data sharing anywhere.
3. Secure data and file transmission without loss.
4. Reach to tough areas

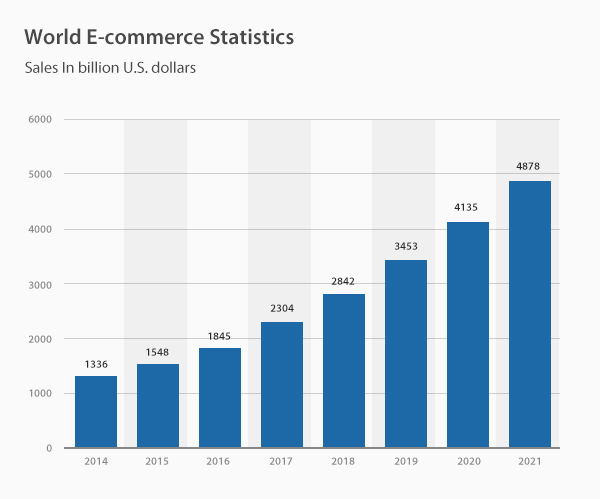
As widespread E-commerce emerges very ghastly in 21th Century. The digital wars begin between crackers, grabbers and malicious data injectors for Money or anything important likewise money example – Crypto Currency (Bitcoin).

The process of grabbing useful data from a victim’s site and in return demanding money is known as **Ransom** and the helping Code is called as **Ransomware**.

**Motivation**

As to achieve fast development in Army, Astronautics or Country, Nations enforcing themselves in the hands of Digital Connectivity but as a security purpose they developed their own protection shields called as **Firewall.**

**But a common man as a soft user only knows basic operations and arranges his/her small works with palmtop, desktop, laptop or mobile phones.**



**Fig no. 2**

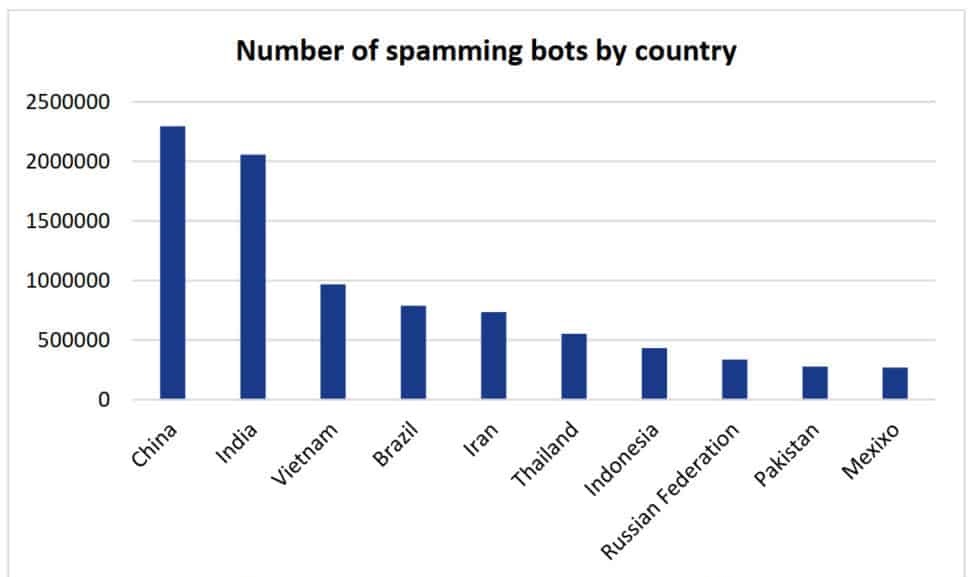
**Above statistics shows how quickly we moved towards Digital world. If you carefully observe the graph represents a Quadratic Equation, which means in upcoming years the pace of E-commerce will be adequately high.**

**Problem Statement**

Whether you are phoning, mailing, surfing, installing or using any internet-based services (sessions calls, API) etc. All services use the network protocols like TCP/IP, UDP, FTP etc. These protocol envelopes the data with sender and receiver information. The data is ad-hoc in these services. The attackers try to figure out loop holes of your system to capture personal snapshots. Loopholes are also called Ports, the gateway through which devices are interlinked.

Although DNS services minimize the attacking potential and other paid firewalls or anti-virus used by people to secure their system through online threats. But still 1 Billion people in India aren’t aware of these crucial tactics.s

Even India is among largest platform of social media apps such as Telegram, WhatsApp etc.



**Fig no. 3**

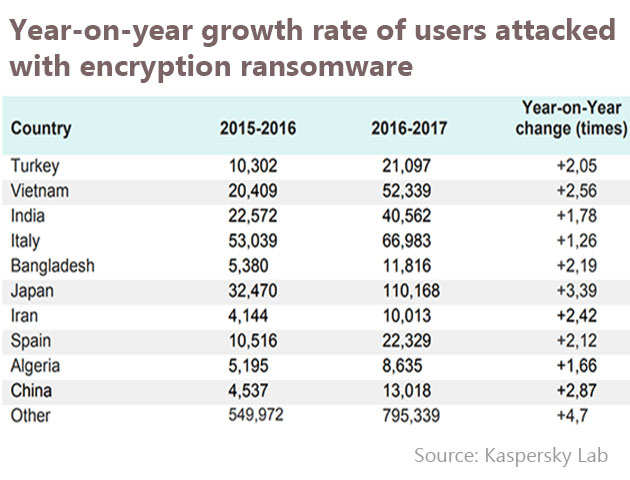
As you can see with a large population with less sophisticated users, India is an international hub to attract attackers and the attackers probably uses Bots to retrieve their credential data. Being online is easy but to maintain yourself in safe zone is conditionally tough for soft users.

A **Bot** is an API based technology uses web developing languages like PHP, JavaScript etc.

These bots basically developed by services provider to broadcast notifications, greetings or any work based on need.

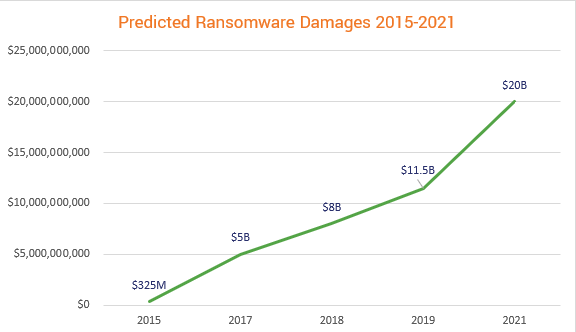
But bots are stealing credentials and can’t alter the form of data. But real threats come in handy when we talk about Ransoming for data.

Below is a depiction of Ransoming of several countries.



**Fig no. 4**

These are only few recorded by Kaspersky Lab but unfortunately every year many more cases of ransom arise with new challenges. Due to limitation of Information and Technology department the current data is not reachable. But some local sources of Business Index magazines show horrible reality of threats.

 The **BussinessLine** predicted the most accurate growth of loss due to Ransoming. A preview of that is given below –

**Fig no. 5**

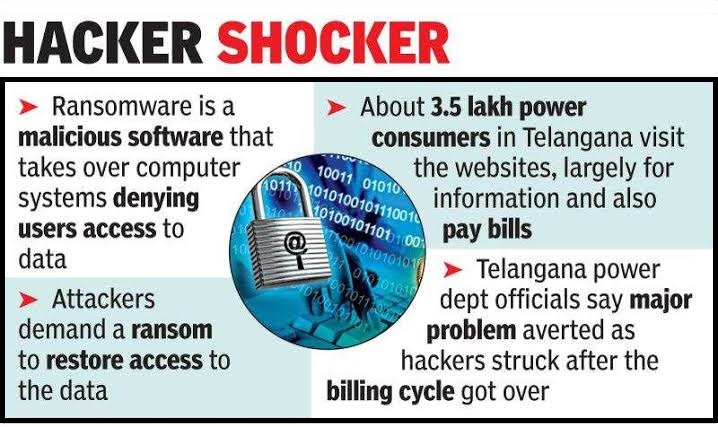
**Declared Problem: -**

The central root of individuals loss in digitalization is majorly due web attacks but almost every case is recoverable except some encrypted attacks.

As we already seen Billions of losses every year is happening due to these ransoming but the prominent way to be successful with suck like deeds is **Crypto Attacks**.

Crypto Attacks are ransom attacks involves encrypted key to decode and encode victim’s data with a possibility of 128-bit, 192 bit or 256 Bit depends on block chain applied to cipher the victim’s info.

