Practical Workbook

Network & Information Security (CT-460)

Name	______________________________
Year	______________________________
Batch	______________________________
Roll No	______________________________
Department	______________________________

Department of Software Engineering
NED University of Engineering & Technology

Practical Workbook

Network & Information Security (CT-460)

Prepared By

Engr. Muhammad Faraz Hyder
Assistant Professor
Software Engineering

Approved By

Prof. Dr. Najmi Ghani Haider
Chairman
Software Engineering

Department of Software Engineering
NED University of Engineering & Technology

C O N T E N T S

Lab Session 01

OBJECT: Analysis of Network Protocols

THEORY

- **Ping**

ping [-t] [-a] [-n count] [-l length] [-w]

A utility to determine whether a specific IP address is accessible. It works by sending a packet to the specified address and waiting for a reply. PING is used primarily to troubleshoot Internet connections.

It is often believed that "Ping" is an abbreviation for Packet Internet Groper, but Ping's author has stated that the names comes from the sound that a sonar makes.

- **Ping –t**

-t Ping the specified host until stopped. To see statistics and continue - type

Control-Break; To stop – type Control-C.

Continuously ping the local host, until you press **Ctrl+C**.

This function has no special skills, but can be used in coordination with other parameters

- **Ping –a**

Resolve addresses to hostnames. Resolve the NetBios name of PC.

Example: C:\>ping -a 192.168.1.21

```
Pinging neduet.edu.pk  [192.168.1.21] with 32 bytes of data:
Reply from 192.168.1.21: bytes=32 time<10ms TTL=254
Reply from 192.168.1.21: bytes=32 time<10ms TTL=254
Reply from 192.168.1.21: bytes=32 time<10ms TTL=254
Reply from 192.168.1.21: bytes=32 time<10ms TTL=254
Ping statistics for 192.168.1.21:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The above shows that the NetBios name of the PC with its IP being 192.168.1.21 is **neduet.edu.pk**

- **Ping –n**

-n count Number of echo requests to send. Send the number of echo packets specified by **count**. Four packets are sent by default. You can use this command to customize the sending count, which is helpful for measuring the network speed. For example, perform the following operation to test the average/maximum/minimum time when 50 packets are returned:

```
C:\>ping -n 50 202.103.96.68
Pinging 202.103.96.68 with 32 bytes of data:
Reply from 202.103.96.68: bytes=32 time=50ms TTL=241
Reply from 202.103.96.68: bytes=32 time=50ms TTL=241
Reply from 202.103.96.68: bytes=32 time=50ms TTL=241
Request timed out.
………………
Reply from 202.103.96.68: bytes=32 time=50ms TTL=241
Reply from 202.103.96.68: bytes=32 time=50ms TTL=241
Ping statistics for 202.103.96.68:
Packets: Sent = 50, Received = 48, Lost = 2 (4% loss),
Approximate round trip times in milli-seconds:
Minimum = 40ms, Maximum = 51ms, Average = 46ms
```

The above shows that 48 packets are returned among 50 packets sent to 202.103.96.68, and two packets are lost due to unknown reasons. For the return of 48 packets, the minimum time is 40ms, the maximum time is 51ms, and the average time is 46ms.

- **Ping –l**

-l size Send buffer size.Define the size of echo packet.

The size of ping packets in Windows is 32 bytes by default. You can customize the size and limit it to 65500 bytes. Such limit is set because of a security hole of Windows system. That is, when 65532 bytes or more are sent to the remote end for once, the remote end probably becomes down. Therefore, Microsoft limits the size of ping packets. However, this parameter is still harmful in coordination with other parameters. For example, the following is an attacking command by coordinating **-l** with **–t**.

```
C:\>ping -l 65500 -t 192.168.1.21
Pinging 192.168.1.21 with 65500 bytes of data:
Reply from 192.168.1.21: bytes=65500 time<10ms TTL=254
Reply from 192.168.1.21: bytes=65500 time<10ms TTL=254
………………
```

In this case, your PC will continuously send packets of 65500 bytes to 192.168.1.21. Sending such packets from several PCs at the same time can make the remote end crash.

- **Ping –w**

-w timeout Timeout in milliseconds to wait for each reply. Specify the timeout, in milliseconds.

By default ,ping waits 1,000ms (1 second) for each response to be returned before displaying the "Request Timed Out" message.if the remote system being pinged is across a high-delay link, such as a satellite link,responses may take longer to be returned. You can use the –w (wait) option to specify a longer time-out.

- **IPconfig**

Ipconfig is a DOS utility which can be used from MS-DOS and a MS-DOS shell to display the network settings currently assigned and given by a network. This command can be utilized to verify a network connection as well as to verify your network settings.

ipconfig [/? | /all | /release | /renew| /flushdns | /displaydns]

/all	Display full configuration information.
/release	Release the IP address for the specified adapter.
/renew	Renew the IP address for the specified adapter.
/flushdns	Purges the DNS Resolver cache.
/displaydns	Display the contents of the DNS Resolver Cache.

- **Ipconfig /renew**

When you use ipconfig/renew, all network adapters on the computer that uses DHCP (except those are manually configures) try to contact a DHCP server and renew their existing configuration and obtain the new configuration.

- **Ipconfig /release**

You can use the ipconfig command with the /release option to immediate release the current DHCP configuration for a host.

• Traceroute

The traceroute command traces the network path of Internet *routers* that *packets* take as they are forwarded from your computer to a destination address. The "length" of the network connection is indicated by the number of Internet routers in the traceroute path.

Traceroutes can be useful to diagnose slow network connections. For example, if you can usually reach an Internet site but it is slow today, then a traceroute to that site should show you one or more hops with either long times or marked with "*" indicating the time was *really* long. If so, the blockage could be anywhere from your Internet service provider to a *backbone provider*, and there is likely little you can do except wait with the infinite patience.

Syntax

There are several command-line switches that can be used with TRACERT, but they are usually not needed for standard troubleshooting.

TRACERT syntax:

tracert [-d] [-h maximum_hops] [-w timeout] target_name

Parameters:

-d Specifies to not resolve addresses to host names.

-h maximum_hops Specifies the maximum number of hops to search for target.

-w timeout Waits the number of milliseconds specified by timeout for each reply.

Target_name Name or IP address of the target host.

• Pathping

This command combines functions of *Ping* and *Tracert*. *Pathping* will first list the number of hops required to reach the address you are testing and then send multiple pings to each router between you and the destination. After that, it computes results based on the packets returned from each router. Because pathping displays the degree of packet loss at any given router or link, you can determine which routers or subnets might be having network problems. Note that the whole process may consume 5-10 minutes because many pings are being sent. There are switches to modify the process and these can be seen by entering "pathping /?" in the command prompt.

- ## Netstat

Displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, **netstat** displays active TCP connections.

Syntax

netstat [**-a**] [**-e**] [**-n**] [**-o**] [**-p** *Protocol*] [**-r**] [**-s**] [*Interval*]

Parameters

-a : Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.

-e : Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with **-s**.

-n : Displays active TCP connections, however, addresses and port numbers are expressed numerically and no attempt is made to determine names.

-o : Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the **Processes** tab in Windows Task Manager. This parameter can be combined with **-a**, **-n**, and **-p**.

-p Protocol : Shows connections for the protocol specified by *Protocol*. In this case, the Protocol can be **tcp**, **udp**, **tcpv6**, or **udpv6**. If this parameter is used with **-s** to display statistics by protocol, Protocol can be **tcp**, **udp**, **icmp**, **ip**, **tcpv6**, **udpv6**, **icmpv6**, or **ipv6**.

-s : Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The **-p** parameter can be used to specify a set of protocols.

-r : Displays the contents of the IP routing table. This is equivalent to the **route print** command.

***Interval* :** Redisplays the selected information every *Interval* seconds. Press CTRL+C to stop the redisplay. If this parameter is omitted, **netstat** prints the selected information only once.

/? : Displays help at the command prompt.

Examples

To display both the Ethernet statistics and the statistics for all protocols, type the following command:

- **netstat -e -s**

To display the statistics for only the TCP and UDP protocols, type the following command:

- **netstat -s -p tcp udp**

To display active TCP connections and the process IDs every 5 seconds, type the following command:

- **netstat -o 5**

To display active TCP connections and the process IDs using numerical form, type the following command:

- **netstat -n –o**

- **Nslookup**

Displays information that you can use to diagnose Domain Name System (DNS) infrastructure. Before using this tool, you should be familiar with how DNS works. The Nslookup command-line tool is available only if you have installed the TCP/IP protocol.

Syntax

