

PRACTICAL : 1 CRYPTOGRAPHIC ALGORITHMS

AIM : Study Of Cryptographic Algorithms.

- 1) Ceaser Cipher
- 2) Vernam Cipher
- 3) Columnar Transposition

THEORY:

Cryptography:

The fundamental of cryptography is to enable two people to communicate over an insecure channel in such a way that an intruder cannot understand what is being said. The information called Plain text is encrypted using an encryption key at the sender's side into cipher text. The receiver upon receiving the cipher text decrypts the cipher and gets the plain text back using the decryption key.

A crypt-system is a five tuple (P, C, K, E, D) , where the following conditions are satisfied:

- 1) P is finite set of possible plain text.
- 2) C is a finite set of possible cipher text.
- 3) K , the key space, is a finite set of possible keys.
- 4) For each $k \in K$, there is an encryption rule $e:k \in E$ and a corresponding decryption rule $d:k \in D$. Each $e:k : P \rightarrow C$ and $d:k : C \rightarrow P$ and are functions such that $d:k(e:k(X))=X$ for every plain text $X \in P$.

(1) Ceaser Cipher

Ceaser Cipher has an important place in history. In this scheme each letter is translated to a letter fixed number of places after it in the alphabet. Ceaser used to shift of 3 places, so that the plain text letter P_i was enciphered as cipher text letter C_i by the rule $C_i = E(P_i) = P_i + 3$

Example : The message "TREATY IMPOSSIBLE" would be coded as "wuhdwb lpsrvvleoh"

Algorithm:

- 1) Start
- 2) Enter string to be encrypted
- 3) Encrypt using $C_i = E(P_i) = P_i + 3$
- 4) Decrypt using $P_i = D(C_i) = C_i - 3$
- 5) Display the string and stop.

(2) Vernam Cipher

Basic encryption involves an arbitrarily long non-repeating sequence of numbers that are combined with the characters. The sequence of random numbers has no repeats and are used only once.

Example:

Plain text	V	E	R	N	A	M
Numeric Equivalent	21	04	17	13	00	12

Plain text	V	E	R	N	A	M
Random Number	76	48	16	82	44	03
Sum	97	52	33	95	44	15
Sum % 26	19	00	07	17	18	15
Cipher text	t	a	h	r	s	p

Algorithm:

- 1) Start
- 2) Enter the input string.
- 3) Store it in array.
- 4) Generate a set of random numbers.
- 5) Add the number to plain text numeric equivalent.
- 6) Take the modulus of the sum with 26.
- 7) Print the Cipher, string with ASCII value of sum.
- 8) Decrypt using, (Cipher equivalent integer - (Random number % 26))
- 9) If the resultant is a negative number, add 26 to it to get the equivalent integer for plain text.
- 10) Stop

(3) Columnar Transposition

The columnar transposition is an rearrangement of characters of the plain text characters . Characters are arranged one row after the another. Any blank spaces left in the matrix so formed are filled by some infrequent characters . The mtrix is read column wise to form the cipher. To recover the plain text, cipher text characters are written row wise and read column wise.

Example:

Encryption

Plain text: "THIS IS A MESSAGE"

T	H	I	S	I
S	A	M	E	S
S	A	G	E	X

Cipher text: "tssha aims eeisx"

Decryption

t	s	s
h	a	a
i	m	g

s	e	e
i	s	x

Plaintext: “ THIS IS A MESSAGE”

Algorithm:

- 1) Start.
- 2) Input the plain text string and store it in an array of length N and pad to multiples of 5.
- 3) Read it column wise n form the cipher.
- 4) Stop.

PRACTICAL : 2 DATA ENCRYPTION STANDARD ALGORITHM

AIM : Study Of Data Encryption Standard Algorithm.

THEORY:

DES algorithm is a careful and complex combination of two fundamental building blocks of encryption i.e Substitution and Transposition. The algorithm derives its strength by repeated application of those techniques one on top of the other for a total of 16 cycles.