A very generous disaster had been seen in the Telangana.



**Fig no. 6**

Electric corporation is backbone of a federal economy and thus attackers are not only individual’s threat, they can buffer the inter as well as intra national relations.

Crypto-technic are evolved by different nations for secure data transmission.

To cipher military messages or codified data signals and also crypto currency enforced using cryptography but every phase comes with pros and cons.

**Goals and Milestones**

To provide more understanding about the major problem and the key role of crypto-technic we tried a small snippet to resolve such problems on machine level. Which we’ll discuss with proceeding chapters.

**Aim 🡪** Nowadays there are many cipher techniques (AES, DES, Black Mount) available with different deploying packages in context of almost every language. Our main aim is to study the internal processing of an encryption decryption on different platforms and subject them for dummy data to grasp encryption key.

**Key features** 🡪

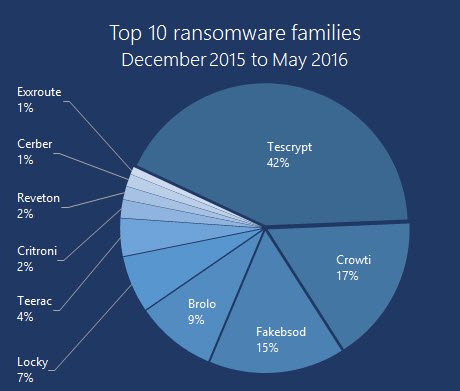
1. Preparing a dummy data file
2. Subjecting to encryption using **Python** module **Pycrypto**.
3. Reading kernel mappings on **Linux** platform.

**Objective: -**

To get a thorough knowledge of attacks, their classification and pre, post precautions.

Process binding to grasp key mapping.

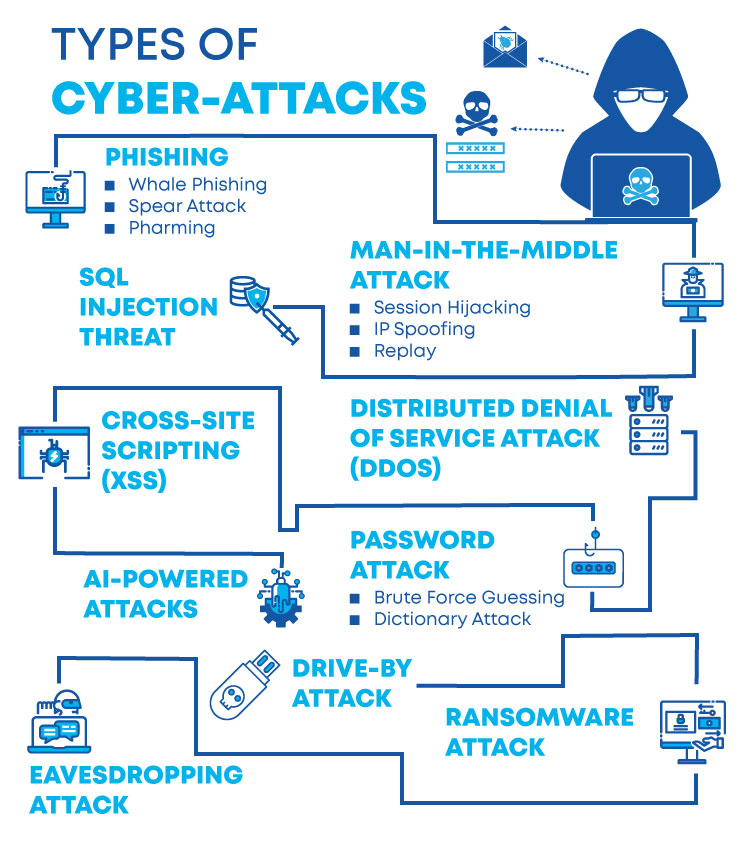
To develop patching modules for windows, Linux, Macintosh and android.

**Top ransom families**

**Fig no. 7**

**Cyber Attack**

A cyberattack is a malicious and deliberate attempt by an individual or organization to breach the information system of another individual or organization. Usually, the attacker seeks some type of benefit from disrupting the victim’s network.



**Fig No. 8**

**Common Types of attacks: -**

**Malware 🡪** Malware is a term used to describe malicious software, including spyware, ransomware, viruses, and worms. Malware breaches a network through a vulnerability, typically when a user clicks a dangerous link or email attachment that then installs risky software. Once inside the system, malware can do the following:

* Blocks access to key components of the network (ransomware)
* Installs malware or additional harmful software
* Covertly obtains information by transmitting data from the hard drive (spyware)
* Disrupts certain components and renders the system inoperable

**Phishing 🡪** Phishing is the practice of sending fraudulent communications that appear to come from a reputable source, usually through email. The goal is to steal sensitive data like credit card and login information or to install malware on the victim’s machine. Phishing is an increasingly common cyberthreat.

**Man-in-the-middle 🡪** Man-in-the-middle (MitM) attacks, also known as eavesdropping attacks, occur when attackers insert themselves into a two-party transaction. Once the attackers interrupt the traffic, they can filter and steal data.

Two common points of entry for MitM attacks:

1. On unsecure public Wi-Fi, attackers can insert themselves between a visitor’s device and the network. Without knowing, the visitor passes all information through the attacker.

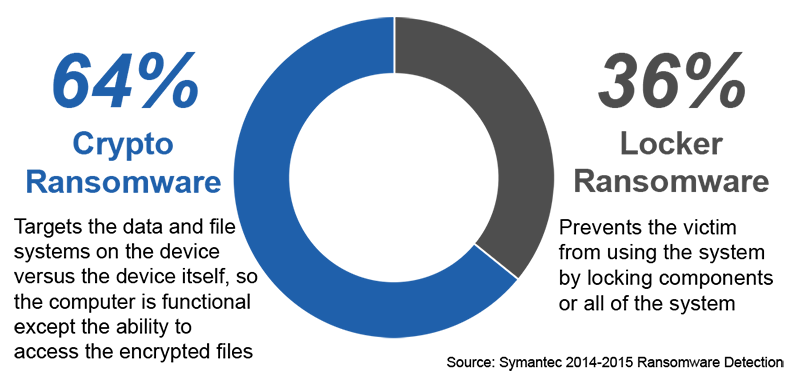
2. Once malware has breached a device; an attacker can install software to process all of the victim’s information.

**SQL Injection 🡪** A Structured Query Language (SQL) injection occurs when an attacker inserts malicious code into a server that uses SQL and forces the server to reveal information it normally would not. An attacker could carry out a SQL injection simply by submitting malicious code into a vulnerable website search box.

**DNS Tunneling 🡪** DNS tunneling utilizes the DNS protocol to communicate non-DNS traffic over port 53. It sends HTTP and other protocol traffic over DNS. There are various, legitimate reasons to utilize DNS tunneling. However, there are also malicious reasons to use DNS Tunneling VPN services. They can be used to disguise outbound traffic as DNS, concealing data that is typically shared through an internet connection. For malicious use, DNS requests are manipulated to exfiltrate data from a compromised system to the attacker’s infrastructure. It can also be used for command and control call-backs from the attacker’s infrastructure to a compromised system.

**What is Ransomware Attack?**

Ransomware is a type of malware attack in which the attacker locks and encrypts the victim’s data and then demands a payment to unlock and decrypt the data.

****This type of attack takes advantage of human, system, network, and software vulnerabilities to infect the victim’s device—which can be a computer, printer, smartphone, wearable, point-of-sale (POS) terminal, or another endpoint.

## Fig No. 9

## How Does Ransomware Work?

## 

## Fig No. 10

1. **Infection 🡪**Ransomware is covertly downloaded and installed on the device.
2. **Execution 🡪**Ransomware scans and maps locations for targeted file types, including locally stored files, and mapped and unmapped network-accessible systems. Some ransomware attacks also delete or encrypt any backup files and folders.
3. **Encryption 🡪**Ransomware performs a key exchange with the Command and Control Server, using the encryption key to scramble all files discovered during the Execution step. It also locks access to the data.
4. **User Notification 🡪**Ransomware adds instruction files detailing the pay-for-decryption process, then uses those files to display a ransom note to the user.
5. **Clean-up 🡪**Ransomware usually terminates and deletes itself, leaving only the payment instruction files.
6. **Payment 🡪**Victim clicks a link in the payment instructions, which takes the victim to a web page with additional information on how to make the required payment. Hidden TOR services are often used to encapsulate and obfuscate these communications to avoid detection by network traffic monitoring.
7. **Decryption 🡪**After the victim pays the ransom, usually via the attacker’s Bitcoin address, the victim may receive the decryption key. However, there is no guarantee the key will be delivered as promised.