```
nslookup [-SubCommand ...] [{ComputerToFind| [-Server]}]
```

EXERCISE

1. Test the reach ability towards a PC [   ] and limit the number of echos to 5.

____

2. Find the route from your PC to Hotmail i.e [www.hotmail.com]

____
____
____
____
____

3. Open a browser connection to http server [www.yahoo.com] and write down the outcome of the command 'netstat -a'.

4. How would you know about the TCP/IP configuration on your PC, write the command and the respective output.

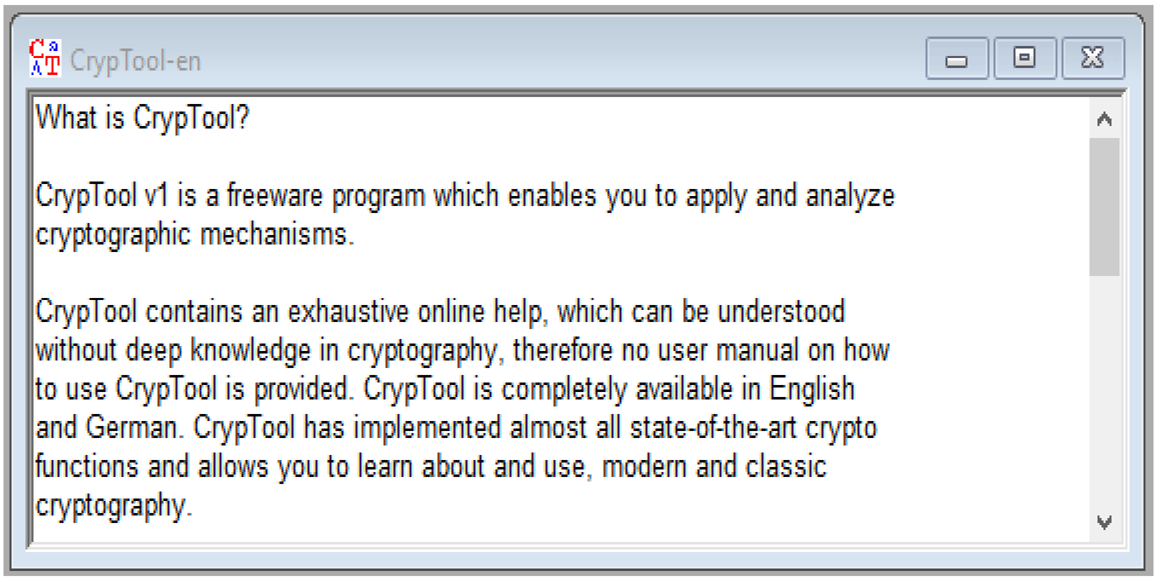
5. Which command we use to display active TCP connections and the process IDs using numerical form.

Lab Session 02

OBJECT: Implementation of Caesar Cipher and Rot-13.

This lab provides an example illustrating the use of the Caesar encryption algorithm. To make it easier to follow the steps that need to be performed with CrypTool, the example is illustrated with a number of screenshots.

First of all, to acquaint ourselves with the Caesar encryption algorithm we will open a document, encrypt it and then decrypt it again. We will then try to get the computer to work out the key with which a plaintext is encrypted. Therefore, we open the file CrypTool-en.txt out of the directory CrypTool\examples via the menu File \ Open.

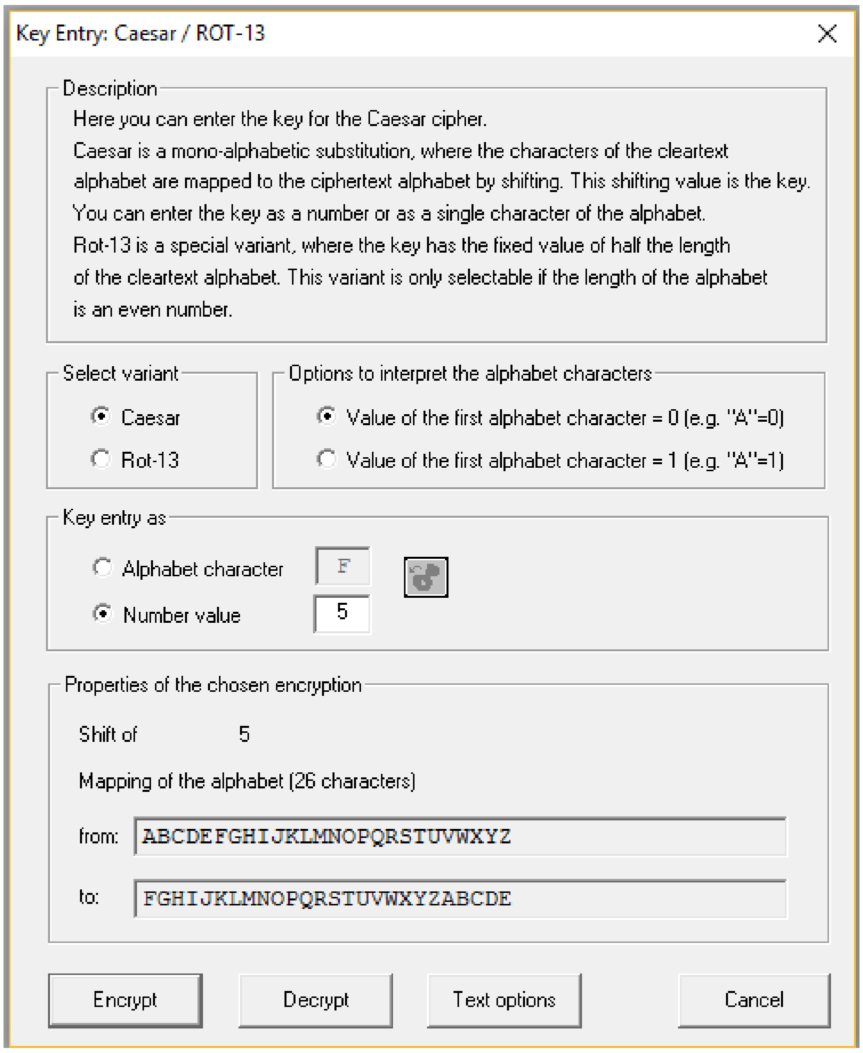


What is CrypTool?

CrypTool v1 is a freeware program which enables you to apply and analyze cryptographic mechanisms.

CrypTool contains an exhaustive online help, which can be understood without deep knowledge in cryptography, therefore no user manual on how to use CrypTool is provided. CrypTool is completely available in English and German. CrypTool has implemented almost all state-of-the-art crypto functions and allows you to learn about and use, modern and classic cryptography.

We will now encrypt this document using the Caesar encryption algorithm. To do this, we select the menu Encrypt/Decrypt \ Symmetric (classic) \ Caesar/Rot-13. Then the following which this dialog box appears. The key we enter is the letter F. Additionally, we change the options how to interpret the alphabet characters: The first alphabet character is set to 0.



Key Entry: Caesar / ROT-13

Description
Here you can enter the key for the Caesar cipher.
Caesar is a mono-alphabetic substitution, where the characters of the cleartext alphabet are mapped to the ciphertext alphabet by shifting. This shifting value is the key.
You can enter the key as a number or as a single character of the alphabet.
Rot-13 is a special variant, where the key has the fixed value of half the length of the cleartext alphabet. This variant is only selectable if the length of the alphabet is an even number.

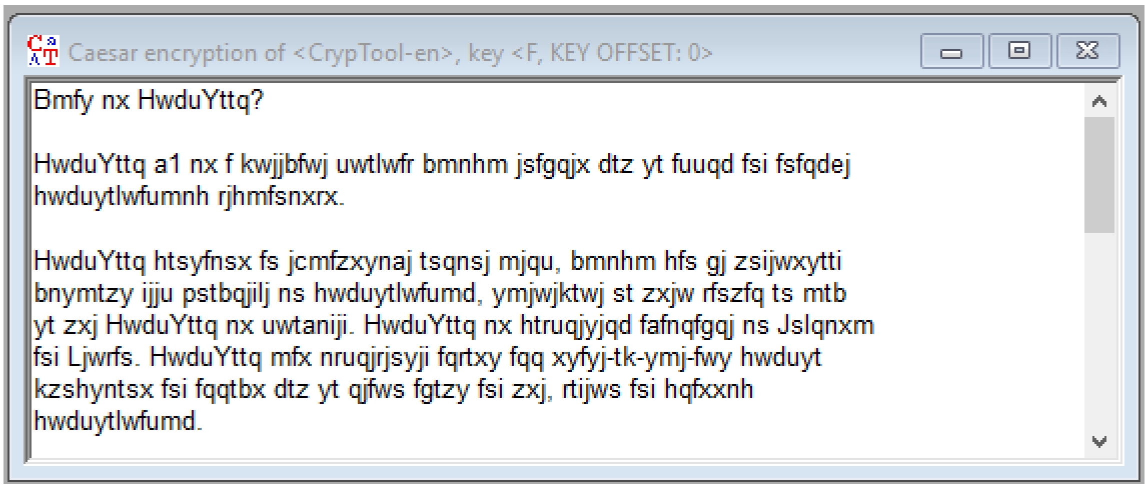
Select variant: [✓ Caesar] Rot-13

Options to interpret the alphabet characters: [✓ Value of the first alphabet character = 0 (e.g. "A"=0)] Value of the first alphabet character = 1 (e.g. "A"=1)

Key entry as: Alphabet character F; [✓ Number value] 5

Properties of the chosen encryption
Shift of 5
Mapping of the alphabet (26 characters)
from: ABCDEFGHIJKLMNOPQRSTUVWXYZ
to: FGHIJKLMNOPQRSTUVWXYZABCDE

Clicking on the **Encrypt** button opens a new window that contains the encrypted text. On closer examination of the text it becomes apparent that the letters have been shifted by 5 position, so that the initial word, CrypTool, is now HwduYttq.



Caesar encryption of <CrypTool-en>, key <F, KEY OFFSET: 0>

```
Bmfy nx HwduYttq?

HwduYttq a1 nx f kwjjbfwj uwtlwfr bmnhm jsfgqjx dtz yt fuuqd fsi fsfqdej
hwduytlwfumnh rjhmfsnxrx.