ALGORITHM:

- 1) Start.
- 2) Enter message and key and convert it into 64 bit block.
- 3) Steps 3 to 13 are repeated on the blocks 16 number of times.
- 4) Divide key into two halves and perform left circular shift according to the round.
- 5) Convert 64 bit key onto 56 bit key by removing every 8th bit.
- 6) Perform permutation to obtain 48 bit key.
- 7) Message is divided into two halves.
- 8) Perform expansion permutation on right half which will convert 32 bit into 48 bit message.
- 9) EXOR the result of expansion permutation & 48 bit key.
- 10) Perform S-box substitution to get 32 bit result.
- 11) Perform P-box permutation.
- 12) EX-OR the result of P-box with left half.
- 13) RHS of message now becomes LHS for next round.
- 14) Original RHS becomes LHS for next round.
- 15) After the 16th round , Final Permutation is performed to get the cipher.
- 16) Stop.

PRACTICAL : 3 RIVEST SHAMIR ADELMAN ALGORITHM

AIM : Study Of RSA Algorithm.

THEORY:

RSA crypto system is a public key system. RSA is similar to other methods in which

solving the cryptosystem amounts to finding terms that add to a particular sum or multiply to a particular product. It relies on number theory. 2 Keys 'd' n 'e' are used for encryption and decryption. They are interchangeable. The plain text block P is encrypted as $P^e \text{ mod } n$. The decryption key is carefully chosen so that $P^{ed} \text{ mod } n = P$. Thus the legitimate receiver who knows 'd'; simply computes $P^{ed} \text{ mod } n = P$ and P without having factor P^e .

ALGORITHM:

- 1) Start.
- 2) Find 2 prime numbers i.e approx 100 digit prime numbers 'p' and 'q'.
- 3) Compute $n = p * q$ where n is also 200 digit number.
- 4) Find 'e' which is relatively prime wrt $(p-1)(q-1)$.
- 5) Find 'd' such that $(e*d) \text{ mod } ((p-1)(q-1)) = 1$.
- 6) Message is divided into blocks of sizes less than n.
- 7) Cipher text is obtained by $C_i = M_i^e \text{ mod } n$.
- 8) Decrypt using $M_i = C_i^d \text{ mod } n$.
- 9) Stop.

PRACTICAL : 4 COVERT CHANNELS

AIM : To implement covert channels using multithreading.

THEORY:

Covert Channel

These are certain programs that communicate information to people who should not receive it. The information travels unnoticed, accompanying other, proper information. The general name for these paths of communication are known as covert channels. They are means of extracting data.

It basically consist of a service program containing Trojan Horse that tries to copy info from a legitimate user to a spy program . The user may not know that a trojan horse is running and may not be in collision to leak information to the spy.

Storage Channels

Certain Covert Channels are called storage channels since they pass info by using presence or absence of objects in storage.

Timing channels

These covert channels pass information using speed at which things happen. They are shared resource channels where time is Shared resource.

ALGORITHM:

- 1) Start.
- 2) In the service program, take the input as binary string.
- 3) For every 1 in the input create a file and/or put a wait state for few millisecs.
- 4) In the mean while Spy program checks to see whether the file exists to indicate a message 1 else 0.
- 5) Stop.

PRACTICAL : 5

CHALLENGE RESPONSE SYSTEMS

AIM : To implement challenge response systems.

THEORY:

A one time password is one that changes every time it is used instead of assigning a static phrase to a user, the system assigns a static mathematical function. The system provides an agreement to the function value and user computes and returns the function value. Such a program is called Challenge Response Systems because the system presents a challenge to the user and judge authenticity of users by user's response.

Examples of One time passwords are:

1) $F(x)=x+1$

With the function, system prompts with value of x and user enters value of (x+1).

2) $F(x)=x(k)$

Receiver uses the argument as a seed for random number generation. The user replies with value of first random number generated.

3) $F(a1,a2,a3,a4,a5)= a3a2a1a4$

User must transform the character string in some pre-defined manner.

4) $F(E(x))=E(D(E(x))+1)$

In this function, the computer sends an encrypted value E(x). The user must decrypt the value, perform some mathematical function and encrypt the result to return it to system.