**Ransomware Removal: How to Mitigate an Active Ransomware Infection**

If you detected a Ransomware infection in your network, here are the immediate steps you should take to mitigate the threat:

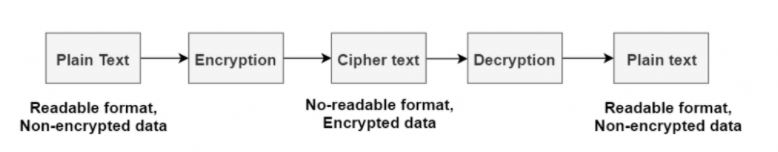
* **Isolate** – identify infected machines, disconnect from networks and lock shared drives to prevent encryption.
* **Investigate** – see what backups are available for encrypted data. Check what strain on ransomware you were hit with, and if there are Decryptor available. Understand if paying the ransom is a viable option.
* **Recover** – if no Decryptor tools are available, restore your data from backup. In most countries, the authorities do not recommend paying the ransom, but this may be a viable option in some extreme cases. Use standard practices to remove ransomware or wipe and reimage affected systems.
* **Reinforce** – run a lesson learned session to understand how internal systems were infected and how to prevent a recurrence. Identify the key vulnerabilities or lacking security practices that allowed the attackers in, and remediate them.
* **Evaluation**– once the crisis has passed, it’s important to evaluate what happened and the lessons learned. How was ransomware successfully executed? Which vulnerabilities made penetration possible? Why did antivirus or email filtering fail? How far did the infection spread? Was it possible to wipe and reinstall infected machines, and were you able to successfully restore from backup? Address the weak points in your security posture to be better prepared for the next attack.

Our main focus on the preliminary step of ransoming i.e**. Ciphering** the data using Encryption.

**Cryptography**

[Cryptography](https://www.geeksforgeeks.org/cryptography-introduction-to-crypto-terminologies/) is technique of securing information and communications through use of codes so that only those persons for whom the information is intended can understand it and process it. Thus, preventing unauthorized access to information. The prefix “crypt” means “hidden” and suffix graphy means “writing”.

In Cryptography the techniques which are used to protect information are obtained from mathematical concepts and a set of rule-based calculations known as algorithms to convert messages in ways that make it hard to decode it. These algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on internet and to protect confidential transactions such as credit card and debit card transactions.

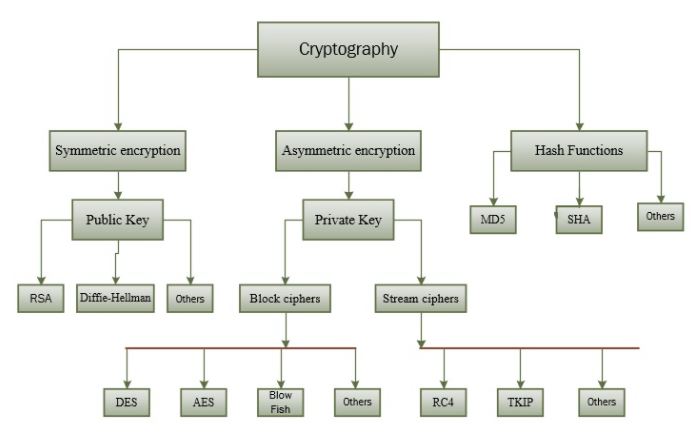


**Fig No. 11**

**Techniques used For Cryptography:**  
In today’s age of computers cryptography is often associated with the process where an ordinary plain text is converted to cipher text which is the text made such that intended receiver of the text can only decode it and hence this process is known as encryption. The process of conversion of cipher text to plain text this is known as decryption.

**Features of Cryptography are as follows:**

1. **Confidentiality:**  
   Information can only be accessed by the person for whom it is intended and no other person except him can access it.
2. **Integrity:**  
   Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.
3. **Non-repudiation:**  
   The creator/sender of information cannot deny his or her intention to send information at later stage.
4. **Authentication:**  
   The identities of sender and receiver are confirmed. As well as destination/origin of information is confirmed.

****

**Types of Cryptography:**  
In general there are three types of cryptography.

**Fig No. 12**

1. **Symmetric Key Cryptography:**  
   It is an encryption system where the sender and receiver of message use a single common key to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner. The most popular symmetric key cryptography system is Data Encryption System (DES).
2. **Hash Functions:**  
   There is no usage of any key in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords.
3. **Asymmetric Key Cryptography:**  
   Under this system a pair of keys is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows the private key.

### **Algorithms**

The **cryptography algorithms** include the following.

In this IoT domain, security matters the most. Though there are many security mechanisms in practice, they do not hold the ability to come up with current day smart applications mainly for the software operating with resource-constraint equipment. In a consequence of this, cryptography algorithms came into practice ensuring enhanced security. So, few of the cryptographic algorithms are as follows:

#### **Triple DES**

Taking over from the conventional DES mechanism, triple DES was currently implemented in the security approaches. These algorithms permit hackers to ultimately gained the knowledge to overcome in an easy approach. This was the extensively implemented approach by many of the enterprises. Triple DES operates with 3 keys having 56 bits per each key. The entire key length is a maximum of bits, whereas experts would contend that 112-bits in key intensity is more probable. This algorithm handles to make a reliable hardware encryption answer for banking facilities and also for other industries.

#### **Blowfish**

To replace the approaches of Triple DES, Blowfish was mainly developed. This encryption algorithm split up messages into clocks having 64 bits and encrypts these clocks separately. The captivating feature that lies in Blowfish is its speed and efficacy. As this is an open algorithm for everyone, many gained the benefits of implementing this. Every scope of the IT domain ranging from software to e-commerce is making use of this algorithm as it shows extensive features for password protection. All these allow this algorithm to be most prominent in the market.

#### **RSA**

One of the public-key encryption algorithms used to encrypt information transmitted through the internet. It was a widely used algorithm in GPG and PGP methodologies. RSA is classified under symmetric type of algorithms as it performs its operation using a couple of keys. One of the keys is used for encryption and the other for decryption purposes.

#### **Twofish**

This algorithm implements keys to provide security and as it comes under the symmetric method, only one key is necessary. The keys of this algorithm are with the maximum length of 256 bits. Of the most available algorithms, Twofish is mainly known by its speed and perfect to be implemented both in the hardware and software applications. Also, it is an openly accessible algorithm and has been in execution by many.

#### **AES (Advanced Encryption Standard)**

The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information. AES is implemented in software and hardware throughout the world to encrypt sensitive data. It is essential for government computer security, cybersecurity and electronic data protection.

The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for an alternative to the Data Encryption Standard (DES), which was starting to become vulnerable to brute-force attacks.

NIST stated that the newer, advanced encryption algorithm would be unclassified and must be "capable of protecting sensitive government information well into the [21st] century." It was intended to be easy to implement in hardware and software, as well as in restricted environments -- such as a smart card -- and offer decent defences against various attack techniques.

AES was created for the U.S. government with additional voluntary, free use in public or private, commercial or non-commercial programs that provide encryption services. However, nongovernmental organizations choosing to use AES are subject to limitations created by

U.S. export control.

**How AES encryption works**

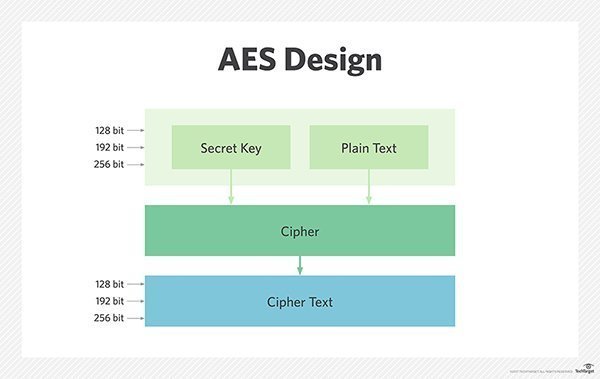
AES includes three block ciphers: AES-128, AES-192 and AES-256.

AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages, while AES-192 uses a 192-bit key length and AES-256 a 256-bit key length to encrypt and decrypt messages. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic

keys of 128, 192 and 256 bits, respectively.

**Symmetric**, also known as *secret key*, ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. A round consists of several processing steps that include substitution, transposition and mixing of the input plaintext to transform it into the final output of ciphertext.



**Fig No. 13**

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array – after which, the cipher transformations are repeated over multiple encryption rounds.

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, and the third mixes columns.

The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete.

**Difference between AES-128 and AES-256**

Overall, security experts consider AES safe against brute-force attacks, in which all possible key combinations are checked until the correct key is found. However, the key size employed for encryption needs to be large enough so that it cannot be cracked by modern computers, even considering advancements in processor speeds based on Moore's law.

A 256-bit encryption key is significantly more difficult for brute-force attacks to guess than a 128-bit key; however, because the latter takes so long to guess, even with a huge amount of computing power, it is unlikely to be an issue for the foreseeable future, as a hacker would

need to use quantum computing to generate the necessary brute force.

Still, 256-bit keys also require more processing power and can take longer to execute. When power is an issue -- particularly on small devices -- or where latency is likely to be a concern, 128-bit keys are likely to be a better option.

When hackers want to access a system, they will aim for the weakest point, which is typically not the encryption, regardless of whether it's a 128-bit key or a 256-bit key. Users should make sure the software under consideration does what they want it to do, that it protects user data in the way it's expected to and that the overall process has no weak points.

Additionally, there should be no Gray areas or uncertainty about data storage and handling. For example, if data resides in the cloud, users should know the location of the cloud. Most importantly, the security software that has been selected should be easy to use to ensure

that users do not need to perform unsecure workarounds to do their jobs.

**AES vs. RSA**

AES is used widely for protecting data at rest. Applications for AES include self-encrypting disk drives, database encryption and storage encryption. On the other hand, the RSA (Rivest-Shamir-Adleman) algorithm is often used in web browsers to connect to websites, in

virtual private network (VPN) connections and in many other applications.

Unlike AES, which employs symmetric encryption, RSA is the base of asymmetric cryptography. Symmetric encryption involves converting plaintext to ciphertext using the same key, or secret key, to encrypt and decrypt it.

On the other hand, the term *asymmetric* comes from the fact that there are two related keys used for encryption: a public and aprivate key. If encryption is performed with the public key, decryption can only happen with the related private key and vice versa. Typically, RSA keys are employed when there are two separate endpoints.

**AES vs. DES**

The U.S. government developed **DES** algorithms more than 40 years ago to ensure government systems all used the same, secure standard to facilitate interconnectivity. DES served as the linchpin of government cryptography for years until 1999, when researchers broke the algorithm's 56-bit key using a distributed computer system. In 2000, the U.S. government chose to use AES to protect classified information. DES is still used in some instances for backward compatibility.

The two standards are both symmetric block ciphers, but AES is more mathematically efficient. The main benefit of AES lies in its key length options. The time required to crack an encryption algorithm is directly related to the length of the key used to secure the communication -- 128-bit, 192-bit or 256-bit keys. Therefore, AES is exponentially stronger

than the 56-bit key of DES. AES encryption is also significantly faster, so it is ideal for applications, firmware and hardware that require low latency or high throughput.

**Protruding Project Tools** 🡪

1. **Strace**
2. **Pycrypto**
3. **Ltrace**
4. **Bash terminal (Linux)**
5. **Python IDE (PyCharm or built-in shell)**
6. **Kali Linux repository and manual.**
7. **Strace 🡪** The Strace command can be used to intercept and record the system calls made, and the signals received by a process. This allows examination of the boundary layer between the user and kernel space which can be very useful for identifying why a process is failing.

## Examples

### **Redirecting trace to a file**

Since Strace often creates a large amount of output, it’s often convenient to redirect it to a file. For example, the following could be used to launch the bash shell, trace any forked child processes, and record all file access to the files. Trace file:

# strace -f -o files.trace -e trace=file bash

### **Counting number of sys calls**

Run the ls command counting the number of times each system call was made and print totals showing the number and time spent in each call (useful for basic profiling or bottleneck isolation):

# strace -c ls

### **Viewing files opened by a process/daemon**

The following example shows the three config files that OpenSSH’s sshd reads as it starts. Note that strace sends its output to STDERR by default, so if you want to pipe it to other commands like grep for further altering you must redirect the output appropriately:

# strace -f -eopen /usr/sbin/sshd 2>&1 | grep ssh

### **Tracing only network related system calls**

Trace just the network related system calls as Netcat attempts to connect to a local telnetd service:

# strace -e trace=network nc localhost 23

**2) Pycrypto** 🡪 is a collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.). The package is structured to make adding new modules easy.

An example usage of the SHA256 module is:

>>> from Crypto.Hash import SHA256

>>> hash = SHA256.new()

>>> hash.update('message')

>>> hash.digest()

'\xabS\n\x13\xe4Y\x14\x98+y\xf9\xb7\xe3\xfb\xa9\x94\xcf\xd1\xf3\xfb"\xf7\x1c\xea\x1a\xfb\xf0+F\x0cm\x1d'

An example usage of an encryption algorithm (AES, in this case) is:

>>> from Crypto.Cipher import AES

>>> obj = AES.new('This is a key123', AES.MODE\_CBC, 'This is an IV456')

>>> message = "The answer is no"

>>> ciphertext = obj.encrypt(message)

>>> ciphertext

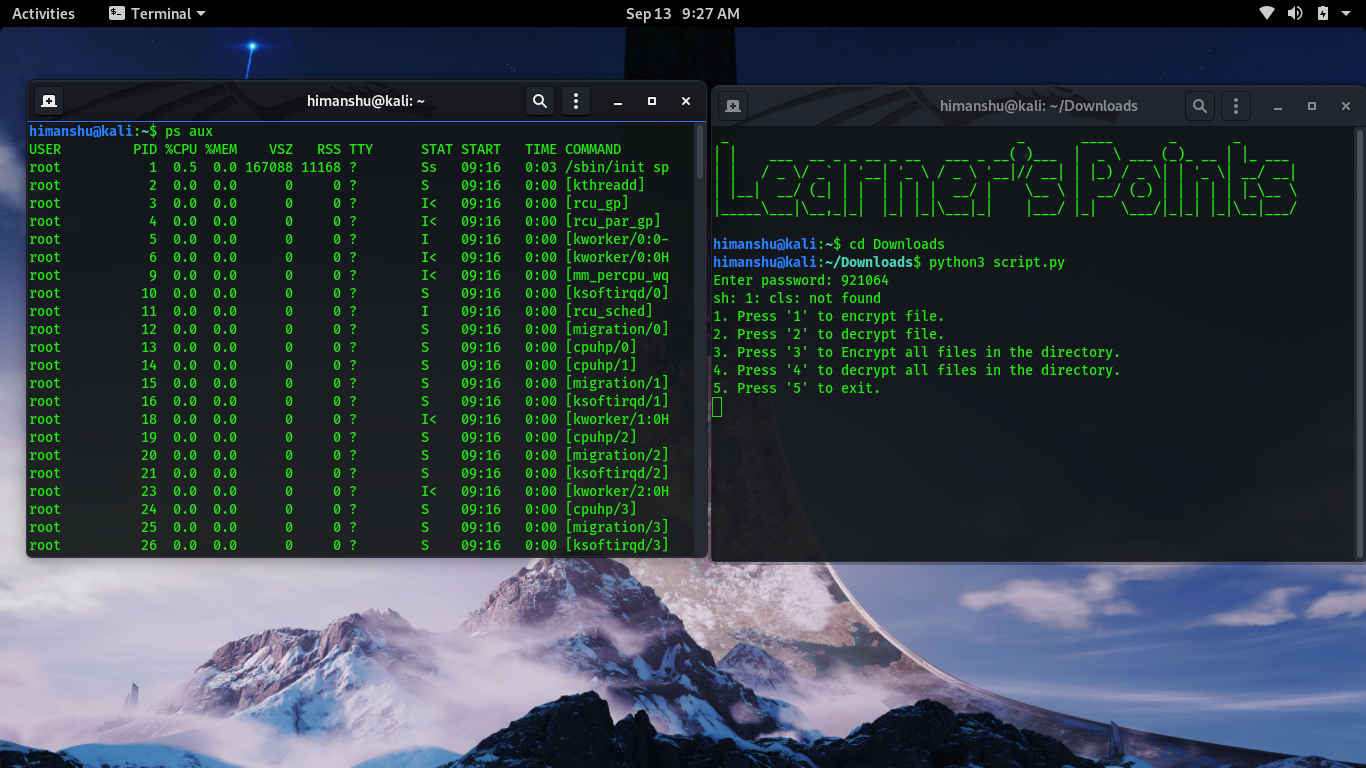
'\xd6\x83\x8dd!VT\x92\xaa`A\x05\xe0\x9b\x8b\xf1'

>>> obj2 = AES.new('This is a key123', AES.MODE\_CBC, 'This is an IV456')

>>> obj2.decrypt(ciphertext)'The answer is no'

**3) Terminal (shell)** 🡪

**ps aux** 🡺 gives a radical information of all running processes with process id and read write address of processes made by virtually or in kernel stack.