HwduYttq htsyfnsx fs jcmfzxynaj tsqnsj mjqu, bmnhm hfs gj zsijwxytti
bnymtzy ijju pstbqjilj ns hwduytlwfumd, ymjwjktwj st zxjw rfszfq ts mtb
yt zxj HwduYttq nx uwtaniji. HwduYttq nx htruqjyjqd fafnqfgqj ns Jslqnxm
fsi Ljwrfs. HwduYttq mfx nruqjrjsyji fqrtxy fqq xyfyj-tk-ymj-fwy hwduyt
kzshyntsx fsi fqqtbx dtz yt qjfws fgtzy fsi zxj, rtijws fsi hqfxxnh
hwduytlwfumd.
```

The plaintext version of this encrypted document can now be obtained by selecting Encrypt/Decrypt \ Classical \ Caesar again. In the dialog box which now appears we enter the key with which the document was encrypted (F). This time we do not want the text to be encrypted, but instead to be decrypted. Therefore the **Decrypt** button must be selected.

Clicking on the **Decrypt** button tells CrypTool to go ahead and decrypt the text. The plaintext appears immediately. We have seen how a text is encrypted using the Caesar encryption algorithm and then decrypted again.

Under the classical Caesar encryption algorithm only the letters are encrypted. During encryption, the lower case letters were converted to upper case letters and encrypted, but all the other characters such as punctuation characters and formatting characters (blank characters and line breaks) were omitted.

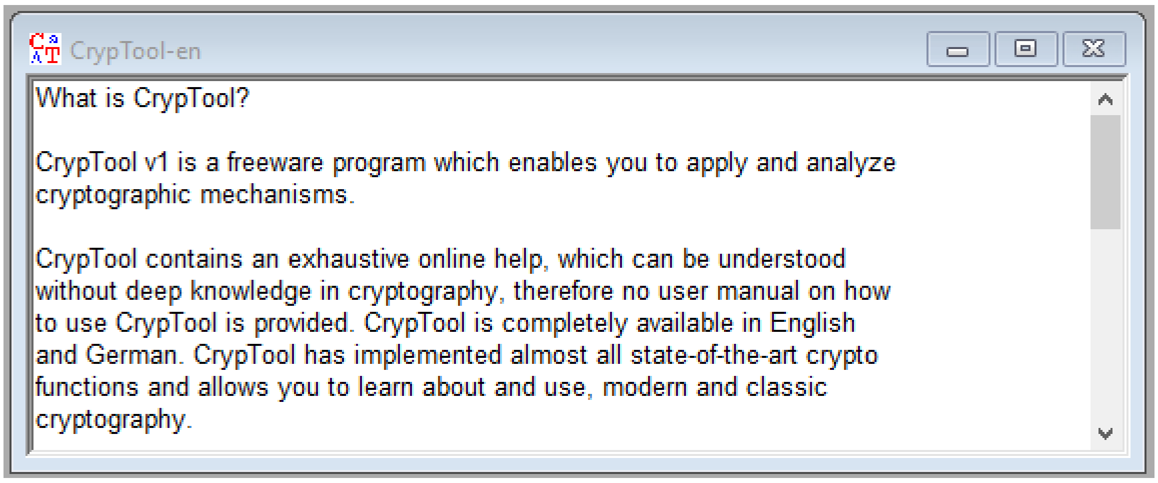
In the classical encryption algorithms CrypTool retains the formatting by default. This can be disabled via the menu option Options \ Text Options. In the following dialog box the option Keep characters not present in the alphabet unchanged and the option Distinguish between uppercase and lowercase must be disabled.

We now return to the window containing the plaintext (by clicking on it with the mouse) and encrypt this document once again with the Caesar encryption algorithm, using the key F. From the results one can see immediately that the encrypted text has the same content as the previous version but this time it contains only upper case letters and the formatting has been removed. For example, the first characters in the encrypted text are once again HWDUYTTQ (the ciphertext of CrypTool).

We would like to save this document under a different name. Select File \ Save As to access the dialog box used for saving a document under a new name. As file name we enter CrypTool.Caesar.txt.

The decryption operation works in exactly the same way as described above. We decrypt this document once again and the plaintext version is restored. This of course now consists only of a sequence of upper case letters. Because no distinction is made any longer between upper and lower case letters, and especially because of the absence of formatting, this text is more difficult to read. However, if we compare the document briefly with the original text, we can see that it is the same text. The original text begins with the words, ”CrypTool is a program which will enable you …” These words appear once again in the first line of the decrypted text.

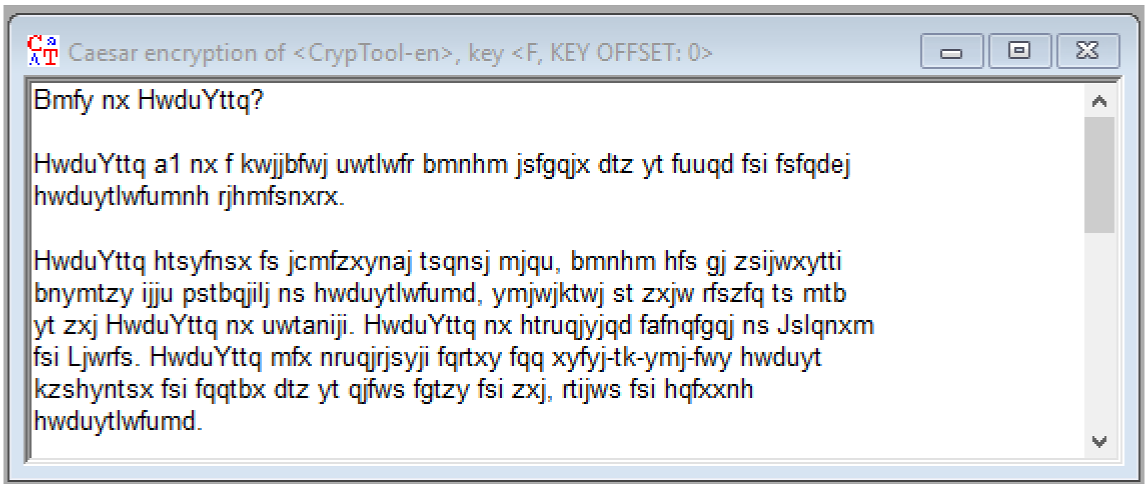
Having now learned the steps involved in encrypting and decrypting a document, we would like to take a look at the security of the Caesar encryption algorithm. To do this, all the windows should be closed apart from the window containing the plaintext and the version of the encrypted text in which the formatting is retained. The only two windows now open should be the following:



What is CrypTool?

CrypTool v1 is a freeware program which enables you to apply and analyze cryptographic mechanisms.

CrypTool contains an exhaustive online help, which can be understood without deep knowledge in cryptography, therefore no user manual on how to use CrypTool is provided. CrypTool is completely available in English and German. CrypTool has implemented almost all state-of-the-art crypto functions and allows you to learn about and use, modern and classic cryptography.



Bmfy nx HwduYttq?

HwduYttq a1 nx f kwjjbfwj uwtlwfr bmnhm jsfgqjx dtz yt fuuqd fsi fsfqdej hwduytlwfumnh rjhmfsnxrx.

HwduYttq htsyfnsx fs jcmfzxynaj tsqnsj mjqu, bmnhm hfs gj zsijwxytti bnymtzy ijju pstbqjilj ns hwduytlwfumd, ymjwjktwj st zxjw rfszfq ts mtb yt zxj HwduYttq nx uwtaniji. HwduYttq nx htruqjyjqd fafnqfgqj ns Jslqnxm fsi Ljwrfs. HwduYttq mfx nruqjrjsyji fqrtxy fqq xyfyj-tk-ymj-fwy hwduyt kzshyntsx fsi fqqtbx dtz yt qjfws fgtzy fsi zxj, rtijws fsi hqfxxnh hwduytlwfumd.

First, make the window containing the plaintext the active window again by clicking on it with the mouse. We now have the frequency distribution of the letters (Analysis \ General \ Histogram) calculated. We repeat the same procedure after making the other window active.

When we examine the histogram of the encrypted document we can see that the letter frequencies have merely been shifted by five position. That means that this old encryption algorithm is not secure. Moreover there are only 26 possible keys. The Caesar encryption algorithm can be broken easily by a ciphertext-only attack. To perform such an attack, restore the window containing the encrypted text to the active window and select Analysis \ Ciphertext-only \ Caesar. The text will automatically be analyzed.

By analyzing the superposition it is possible to discover the key which was used to encrypt this document. In this case it was the letter F.When you click on the **Decrypt** button in the message window, the plaintext appears, i.e. the text that has been decrypted with the key F that was discovered. So CrypTool managed successfully to find the key with which the document had been encrypted. The only information it needed to do this was the encryption algorithm.

EXERCISE

6. Explain the working of caesar cipher and rot-13.

7. What is the difference between rot-13 and caesar. How can caesar cipher become rot-13?

8. Break the encrypted text "pgf wpkxgtukva", what is the key?

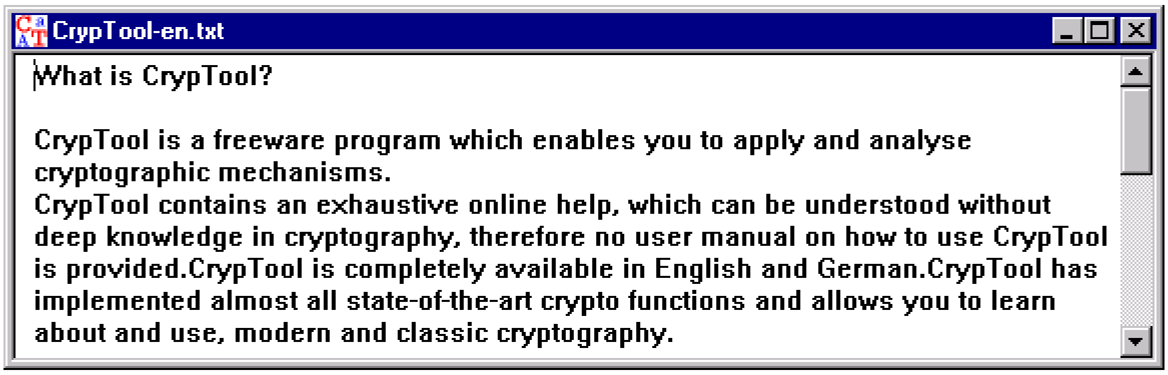
9. Why rot-13 is considered an inverse of itself? Encrypt the same text twice using the same key and see the result?

10. What does frequency analysis tell us about any encryption algorithm knowing that some letters are more common in the English language than others? How can this be used in cryptanalysis? (Take the default text in cryptool and run analysis->tools->histogram)

Lab Session 03

OBJECT: **Implementation of Vigenere Cipher and Atbash.**

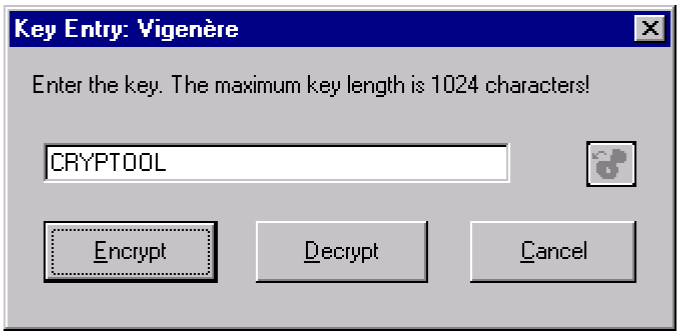
First of all, to acquaint ourselves with the Vigenère encryption algorithm we will open a document, encrypt it and then decrypt it again. We will then try to get the computer to work out the key with which a plaintext is encrypted. To this end we open a document which contains a part of the CrypTool Help text ”Introduction to CrypTool”. The file name of the document to be opened is CrypTool-en.txt. This file is opened via the menu selection File \ Open.



What is CrypTool?

CrypTool is a freeware program which enables you to apply and analyse cryptographic mechanisms.
CrypTool contains an exhaustive online help, which can be understood without deep knowledge in cryptography, therefore no user manual on how to use CrypTool is provided.CrypTool is completely available in English and German.CrypTool has implemented almost all state-of-the-art crypto functions and allows you to learn about and use, modern and classic cryptography.

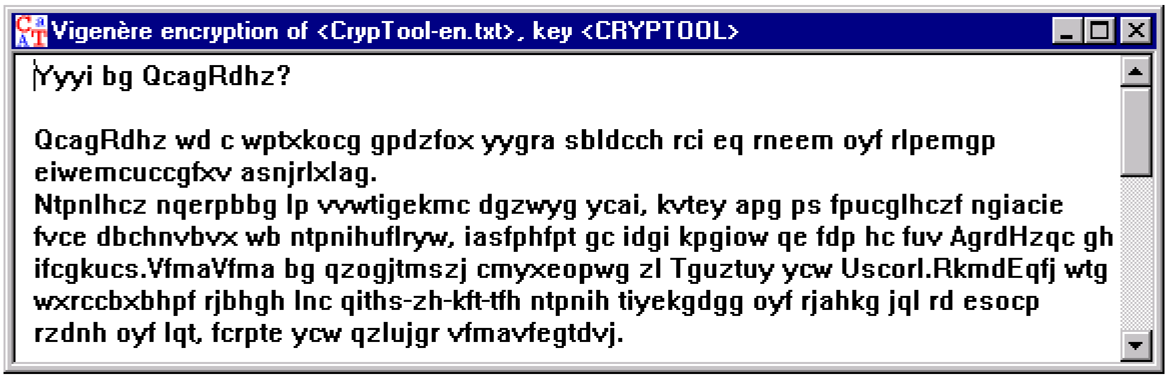
We now want to encrypt this document using the Vigenère encryption algorithm. To do this, we select Encrypt/Decrypt \ Classical \ Vigenère, following which this dialog box appears. As key, we enter CRYPTOOL.



Enter the key. The maximum key length is 1024 characters!

CRYPTOOL

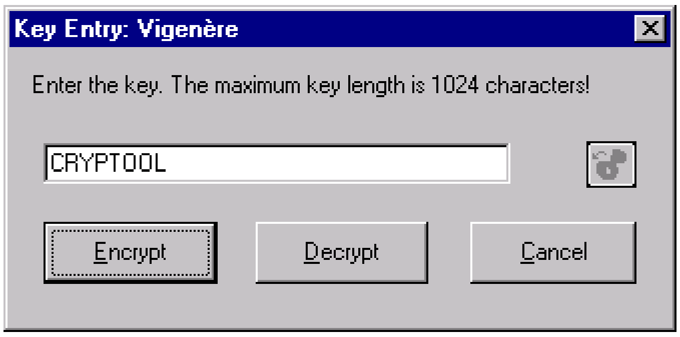
Clicking on the Encrypt button opens a new window that contains the encrypted text.



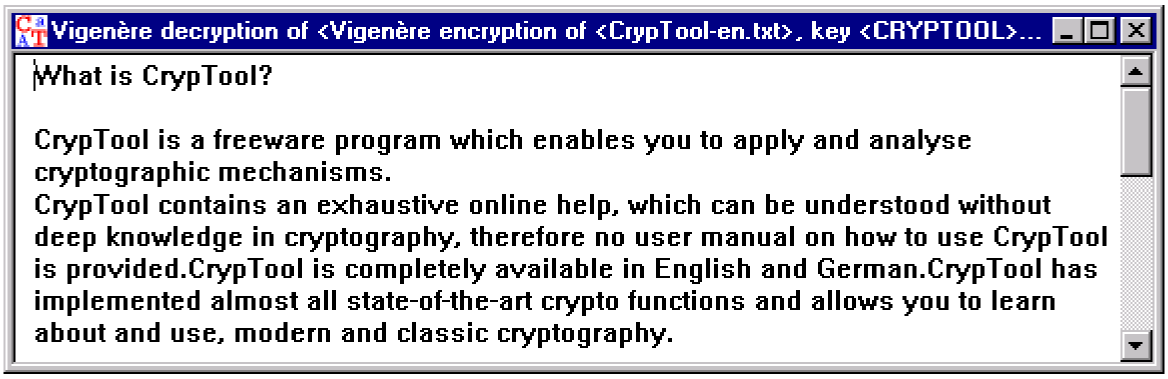
Yyyi bg QcagRdhz?

QcagRdhz wd c wptxkocg gpdzfox yygra sbldcch rci eq rneem oyf rlpemgp eiwemcuccgfxv asnjrlxlag.
Ntpnlhcz nqerpbbg lp vvwtigekmc dgzwyg ycai, kvtey apg ps fpucglhczf ngiacie fvce dbchnvbvx wb ntpnihuflryw, iasfphfpt gc idgi kpgiow qe fdp hc fuv AgrdHzqc gh ifcgkucs.VfmaVfma bg qzogjtmszj cmyxeopwg zl Tguztuy ycw Uscorl.RkmdEqfj wtg wxrccbxbhpf rjbhgh lnc qiths-zh-kft-tfh ntpnih tiyekgdgg oyf rjahkg jql rd esocp rzdnh oyf lqt, fcrpte ycw qzlujgr vfmavfegtdvj.

The plaintext version of this encrypted document can now be obtained simply by selecting Encrypt/Decrypt \ Classical \ Vigenère. In the dialog box which then appears we enter the key with which the document was encrypted (CRYPTOOL). This time we do not want the text to be encrypted, but instead to be decrypted. Therefore the Decrypt field must be selected. This is done by clicking on the field itself or on the radial button to the left of the field.



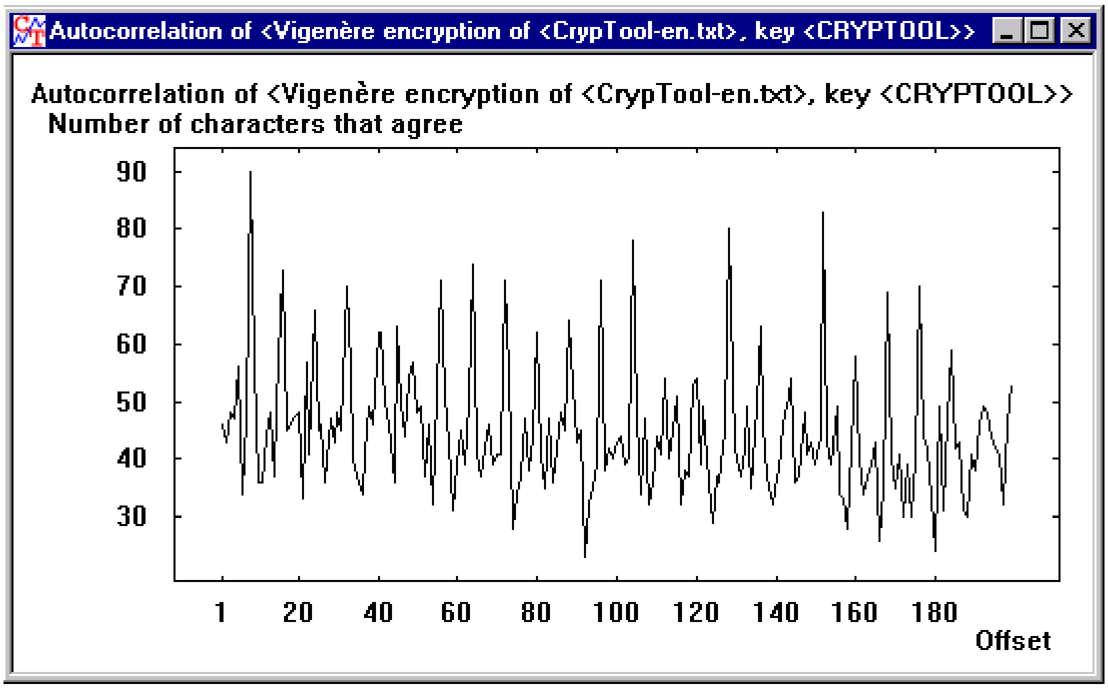
Clicking on the Decrypt button instructs CrypTool to decrypt the text. The plaintext appears immediately.


Vigenère decryption of <Vigenère encryption of <CrypTool-en.txt>, key <CRYPTOOL>...

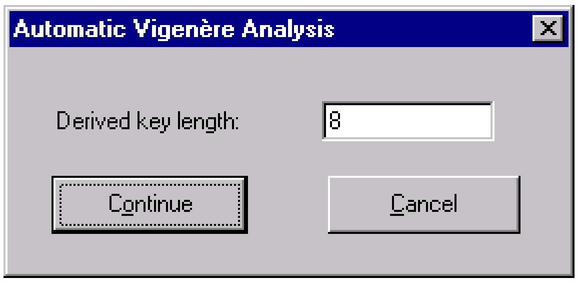
What is CrypTool?

CrypTool is a freeware program which enables you to apply and analyse cryptographic mechanisms.
CrypTool contains an exhaustive online help, which can be understood without deep knowledge in cryptography, therefore no user manual on how to use CrypTool is provided.CrypTool is completely available in English and German.CrypTool has implemented almost all state-of-the-art crypto functions and allows you to learn about and use, modern and classic cryptography.