PRACTICAL : 6

SQL INJECTION

AIM: To implement SQL injection attack.

THEORY:

SQL injection is a technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal, escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. It is in fact an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another.

e. g. suppose username : a100' OR 1=1- -

password : anything

then the query in DB is executed as

select * from usertable

where username='a100' OR 1=1- -'

and password='anything'

becomes true as 1=1 is true and - - marks the start of SQL comment

ALGORITHM:

1. Start
2. Create a GUI for entering username and password

3. Establish database connection using JDBC
4. Perform SQL injection attack
5. End

PRACTICAL : 7 MONITORS

AIM: To study and implement different types of monitors.

THEORY:

Monitors are those units of DBMS, responsible for structural integrity of databases. a monitor can check values being entered to ensure their consistency with the rest of the database or with characteristics of a particular field.

1. Range Comparison : A range comparison monitor tests each time to ensure that the value is within an acceptable range. It is rejected and not entered into the databases. They are also convenient for numeric quantities.
e. g. a salary field might be limited to \$200000. They can be used to ensure internal consistency of database and to test existing values for reasonableness.
2. State constraints: It describes the condition of entire database. At no time should database values violate these constraints.
e. g. consider a database of employee classification. At any time at most one employee is classified as 'president' furthermore, each employee has an employee number different from that of every other employee. If a mechanical / software failure causes portions of the database file to be duplicated, one of these uniqueness constraints might be violated.
3. Transition constraints : State constraints describe the state of a correct database. Transition constraints describe conditions necessary before changes can be applied to a database.
e. g. before a new employee can be added to the database, there must be a position not in the database with status 'vacant'. After the employee is added, exactly one slot must be changed from 'vacant' to not of new employee.

PRACTICAL : 8 NMAP

AIM: To study Network Mapper (NMAP).

THEORY:

NMAP is designed to allow system administrator and various individuals to scan large WWS to determine which hosts are up and what services they are offering. NMAP supports a large number of scanning techniques such as UDP, TCP, connect (), TCP, ICMP, IP protocol. NMAP also offers no of advance features such as remote access detection via TCP/IP finger printing, dynamic delay, parallel scanning detection of down host via parallel host. NMAP should be run as root whenever possible. The result of NMAP is usually list of interesting ports on the machine being scanned. NMAP gives the port "well known" service name (if any), number, state protocol. The state is neither

open,filtered or unfiltered.Open state means the larger machine will accept connections on that port.Filtered port means that a firewall ,filter,other network obstacle is converting the port 'A' preventing NMAP in determining whether the port is open.Unfiltered state means port is known by map to be closed and no firewall filter seems to be interfering with NMAP attempts to determine this.The main feature of NMAP is that it is flexible,powerful and easy to use.

EXAMPLE:

```
$nmap -v target.example.com
```

This will scan all the received ports on the machine target.example.com 'v' indicates verbose mode.

```
$nmap -ss -o target.example.com
```

launches a stealth SXN scan against each machine.

COMMANDS USED

```
$nmap -SJ      :TCP stealth port scan
$nmap -ST      :TCP connect( ) port scan
$nmap -SU      :UDP port scan
$nmap -SP      :ping scan
$nmap -SF,-SX,-SN :stealth FIN or NULL scan
$nmap -O       :use TCP/IP fingerprinting to access remote o.s.
$nmap -P<range> :ports to scan
$nmap -f       :only scan ports listed in nmap services
$nmap -v       :verbose
$nmap -iI<inputfile> :gets target from file
$nmap -S<your ip> -e <device name> :specify source address
```

SAMPLE COMMANDS:

```
nmap -SS 192.168.1.165
nmap -ST 192.168.150.100
nmap -SP 192.168.150.100
nmap -() 192.168.0.0/16
```

PRACTICAL: 9 SNORT

AIM :To study SNORT

SOFTWARE : Snort 2-2-0-1-0 rvsdag 138 b.rpm

O.S. :Linux

THEORY:

There are three main nodes in which snort can be configured.sniffer,packet logger and network instruction detection system.sniffer node simply read the packet of the network. and display them in a continues stream in console.Packet logger modifies the log packet to the disk.Network instruction detection mode is most complex and configure allowing snort to analyze network traffic to match against a user defined rule set and perform general action based on what it sets.