****

**Fig No. 14**

As you can see the output of command gives processes with their status of activity.

**D** Uninterruptible sleep (usually IO)

**R** Running or runnable (on run queue)

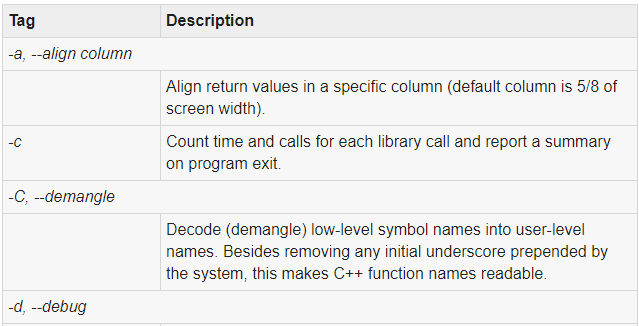
**S** Interruptible sleep (waiting for an event to complete)

**T** Stopped, either by a job control signal

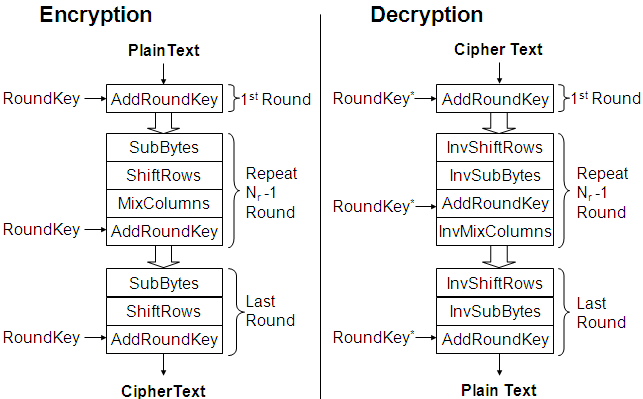
**W** paging (not valid since the 2.6.xx kernel)

**X** dead (should never be seen)

**Z** Defunct ("zombie") process, terminated but not reaped by its parent.

**4) Ltrace 🡪** The ltrace command can be used to intercept and record the dynamic calls made to shared libraries. The amount of output generated by the ltrace command can be overwhelming for some commands (especially if the -S option is used to also show system calls). You can focus the output to just the interaction between the program and some list of libraries.

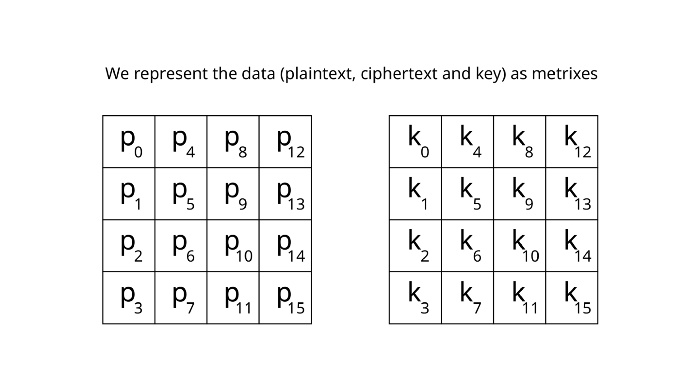
**Fig No. 15**

**PROCESS FLOW CHART OF THE PROJECT**

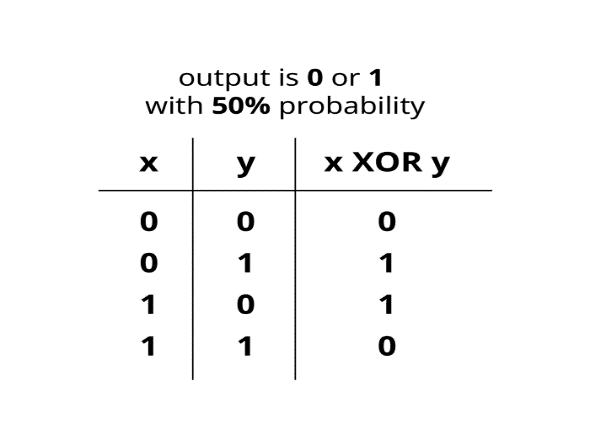
**Fig No. 16**

We can see the algorithm flow likewise:

* **Add round key**
* **Substitute bytes**
* **Shift rows**
* **Mix columns**
* **Add round key**



But before going to first step of algorithm, let’s talk about [block cipher](https://en.wikipedia.org/wiki/Block_cipher). Block cipher is cryptosystem which encrypts data not by bit but by block which is group of bits, applying algorithm per block.

Because AES is also block cipher, we first represent data such as plaintext, ciphertext and key as block. Each block has 1byte(8bit) so in total 16x8=128bit, notice that we have 128-bit key length. And as you can see the order of p\_0, p\_1 …, the data represented as column by column order.

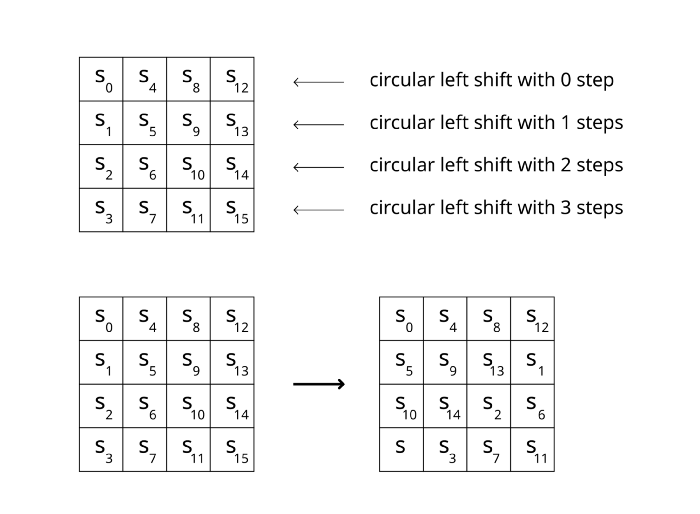
This is the first step of AES algorithm; add round key operation, and this is simply XOR operation. We have 128-bit length plaintext and 128-bit length key so XOR operate bit by bit. And as you can see the diagram the probability of having 0 or 1 is 50% each.

## Fig No. 17

## Round function

We can see the red text “ROUND FUNCTION” in the flow chart of AES, which grouped several functions. And round is simply group of functions, algorithm. And we can say executing 10 rounds as executing 10 times of grouped algorithm. Basically for 128-bit length key, AES takes 10 rounds, 192-bit key for 12 rounds and 256-bit key for 14 rounds.

## Substitute bytes

**Shift rows**

**Fig No. 18**

In the shift rows section, execute circular left shifting for each row. For first row of box shift 0 step to left, second row of box shift 1 step to left, and so on. So after finishing shifting rows, first rows changes from s\_0, s\_4, s\_8, s\_12 to s\_0, s\_4, s\_8, s\_12, second rows changes from s\_1, s\_5, s\_9, s\_13 to s\_5, s\_9, s\_13, s\_1.

**Mixing Columns round key adding and generation of subkey.**

Hence Repetition of these steps for 14 times in 256-bit AES encryption done with one static and 13 dynamic key generations.

**Working** 🡪

## Import modules

from Crypto.Cipher import AES   
from Crypto import Random   
import binascii

Used Random module for simply generating our private key for this example, binascii module for encoding encrypted data to hexcode which helps to see encrypted data.

## Solve padding problems

As we talked before in block cipher, data broke up into 128-bits and make metrixes for that data. But what if the data is less than 128-bit size? If length of data is not 0 (mod 128), then this is the problem. So to solve this problem, we add padding.