We now wish to find the key with which the document was encrypted. Restore the window containing the encrypted document (via a mouse click) and select Analysis \ ciphertext only \ Vigenère, following which the text will automatically be analyzed. First of all a new, autocorrelation window opens.

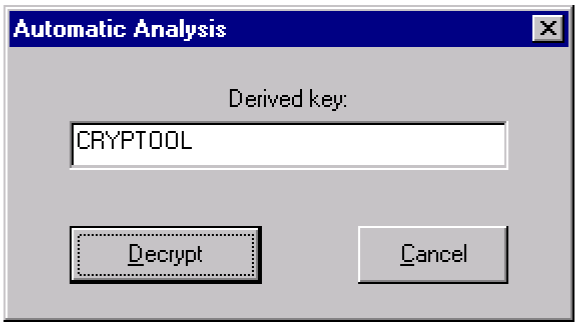


It is possible to work out the key length from the regular peaks in the autocorrelation. It is correctly calculated at 8 characters.



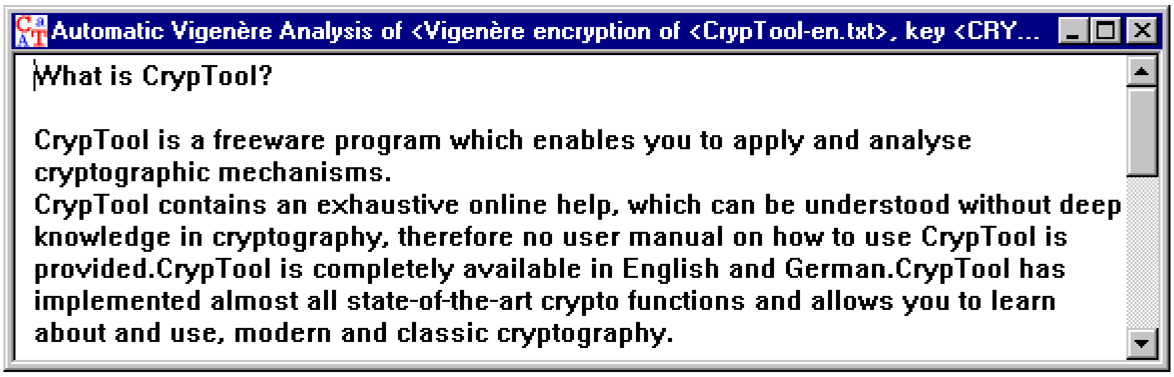
Derived key length: 8

When the Continue button is clicked, another dialog box opens to display the key that has been computed.



Derived key: CRYPTOOL

Clicking on Decrypt closes this dialog box and opens another window which contains the unencrypted text.


Automatic Vigenère Analysis of <Vigenère encryption of <CrypTool-en.txt>, key <CRY...

What is CrypTool?

CrypTool is a freeware program which enables you to apply and analyse cryptographic mechanisms.
CrypTool contains an exhaustive online help, which can be understood without deep knowledge in cryptography, therefore no user manual on how to use CrypTool is provided.CrypTool is completely available in English and German.CrypTool has implemented almost all state-of-the-art crypto functions and allows you to learn about and use, modern and classic cryptography.

As you can see, the distinction between upper and lower case and the formatting (blank characters, line breaks etc.) have been retained. Special characters such as umlauts, hyphens and slashes were kept on in the encrypted document. Normally in a classical encryption algorithm all formatting is removed and only the letters are encrypted, but to make the text easier for you to read and compare, CrypTool retains the formatting in the encrypted text (and hence also in the decrypted text). By disabling the formatting options accessed via Options \ Text Options, it is possible to disable retention of formatting and, by checking the relevant boxes again, to reinstate formatting.

EXERCISE

11. Explain the working of the vigenere cipher and the term polyalphabetic substitution. Encrypt the text "NED University" with a key "karachi". what is the result.

____

____

____

____

____

12. Decrypt the text "LXFOPVEFRNHR" using the key "LEMONLEMONLE" and explain the process.

____

____

____

____

____

13. Using brute force decrypt the text "ZL ANZR VF XUNA NAQ V NZ ABG N GREEBEVFG" using vignere cipher where the key is a single letter.

14. How would you define the vigenere cipher with respect to caesar cipher where the key length is 1.

15. What is atbash substitution, how does it work? Is the atbash a weak or strong cipher and why?

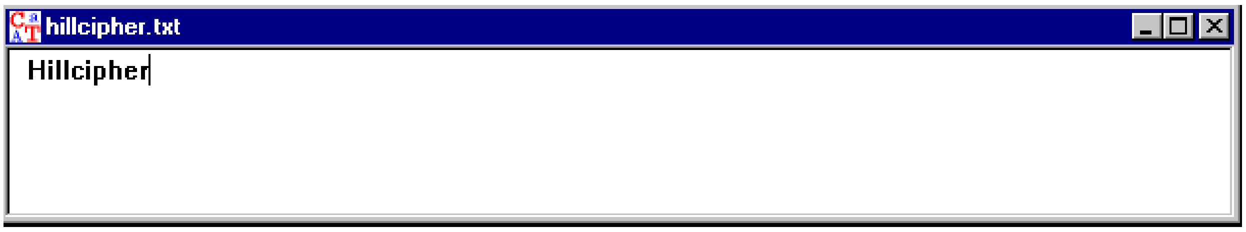
16. A few English words 'Atbash' into other English words. For example, "hob"="sly", "hold"="slow","holy"="slob","horn"="slim","zoo"="all","irk"="rip","low"="old","glow"="told", and "grog"="tilt." Some other English words Atbash into their own reverses, e.g., "wizard" = "draziw. Prove this using cryptool and state the advantage or disadvantage of this?

17. If we go in to the atbash menu and select key entry: remaining characters are filed in ascending order, what is the effect of choosing a key 'LEMON' in one instance and in another instance the key 'ZXC'. Which key is stronger?

Lab Session 04

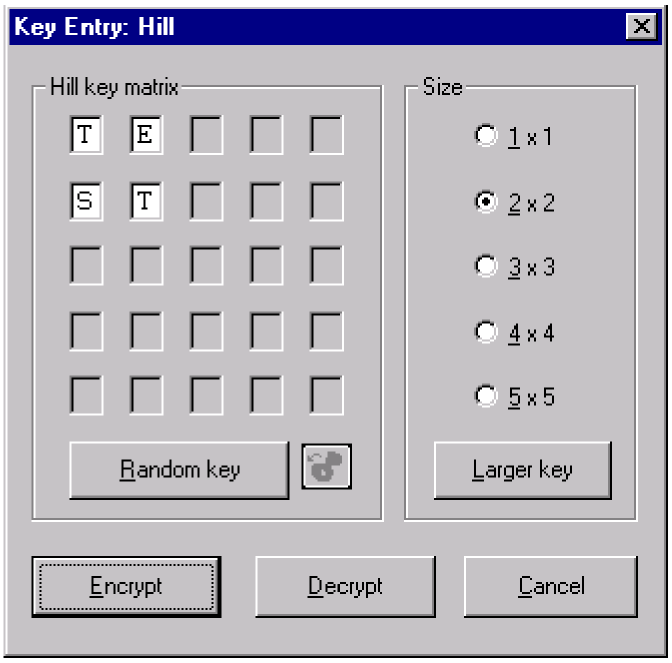
OBJECT: Implementation of Hill Cipher.

The text HILLCIPHER is to be encrypted with the key T E S T. To do this, a new document is created (via File \ New) and the text HILLCIPHER is entered. Under the classical encryption algorithms lower case letters are always replaced by the corresponding upper case letters. However, CrypTool keeps upper and lower case letters distinct in order to make the text easier for you to read. These two aspects can be disabled independently of each other via the options Keep characters not present in the alphabet unchanged and Keep uppercase / lowercase (if possible) in the Text Options dialog box. Therefore it does not matter whether the text contains lower case letters. To demonstrate this we will use Hillcipher instead of HILLCIPHER:



Hillcipher

We now select Encrypt/Decrypt \ Classical \ Hill, following which this dialog box appears:

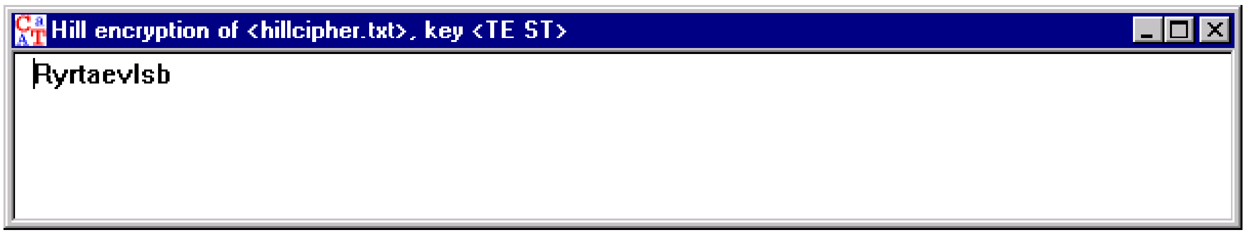


Hill key matrix

T	E			
S	T			

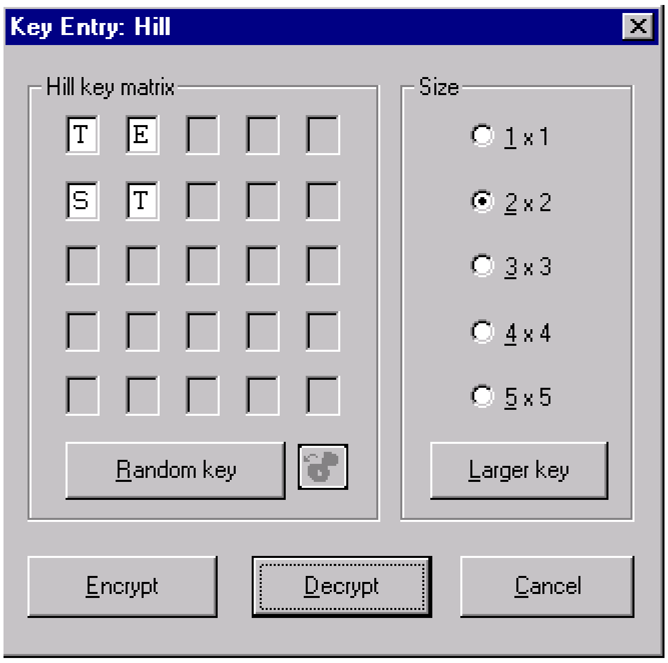
Size: 1 x 1 [✓ 2 x 2] 3 x 3 4 x 4 5 x 5

After entering the key T E S T it can be saved in the clipboard (in order to avoid having to enter it again later on at the decryption stage) by clicking on the Copy button. Information on the clipboard will be found in the Help facility on the toolbar. Now click on the Encrypt button, and a new window that contains the encrypted text opens.


Hill encryption of <hillcipher.txt>, key <TE ST>

Ryrtaevlsb

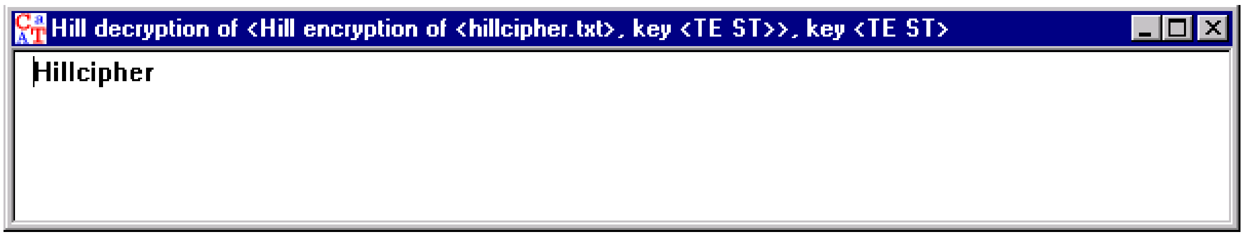
The encrypted text is now decrypted for checking purposes. The decryption procedure is identical to encryption, except that this time the Decrypt option must be selected, i.e. the radial button next to that option must be selected. This is done by clicking on the field with the mouse. The key can now either be typed in manually as before or else it can be inserted from the clipboard using the Paste button if it had previously been copied.


Key Entry: Hill

Hill key matrix: T E / S T

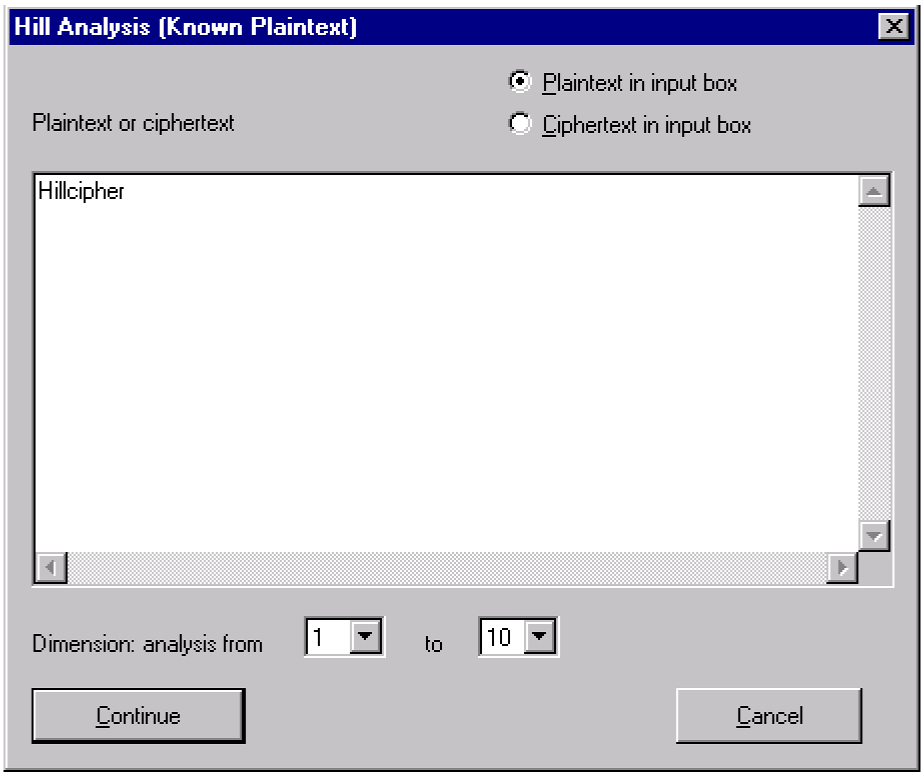
Size: ( ) 1 x 1 [✓ 2 x 2] ( ) 3 x 3 ( ) 4 x 4 ( ) 5 x 5

Now click on Decrypt, and the plaintext reappears.


Hill decryption of <Hill encryption of <hillcipher.txt>, key <TE ST>>, key <TE ST>

Hillcipher

CrypTool will now work out the key using a Known Plaintext attack. Such an attack requires that both the plaintext and the encrypted text are available. The easiest approach is to copy the plaintext (mark with the mouse pointer and then press the key combination Ctrl+C) from one of the two windows which contain Hillcipher to the clipboard. Now make the window containing the encrypted text the active window again (by clicking on it with the mouse) and select the menu option Analysis \ Known Plaintext \ Hill to view the Analysis Hill cipher window. The plaintext is then pasted from clipboard into this window (using the key combination Ctrl+V).

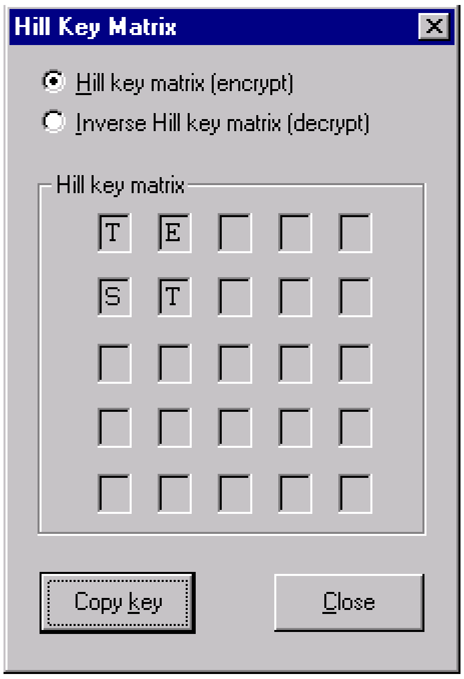


[✓ Plaintext in input box] Ciphertext in input box

Hillcipher

Dimension: analysis from 1 to 10

The analysis starts when the Continue button is clicked and the key T E S T appears in the following window.



[✓ Hill key matrix (encrypt)] Inverse Hill key matrix (decrypt)

Hill key matrix

T	E			
S	T			

In this way the key was worked out simply from knowing the plaintext and the corresponding encrypted text.

EXERCISE

1. Explain the working of the Hill cipher. Encrypt the text "Hello World" with a key K =$\begin{bmatrix} C & B \\ D & E \end{bmatrix}$. What is the result?

2. Decrypt the text "ZLBT" using the key K = $\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$ and explain the process.

3. Using analysis in Cryptool, apply the known plain text attack and find out the key of Hill Cipher. Known Plain Text “friday” and corresponding cipher text is “PQCFKU”

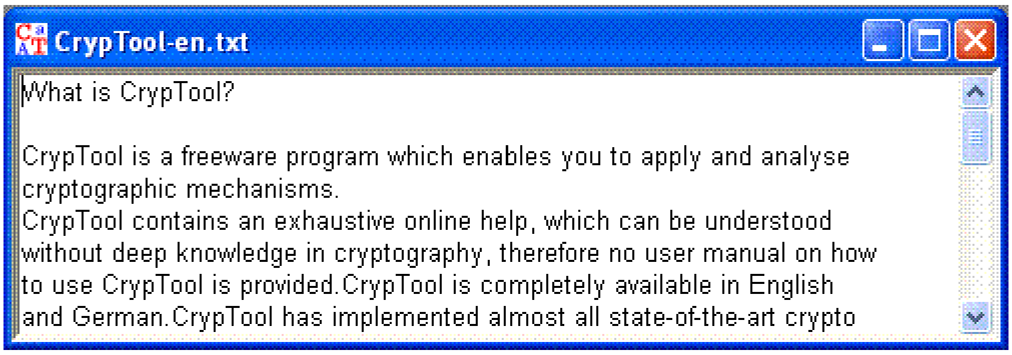
Lab Session 05

OBJECT: Implementation of Playfair Cipher.

We start with investigating the Playfair encryption algorithm by opening a document, encrypting it and decrypting it again. After that we will try to find the key that was used to encrypt the plaintext by using the manual Playfair analysis step by step.

1. Playfair Encryption

1.1 Open the document to be encrypted: e.g. CrypTool-en.txt, which is part of the online help section. You can open that document via the menu File \ Open.

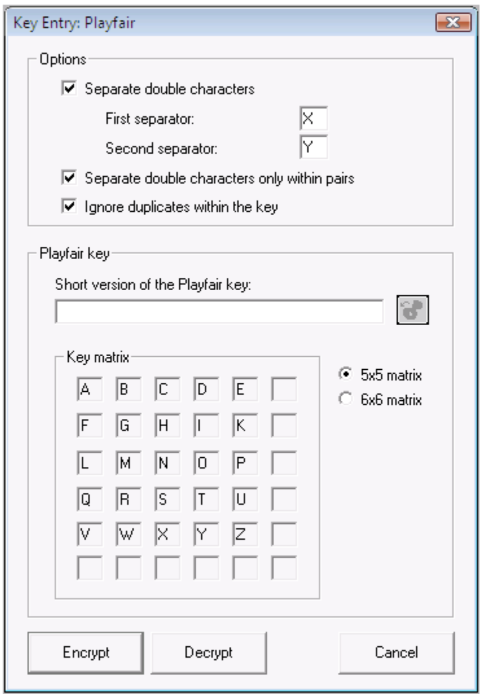

CrypTool-en.txt

What is CrypTool?

CrypTool is a freeware program which enables you to apply and analyse
cryptographic mechanisms.
CrypTool contains an exhaustive online help, which can be understood
without deep knowledge in cryptography, therefore no user manual on how
to use CrypTool is provided.CrypTool is completely available in English
and German.CrypTool has implemented almost all state-of-the-art crypto

1.2 Now we encrypt this document using the Playfair encryption algorithm.

From the menu, choose Encrypt/Decrypt \ Symmetric (classic) \ Playfair. The following dialog box is displayed:


Key Entry: Playfair

Options
- [✓] Separate double characters
  - First separator: X
  - Second separator: Y
- [✓] Separate double characters only within pairs
- [✓] Ignore duplicates within the key

Playfair key

Short version of the Playfair key:

Key matrix

A	B	C	D	E	
F	G	H	I	K	
L	M	N	O	P	
Q	R	S	T	U	
V	W	X	Y	Z	

[✓ 5x5 matrix] 6x6 matrix

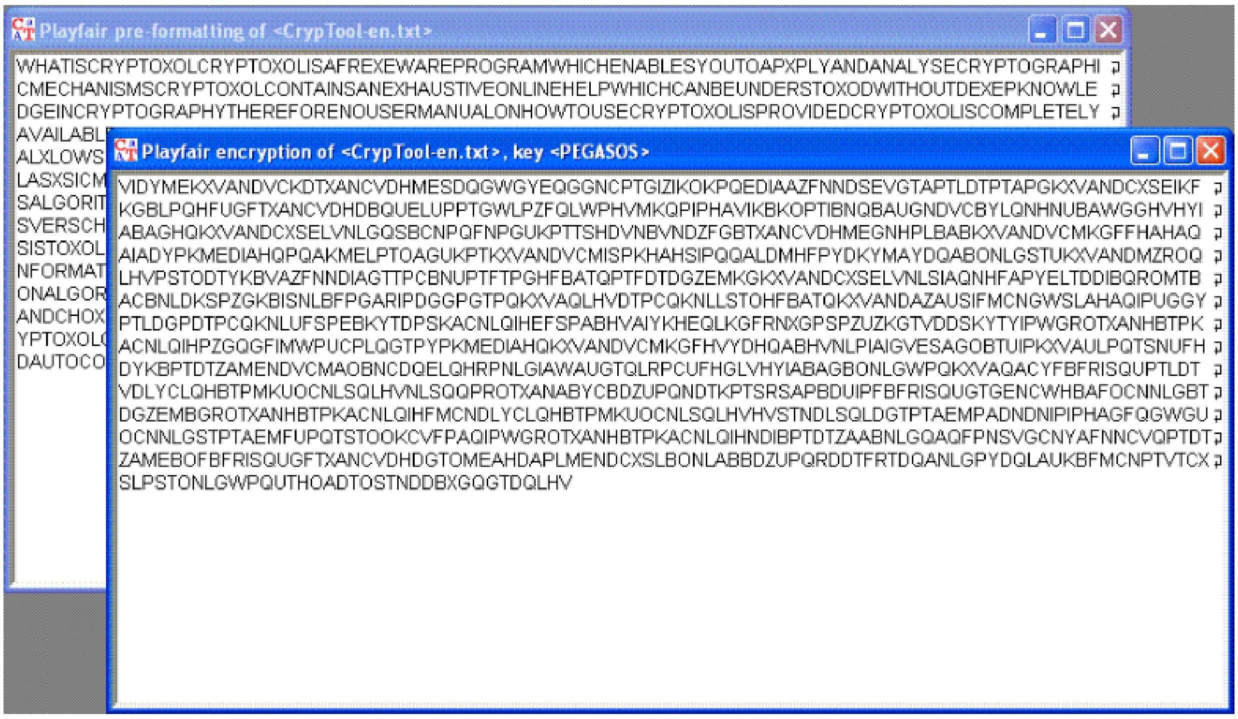
Using the standard 5x5 key matrix, the Playfair alphabet consists of the normal capital letters A to Z, where J is taken out.

1.3 Please use PEGASOS as the key.

From this key the following Playfair key matrix will be calculated:

P E G A S
O B C D F
H I K L M
N Q R T U
V W X Y Z

1.4 When you push the button Encrypt the Playfair key entry dialog will be closed and two windows are opened: one with the pre-processed plaintext and another with the encrypted text:



```
WHATISCRYPTOXOLCRYPTOXOLISAFREXEWAREPROGRAMWHICHENABLESYOUTOAPXPLYANDANALYSECRYPTOGRAPHI
CMECHANISMSCRYPTOXOLCONTAINSANEXHAUSTIVEONLINEHELPWHICHCANBEUNDERSTOXODWITHOUTDEXEPKNOWLE
DGEINCRYPTOGRAPHYTHEREFORENOUSERMANUALONHOWTOUSECRYPTOXOLISPROVIDEDCRYPTOXOLISCOMPLETELY
AVAILABL
ALXLOWS
LASXSICM
SALGORIT
SVERSCH
SISTOXOL
NFORMAT
ONALGOR
ANDCHOX
YPTOXOL
DAUTOCO
```

```
VIDYMEKXVANDVCKDTXANCVDHMESDQGWGYEQGGNCPTGIZIKOKPQEDIAAZFNNDSEVGTAPTLDTPTAPGKXVANDCXSEIKF
KGBLPQHFUGFTXANCVDHDBQUELUPPTGWLPZFQLWPHVMKQPIPHAVIKBKOPTIBNQBAUGNDVCBYLQNHNUBAWGGHVHYI
ABAGHQKXVANDCXSELVNLGQSBCNPQFNPGUKPTTSHDVNBVNDZFGBTXANCVDHMEGNHPLBABKXVANDVCMKGFFHAHAQ
AIADYPKMEDIAHQPQAKMELPTOAGUKPTKXVANDVCMISPKHAHSIPQQALDMHFPYDKYMAYDQABONLGSTUKXVANDMZROQ
LHVPSTODTYKBVAZFNNDIAGTTPCBNUPTFTPGHFBATQPTFDTDGZEMKGKXVANDCXSELVNLSIAQNHFAPYELTDDIBQROMTB
ACBNLDKSPZGKBISNLBFPGARIPDGGPGTPQKXVAQLHVDTPCQKNLLSTOHFBATQKXVANDAZAUSIFMCNGWSLAHAQIPUGGY
PTLDGPDTPCQKNLUFSPEBKYTDPSKACNLQIHEFSPABHVAIYKHEQLKGFRNXGPSPZUZKGTVDDSKYTYIPWGROTXANHBTPK
ACNLQIHPZGQGFIMWPUCPLQGTPYPKMEDIAHQKXVANDVCMKGFHVYDHQABHVNLPIAIGVESAGOBTUIPKXVAULPQTSNUFH
DYKBPTDTZAMENDVCMAOBNCDQELQHRPNLGIAWAUGTQLRPCUFHGLVHYIABAGBONLGWPQKXVAQACYFBFRISQUPTLDT
VDLYCLQHBTPMKUOCNLSQLHVNLSQQPROTXANABYCBDZUPQNDTKPTSRSAPBDUIPFBFRISQUGTGENCWHBAFOCNNLGBT
DGZEMBGROTXANHBTPKACNLQIHFMCNDLYCLQHBTPMKUOCNLSQLHVHVSTNDLSQLDGTPTAEMPADNDNIPIPHAGFQGWGU
OCNNLGSTPTAEMFUPQTSTOOKCVFPAQIPWGROTXANHBTPKACNLQIHNDIBPTDTZAABNLGQAQFPNSVGCNYAFNNCVQPTDT
ZAMEBOFBFRISQUGFTXANCVDHDGTOMEAHDAPLMENDCXSLBONLABBDZUPQRDDTFRTDQANLGPYDQLAUKBFMCNPTVTCX
SLPSTONLGWPQUTHOADTOSTNDDBXGQGTDQLHV
```

- The pre-processing step removes all characters from the cleartext that are not part of the **Playfair alphabet** and inserts the character X between two equal characters in a row. E.g. the cleartext `CrypTool ist` (German for CrypTool is …) will be pre-processed to **`CRYPTOXOLIST`**:
- Based on the Playfair key matrix above, the pre-processed text can now be encrypted. The basic idea is that two consecutive characters (a di-gram) determine the two opposite edges of a rectangle in the key matrix.
  We consider *one dimensional* rectangles for the special case, when both characters are in the same row or column of the key matrix.

1.5 Now, let us manually encrypt the pre-processed plaintext **CRYPTOXOLIST**. The following key matrix rectangles have to be considered:

P	E	G	**A**	S
O	B	**C**	**D**	F
H	I	**K**	L	M
N	Q	**R**	**T**	U
V	W	**X**	**Y**	Z

< -- >

Plaintext di-gram	Rectangle	Dimension	Encryption
CR	**C,K,R,X**	1	KX
YP	**Y,V,P,A**	2	VA
TO	**T,N,O,D**	2	ND
XO	X,V,O,C	2	VC
LI	L,M,I,K	1	MK
ST	S,A,T,U	2	AU

For two dimensional rectangles we enumerate the edges as follows: we start with the first plaintext character, then choose the character in the other edge *on the same row*, then the second cleartext character. The last character is taken from the remaining edge.

In the less common case of one dimensional rectangles we also start with the first plaintext character, followed by the bottom resp. right neighbor, the second cleartext character and then its bottom resp. right neighbor.

The second and fourth character of the rectangle (the unused edges) are taken to be the encrypted di-gram, in the example above: KXVANDVCMKAU.

1.6 To proceed with the next step (decryption) you can avoid entering the key again by recovering it from the ciphertext window. To do this

- activate the window with the ciphertext
- click on the key icon or select menu Edit \ Show Key
- press **Copy key**

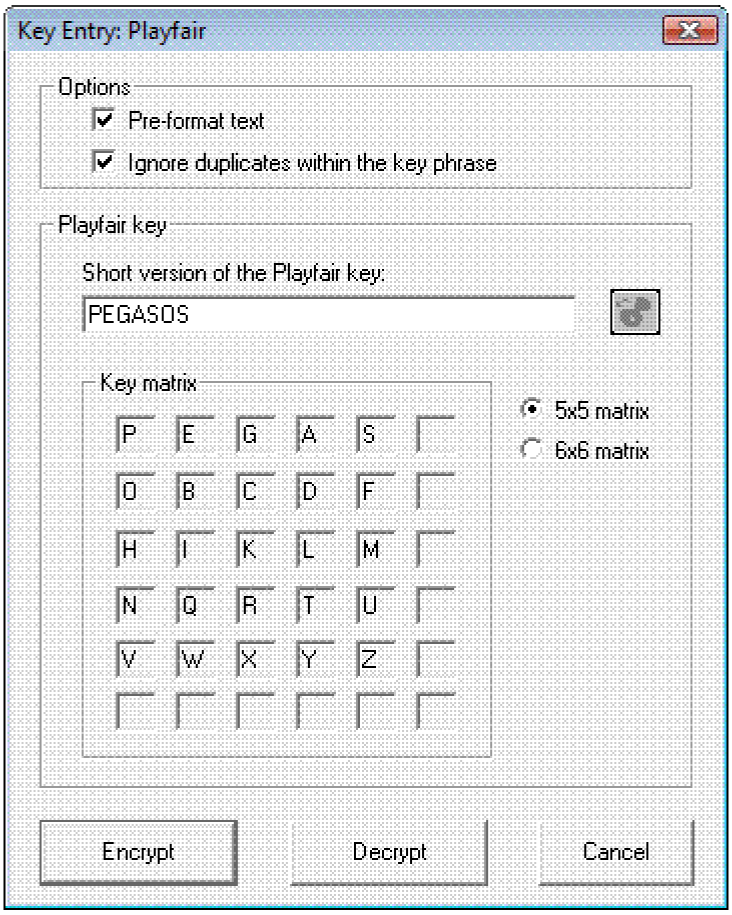
Now the key matrix is stored in CrypTool's key store.

2. Playfair Decryption

We now can recover the plaintext of the document encrypted in the previous step.

2.1 Select the menu Encrypt/Decrypt \ Symmetric (classic) \ Playfair.

2.2 Enter the encryption key (PEGASOS). If you have saved the key in the internal key storage then you can recover it by clicking on the key icon.



Key Entry: Playfair

Options
- [✓] Pre-format text
- [✓] Ignore duplicates within the key phrase

Playfair key

Short version of the Playfair key: PEGASOS

Key matrix

P	E	G	A	S	
O	B	C	D	F	
H	I	K	L	M	
N	Q	R	T	U	
V	W	X	Y	Z	

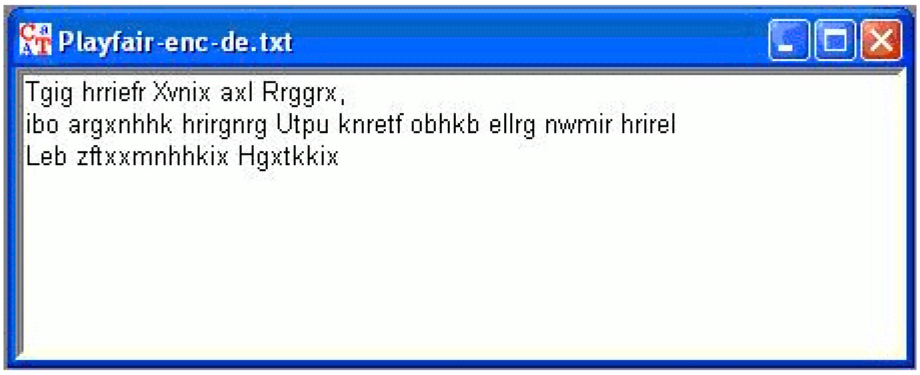
(● 5x5 matrix) ○ 6x6 matrix

2.3 Press the **Decrypt** button to recover the plaintext.

3. Manual analysis of the Playfair cipher

Let us now try to decipher a text encrypted with the Playfair cipher where we don't know the key (ciphertext-only analysis).

3.1 To do this we load the text to be decrypted from the file `Playfair-enc-de.txt` (this file can be found in the `examples` directory distributed with CrypTool):



Playfair-enc-de.txt