EXAMPLES

1. Sniffer mode: In order to print out the TCP/IP packet header to the screen, following commands are used.
2. Packet logger mode: In order to record the packet to the disk, you need to specify the logging directory and snort will automatically go to packet logger mode.

```
$ 1 snort-dev-&-log
```

The directory log in the current directory when snort will run in this mode. It collects every packet it sees and places it in a directory hierarchy based upon the IP address of one of the hosts. 1 indicates that snort uses the address of the remote computer as directory in which it places sometime it uses local host address.

```
$1 snort-dev-l-log m-192-168-1-0124
```

It indicates that the user wants to print the data link TCP/IP radices application data into the directory/log. All incoming packets will be recorded in the sub-directory of the log/directory. Directory name will be based on the address of the remote host.

3. Network intrusion detection mode:

To enable n/w intrusion detection (NIDS)

```
$snort dev-l-log-h-192-168 1-0124-c
```

Snort conf is the name of the rule file. This will apply the rule-set in the snort conf file to each packet to decide if an action based upon the rule is taken.

```
$1 snort-d-n 192.168.01/24-1 1log-c snort conf .
```

This will configure snort to run in its mode basic. NIDS from logging present that the rule tells it to in plain.

ASCII to a hierarchical directory structure

SAMPLE COMMANDS

```
#snort -v # snort-vd #snort -vde
```

```
#snort -v -d -e #snort -dev -l-log
```

```
#snort -dev -l -lg -h 192.168.1 -0214
```

```
#snort -l.log -b
```

```
#snort -div -v packet.log
```

PRACTICAL : 10 IP TABLES

AIM: Study IP Tables.

THEORY:

IP tables is a packet filter in the Linux Kernel. It uses the Net Filter architecture to capture packets. But apart from packet filtering it can also perform network address translation, masquerading and packet mangling.

IP tables can be configured to be:

1. A simple packet filtering based on the rule set specified by the user. In this mode you never alter packets, packets are simply matched based on specified criteria and appropriate action is taken.
2. NAT in addition to packet filtering, masquerading is special form of source NAT-port.

Forwarding and transparent proxying are special forms of destination NAT.

3. Packet mangling in addition to Packet Filtering .packet mangling where you actually change the packets.

IP tables also has connection tracking facility.This is basically in the realm of NAT but is actually implemented in the IPtables as a different module.

Download and Install the IPtables package.

Before you begin,you need to make sure that the IPtables software.RPM is installed when searching for the RPM's , remember that filename has package 1.2.9-1.0.1386 rpm

How to start IP tables:

you can start,stop and erstart after booting by using

```
[ root @ goodgirl tmp] # service iptables start
```

```
[ root @ goodgirl tmp] # service iptables stop
```

```
[ root @ goodgirl tmp] # service iptables restart
```

Determining the states of IPtables

You can determine whether IPtables is running or not via the service IPtables status command.

Fedora core will give a simple message.

e.g

```
[root @ goodgirl tmp] # service iptables status
```

```
firewall is stopped
```

```
[ root @ goodgirl tmp] #
```

COMMON COMMANDS:

-t <table> :If you do not specify table then the filter table is default.

-j<target> :Jump to specified target chain when the packet matches the current rule .

-A :Append rule to end chain.

COMPUTER NETWORKS

List of Lab Experiments

Name of the subject	Computer Network	
Semester	V	
Number of sessions /week	2 hours/week	
Sr. no.	Major topic	Title of the experiment

1	Cryptography	Study Cryptographic Algorithms a)Ceaser Cipher b)Vernam Cipher c)Columnar Transposition
2	Cryptography	Data Encryption Standard .
3	Cryptography	Rivest Shamir Adelman (RSA) Algorithm
4	Program Security	Covert Channels
5	Operating System Security	Challenge Response Systems
6	Program Security	SQL Injection
7	Database Security	Monitors
8	Network Security	NMAP
9		SNORT
10	Advance Computer Network	IP Tables