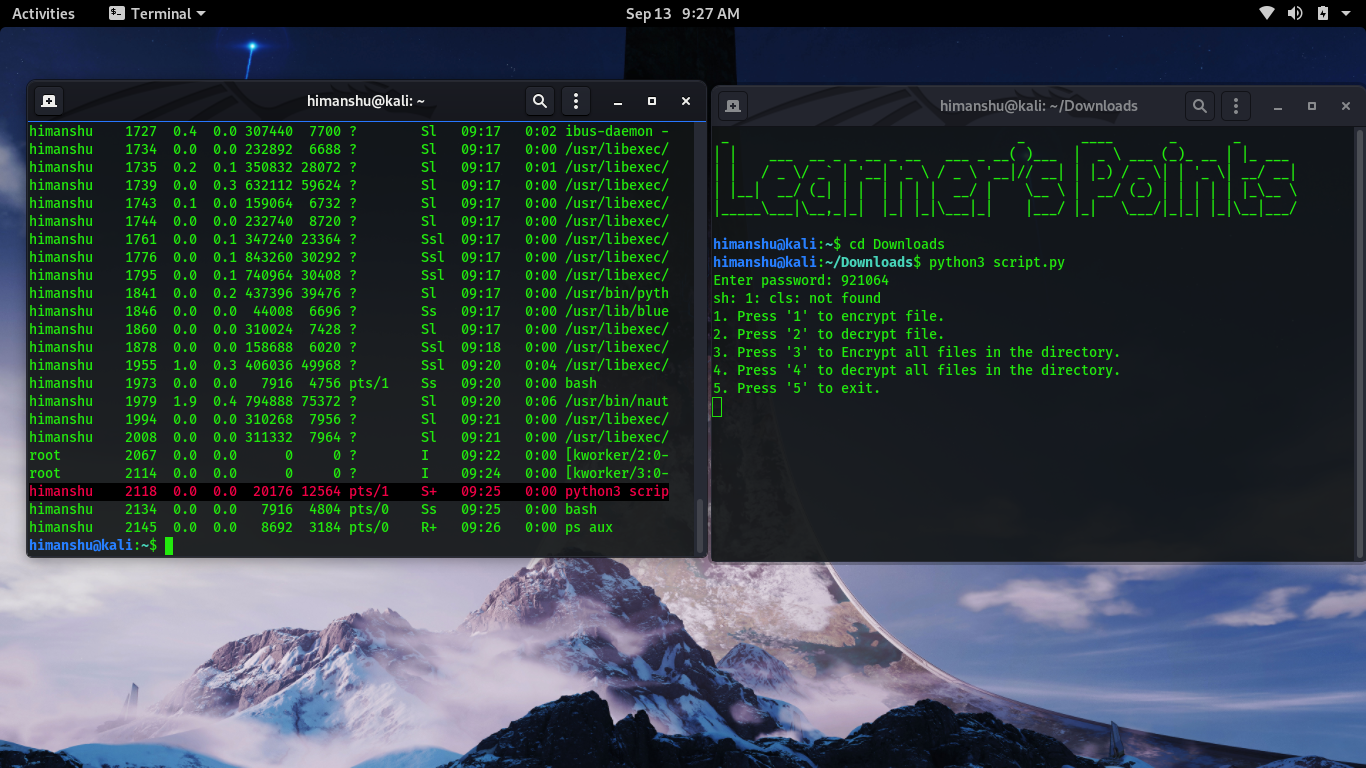
def append\_space\_padding(str, blocksize=128):   
 pad\_len = blocksize - (len(str) % blocksize)   
 padding = 'a'\*pad\_len   
 return str + padding def remove\_space\_padding(str, blocksize=128):   
 pad\_len = 0   
   
 for char in str[::-1]:   
 if char == 'a':   
 pad\_len += 1  
 else:  
 break return str[:-pad\_len]

we defined append\_space\_adding and remove\_space\_adding functions. In append\_space\_padding , add padding value ‘a’ before we encrypt data, in remove\_space\_padding , we remove padding value ‘a’, this is going to be used after decrypt the data.

## Encrypt/Decrypt

def encrypt(plaintext, key):   
 aes = AES.new(key, AES.MODE\_ECB)   
 return aes.encrypt(plaintext) def decrypt(ciphertext, key):   
 aes = AES.new(key, AES.MODE\_ECB)   
 return aes.decrypt(ciphertext).decode('UTF-8')