```
Tgig hrriefr Xvnix axl Rrggrx,
ibo argxnhhk hrirgnrg Utpu knretf obhkb ellrg nwmir hrirel
Leb zftxxmnhhkix Hgxtkkix
```

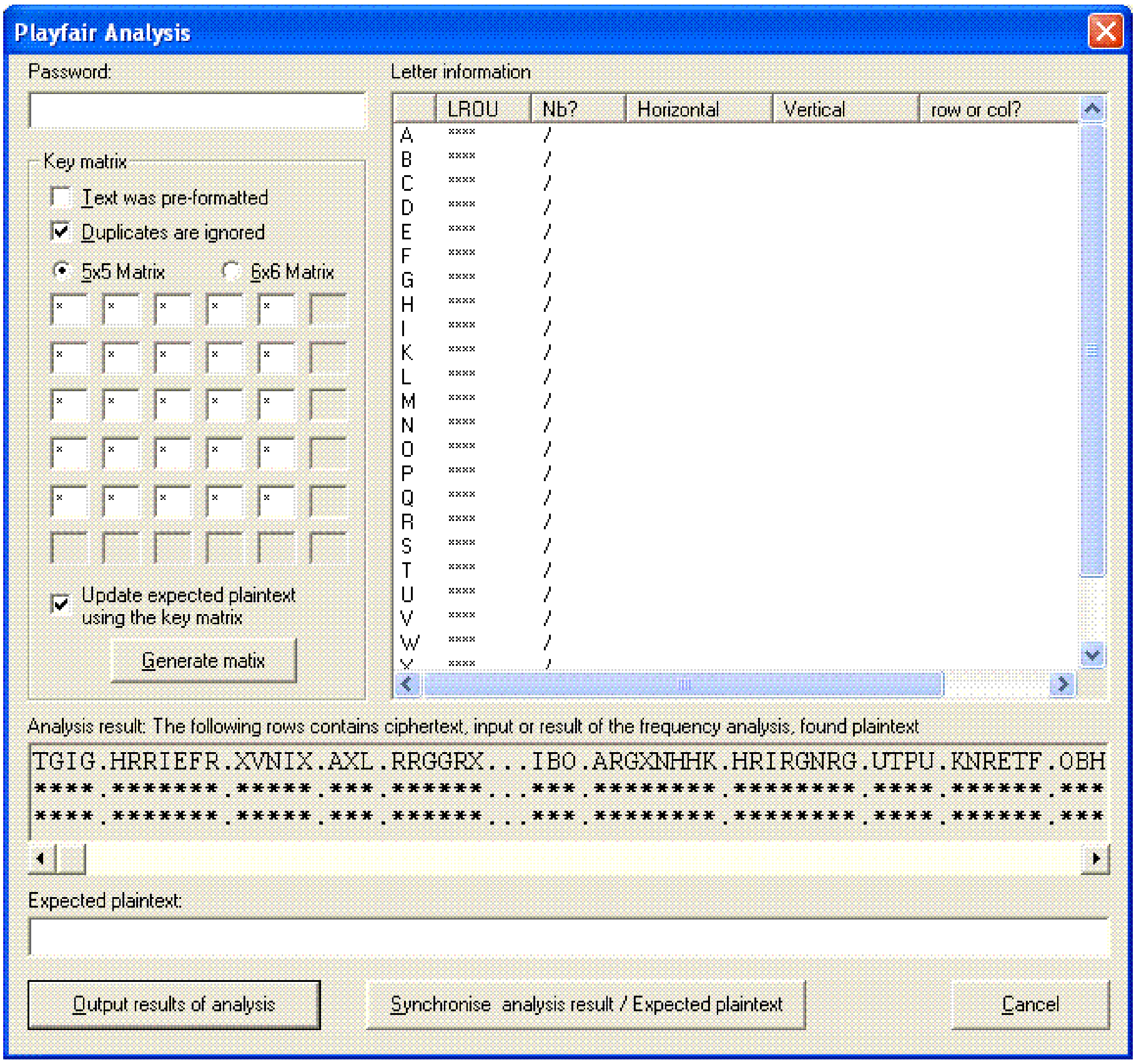
3.2 This text was created with the Playfair encryption algorithm *without* pre-processing: This option effectively leaves characters not contained in the Playfair alphabet unencrypted (e.g. blanks, punctuation marks, line feeds), so that the plaintext word and sentence boundaries are obvious.

Additionally, the insertion of the separating X between two equal, consecutive characters is switched off so double characters are left unencrypted.

3.3 CrypTool does only support a semi-manual analysis of the Playfair cipher. Therefore we need some clue about the document to be analyzed. Let us assume that the text is a German letter, which often start with the same words.

3.4 Let us now open the Playfair analysis dialog window by selecting the menu Analysis \ Symmetric Encryption (classic) \ Manual Analysis \ Playfair.

Here you see the text of the opened ciphertext file in the first row of the field "Analysis result".



Playfair Analysis
Password:
Key matrix
Text was pre-formatted
Duplicates are ignored
5x5 Matrix
6x6 Matrix
Update expected plaintext using the key matrix
Generate matix
Output results of analysis
Synchronise analysis result / Expected plaintext
Cancel

Key matrix: [ ] Text was pre-formatted; [✓] Duplicates are ignored; [● 5x5 Matrix] 6x6 Matrix; [✓] Update expected plaintext using the key matrix

Letter information

	LRОU	Nb?	Horizontal	Vertical	row or col?
A	****	/			
B	****	/			
C	****	/			
D	****	/			
E	****	/			
F	****	/			
G	****	/			
H	****	/			
I	****	/			
K	****	/			
L	****	/			
M	****	/			
N	****	/			
O	****	/			
P	****	/			
Q	****	/			
R	****	/			
S	****	/			
T	****	/			
U	****	/			
V	****	/			
W	****	/			
X	****	/			

Analysis result: The following rows contains ciphertext, input or result of the frequency analysis, found plaintext