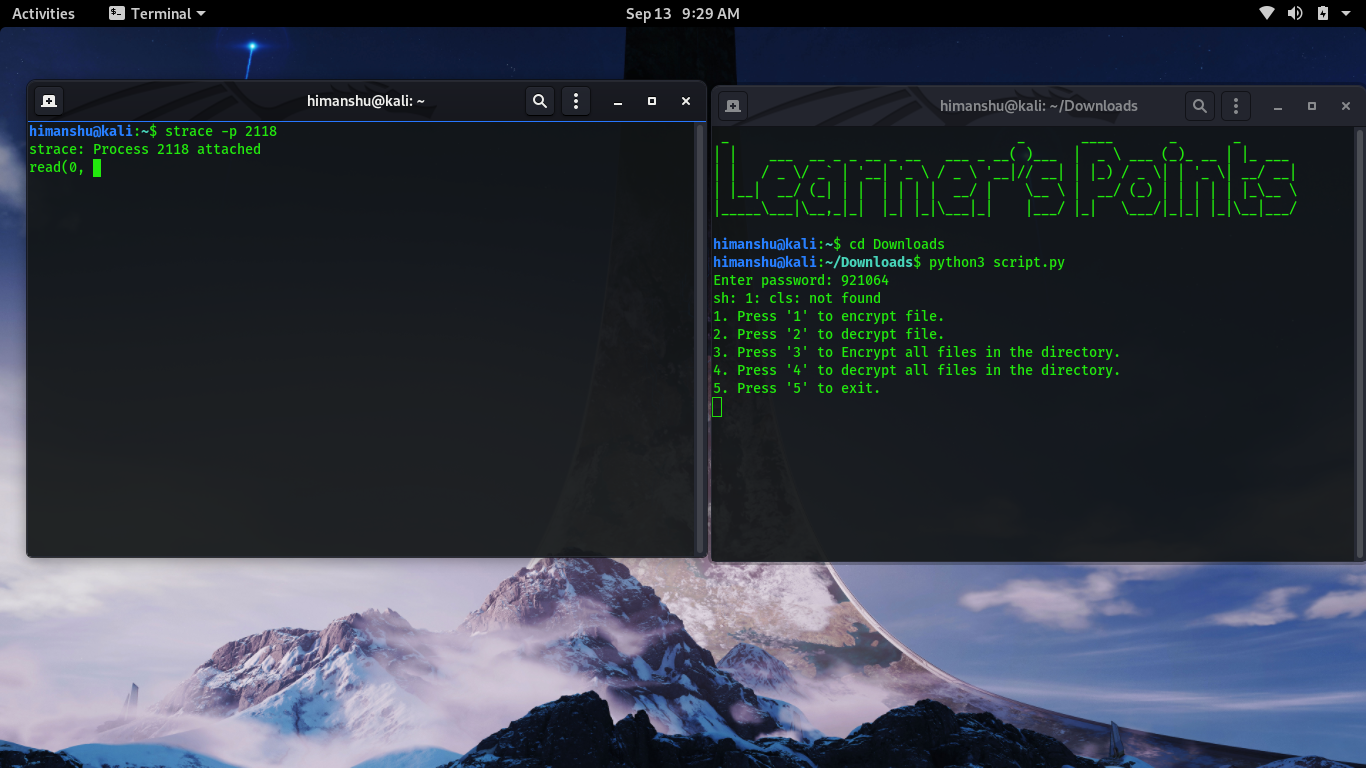
**Reverse Engineering 🡪**

****

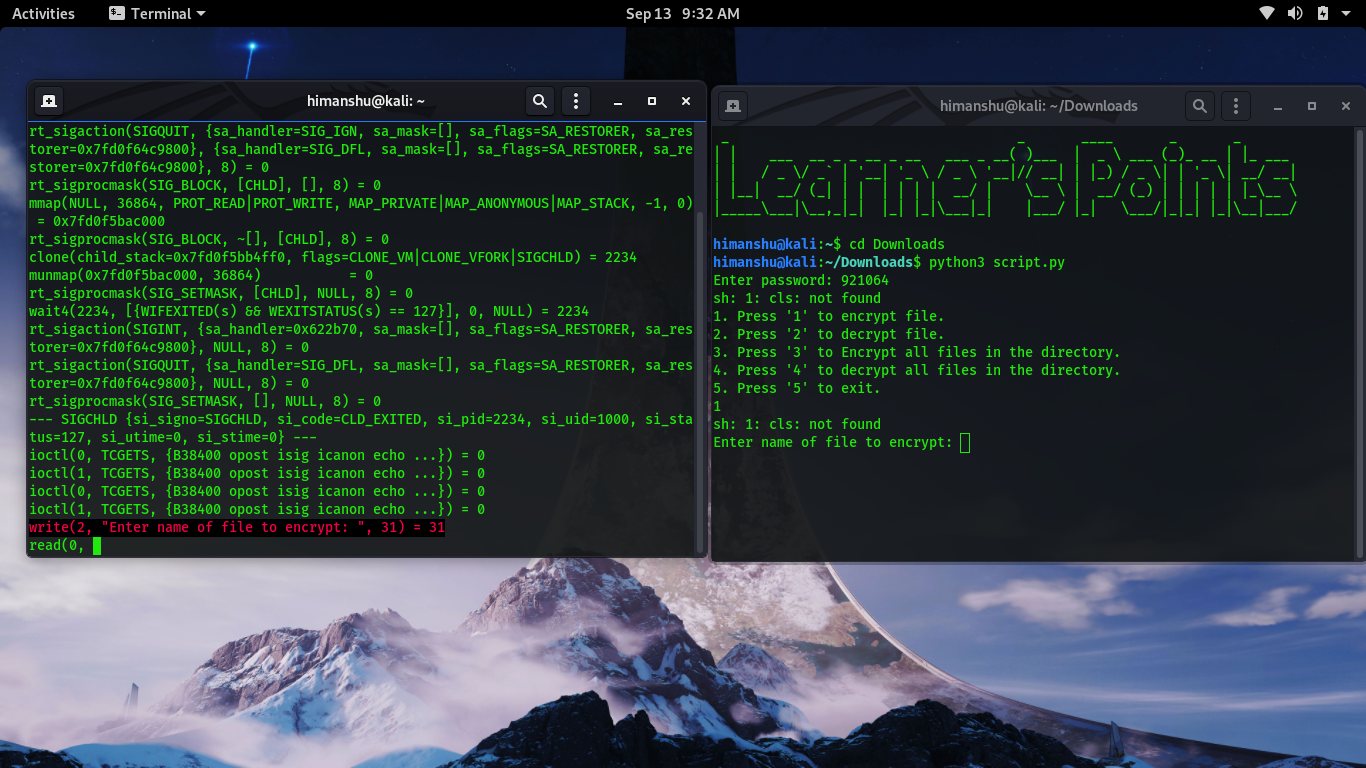
**Fig No. 19**

Running the python AES encryption and grabbing the process id from kernel via **command - ls /proc**

The output returned with process id with its memory size in virtual space.

**Fig No. 20**

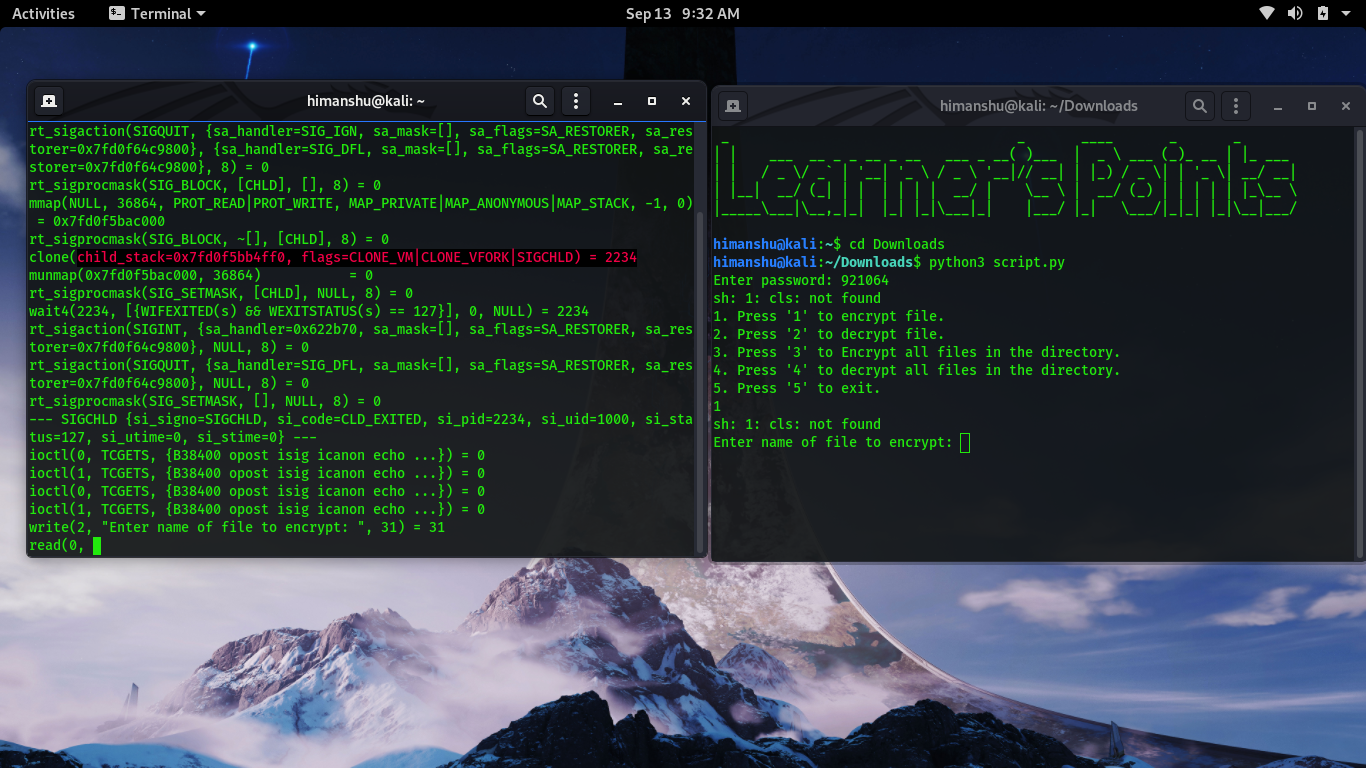
The strace utility patched to running pid of encryption via command.

**Strace -p pid**

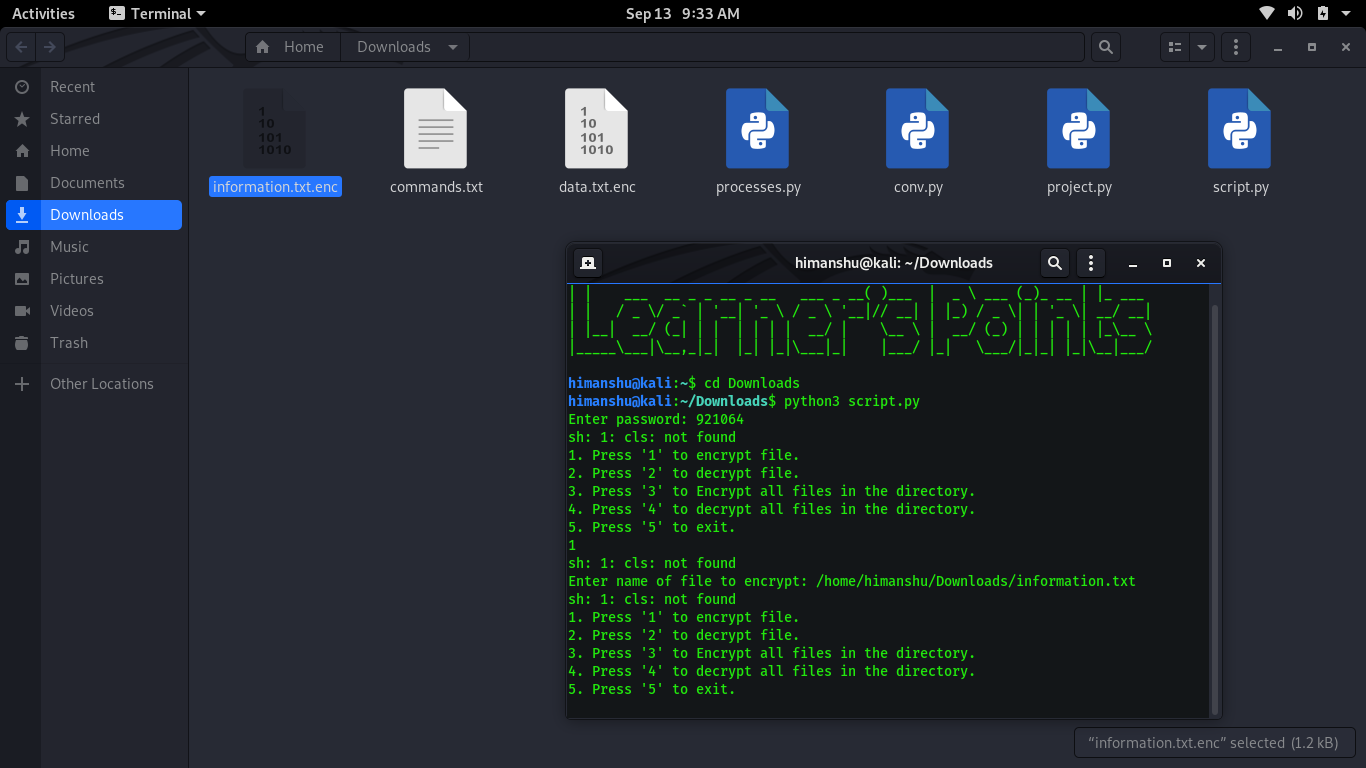
**Fig No. 21**

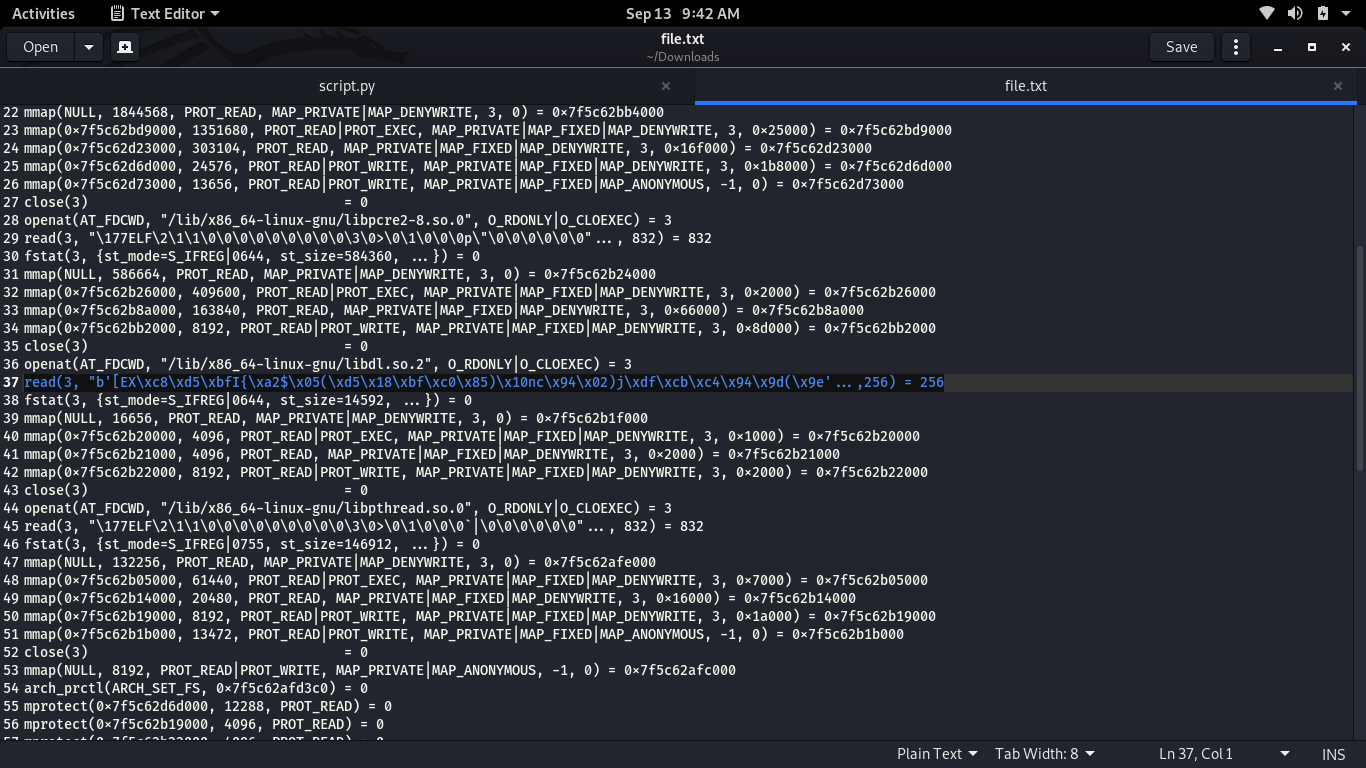
It returns all read write permissions and running on device demands of resources with the return of pointers and memory address and this is the peak time of capturing data from kernel space by commands of physically visiting the kernel memory.

Hence bytes are allocated for storing key first containing a hexadecimal address of file.

****After selecting the option 1 for file encryption the strace utility shows output of screen with address and here one can find optimal screen display codes which are allocated by operating systems for output streams.

**Fig No. 22**

****

**Fig No. 23**

**Fig No. 24**

Using **script.python** using Pycrypto facility we encoded the **information.txt** into **information.txt.enc** encrypted format.

As you can see Strace utility shows the reading of hexadecimal key in 256-bit size memory. One can directly copy and convert it to decimal or further moving with process we will use hexadecimal to decimal converter python script for original key generation.

**FUTURE ENHANCEMENTS (WORK TO BE DONE IN 8TH SEM)**

* In future work of this project, we will try to make it a well-developed for GUI Dos based systems.
* We will also implement it as user friendly patching software
* May turn up these modules in collaboration as a driver update for every BIOS or system.
* We will try to add even more functionalities to make it more user benefiting.

**Bibliography**

**Web-sites**

**1) ILETS Mentor**

**2) vpnmentor**

**3) Helpnetsecurity**

**4)The Economic Times**

**5)Business Line**

**6)Business Help**

**7) Safety Detectives**

**8) Great Learning**

**9) Assure Stor**

**10) The Windows Club**

**11) Security Boulevard**

**12) ResearchGate**

**13)www.rankred.com**

**14)www.techtarget.com**

**15) Wikipedia**

**Books**

1. **The Book Code Simon Singh**
2. **Applied Cryptography by Bruce Schenirer**

**CHECKLIST**

|  |  |  |
| --- | --- | --- |
| **1.** | **Is the report properly hard/spiral bound?** | **Yes** |
| **2.** | **Is the cover page in proper format?** | **Yes** |
| **3.** | **Is the title page (inner cover page) in proper format?** | **Yes** |
| **4.** | **(a) Is the certificate from company in proper format?**  **(b) Has it been signed by the HOD?** | **Yes**  **Yes** |
| **5.** | **(a) Is the acknowledgement from student in the proper format?**  **(b) Has it been signed by the student?** | **Yes**  **Yes** |
| **6.** | **Does Table of Contents include page numbers?**   1. **Are the pages numbered properly?** 2. **Are the figures numbered properly?** 3. **Are the Table numbered properly?** 4. **Are the captions for figures and tables proper?** 5. **Are the appendices numbered properly?** | **Yes**  **Yes**  **Yes**  **Yes**  **Yes**  **Yes** |
| **7.** | **Is the conclusion of the report based on discussion of the work?** | **Yes** |
| **8.** | **Are References or bibliography given in the report?**  **Have the references been cited inside the text of report?**  **Is the citation of references in proper format?** | **Yes**  **Yes**  **Yes** |

**Declaration by Student**

I certify that I have properly verified all the items in the checklist and ensure that the report is in proper format as specified in the course handout.

**Name: Himanshu Dixit**

**Place:**

**Date: 10/12/2020**

**Signature of the Student:**

**Verification by Faculty Project Guide**

I have duly verified all the items in the checklist and ensured that the report is in proper format.

**Name:**

**Place:**

**Date:**

**Signature of the Project Guide:**