```
TGIG.HRRIEFR.XVNIX.AXL.RRGGRX...IBO.ARGXNHHK.HRIRGNRG.UTPU.KNRETF.OBH
****.*******.*****.***.******...***.********.********.****.******.***
****.*******.*****.***.******...***.********.********.****.******.***
```

Expected plaintext:

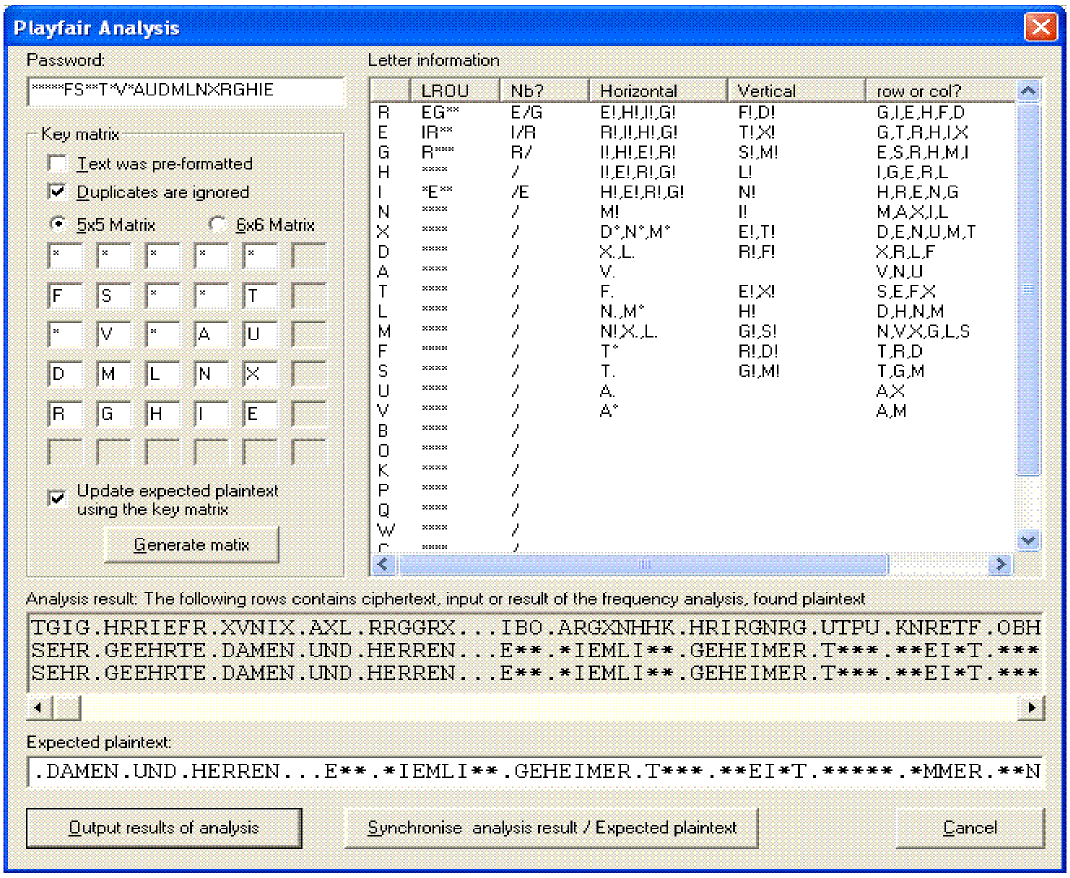
3.5 We assume: The letter probably begins with **Sehr geehrte Damen und Herren** (German for *Dear Sirs*).

Enter this text in the field labeled **Expected plaintext.** After each key stroke the **Letter information** table is updated.

We now already have a couple of di-grams fixed:

SE > TG	AM > VN
HR > IG	EN > IX
GE > HR	UN > AX
EH > RI	DH > LR
RT > EF	ER > RG
ED > RX	RE > GR

3.6 By pressing the **Generate matrix** button, those parts of the Playfair key matrix are calculated, that can be determined based on the given information.



Playfair Analysis

Password:

```
*****FS**T*V*AUDMLNXRGHIE
```

Key matrix

[ ] Text was pre-formatted

[✓] Duplicates are ignored

(●) 5x5 Matrix ( ) 6x6 Matrix

*	*	*	*	*	
F	S	*	*	T	
*	V	*	A	U	
D	M	L	N	X	
R	G	H	I	E	

[✓] Update expected plaintext using the key matrix

Letter information

	LROU	Nb?	Horizontal	Vertical	row or col?
R	EG**	E/G	E!,H!,I!,G!	F!,D!	G,I,E,H,F,D
E	IR**	I/R	R!,I!,H!,G!	T!,X!	G,T,R,H,I,X
G	R***	R/	I!,H!,E!,R!	S!,M!	E,S,R,H,M,I
H	****	/	I!,E!,R!,G!	L!	I,G,E,R,L
I	*E**	/E	H!,E!,R!,G!	N!	H,R,E,N,G
N	****	/	M!	I!	M,A,X,I,L
X	****	/	D*,N*,M*	E!,T!	D,E,N,U,M,T
D	****	/	X.,L.	R!,F!	X,R,L,F
A	****	/	V.		V,N,U
T	****	/	F.	E!,X!	S,E,F,X
L	****	/	N.,M*	H!	D,H,N,M
M	****	/	N!,X.,L.	G!,S!	N,V,X,G,L,S
F	****	/	T*	R!,D!	T,R,D
S	****	/	T.	G!,M!	T,G,M
U	****	/	A.		A,X
V	****	/	A*		A,M
B	****	/			
O	****	/			
K	****	/			
P	****	/			
Q	****	/			
W	****	/			
C	****	/			

Analysis result: The following rows contains ciphertext, input or result of the frequency analysis, found plaintext

```
TGIG.HRRIEFR.XVNIX.AXL.RRGGRX...IBO.ARGXNHHK.HRIRGNRG.UTPU.KNRETF.OBH
SEHR.GEEHRTE.DAMEN.UND.HERREN...E**.*IEMLI**.GEHEIMER.T***.**EI*T.***
SEHR.GEEHRTE.DAMEN.UND.HERREN...E**.*IEMLI**.GEHEIMER.T***.**EI*T.***
```

Expected plaintext:

```
.DAMEN.UND.HERREN...E**.*IEMLI**.GEHEIMER.T***.**EI*T.*****.*MMER.**N
```

Generate matix
Output results of analysis
Synchronise analysis result / Expected plaintext
Cancel

3.7 Press the button **Synchronise analysis result \ Expected plaintext**.

The selected checkbox-option **Update expected plaintext** causes that the result of the decryption of the ciphertext based on the partial matrix is copied into the field "Expected plaintext". Characters that cannot (yet) be decrypted are marked with the wildcard character '`*`' and can now be replaced by guessing.

3.8 Now the character sequence **`E**.*IEMLI**`** catches our eye (at least the eye of people knowing some German). If we complete this with the field "Expected plaintext" to read **`EIN.ZIEMLICH`** two more di-gram mappings result: IN > BO and CH > HK. Based on the previous matrix and the first di-gram mapping we conclude, that the characters I,N,B,O have to be in the same column, i.e. B must be directly below I and O directly below N. Clicking on **Generate matrix** swaps row two and four, then row two and three etc. resulting in the following key matrix:

```
* * * K B
U Z * * A
X D M L N
T F S C O
E R G H I
```

3.9 Unfortunately, decrypting based on this matrix results in words that do not exist in the German language:

```
SEHR.GEEHRTE.DAMEN.UND.HERREN...EIN.ZIEMLICH.
GEHEIMER.*X**.BLEIOT.NICH*.IMMER.**NGE.GEHEIM..
MI*.*DXUNDLICHEN.GRUX**EN
```

However, we can now guess the cleartext with high probability, except for the word **`*X**`**, which we correct to read **`****`** (this is to avoid conflicts). We now correct the remaining words to read:

```
SEHR.GEEHRTE.DAMEN.UND.HERREN...EIN.ZIEMLICH.
GEHEIMER.****.BLEIBT.NICHT.IMMER.LANGE.GEHEIM..
MIT.FREUNDLICHEN.GRUESSEN
```

3.10 This results in more di-gram mappings BT > TF, TI > BE, LA > NW, TF > BZ, RE > FT, SS > KK and UE > XT. One more click on **Generate matrix** yields:

```
SEHR.GEEHRTE.DAMEN.UND.HERREN...EIN.ZIEMLICH.
GEHEIMER.TE**.BLEIBT.NICHT.IMMER.LANGE.GEHEIM..
MIT.FREUNDLICHEN.GRUESSEN
```

```
T F S K B
U Z V W A
X D M L N
* * * C O
E R G H I
```

3.11 In a final step we correct **TE**** to read **TEXT** giving the di-gram mapping XT > PU. The Playfair analysis now gives us the key matrix

```
T F S K B
U Z V W A
X D M L N
P * * C O
E R G H I
```

3.12 The cleartext originally had been encrypted with the password **HIERGIBTESWASZUENTDECKEN** (see advice regarding the key matrix), and the options **Preformat text** and **Ignore duplicates within the key phrase** were deactivated. We finally get the complete plaintext:

SEHR.GEEHRTE.DAMEN.UND.HERREN...EIN.ZIEMLICH.
GEHEIMER.TEXT.BLEIBT.NICHT.IMMER.LANGE.GEHEIM..
MIT.FREUNDLICHEN.GRUESSEN

3.13 After clicking the button **Output results of analysis**, the Playfair analysis dialog window is closed and the cleartext is displayed.

Remark:

Advice regarding the key matrix:

A cyclicly shifted key matrix leads to the same result. Thus, the originally chosen password cannot be concluded from the calculated matrix.

EXERCISE

4. Encrypt the text "SH IS HE RX RY LO VE SH EA TH LE DG ER "with a key K= SHERRY what is the result?

5. Encrypt and Decrypt your name as a text and NED as a key using Playfair cipher and show the steps involved?

Lab Session 06

OBJECT: Hash Function – Implementation of MD5

A **hash function** is a (mathematical) function which receives an input of arbitrary length and returns a function value (hash value) of a fixed length (usually 128 or 160 bits). The hash functions which are used in cryptography should be called **one-way hash functions**.

The hash function itself must be public, i.e. everyone should be able to (efficiently) calculate the hash value for a given input. Conversely, however, it must be (computationally) infeasible to find an input for a given value which possesses exactly the predetermined value as hash value (this is referred to as **a one-way characteristic**).

In cryptographic practice, generally hash functions must satisfy a second, more stringent requirement: it must be impossible to find two values which are mapped by the hash function onto the same hash value (collisions), since otherwise it would be possible to replace an already signed message after the event. Hash functions are especially used in connection with digital signatures. Here, for reasons of efficiency, the hash value of a message is signed instead of the message itself.

The hash value H = H(M) of an original message M, is used to check, if the received message M' has been changed.

Example: Some earlier computer magazines added row check sums to the printouts of source code examples. Using these check sums it was possible to recover fast typing errors of the printed source code, because only the row check numbers of the source code have to be checked to discover an error within a row (CDs and downloads didn't exist in these times).

For these checksum methods it is easy to find two different row messages R1, R2 with the same row check sum - meaning H(R1) = H(R2) but R1 != R2: So it is possible to compute a malicious modified source code, without the possibility to detect this modification only via checking the row check sums. We call two different rows with equal check sum a collision. In the notion of hash values we call this a collision if

M != M' and H(M) = H(M')

For cryptographic purposes we must consider the requirement that for the hashing method H and any message M it must be practically impossible to compute a collision. This means to compute a fake massage M' != M, with the characteristic H(M') = H(M).

In the example with the printed source code in earlier computer magazines simple checksum methods have been satisfying, because messages (lines) were not faked by will, but only accidentally mistyped.

CrypTool provides cryptographically strong hash methods via the **Indiv. Procedures \ Hash** menu. These hash methods compute for any document a hash value of constant length, independent whether if the input length is 1 byte or 1 Mbyte. The length of the output (hash value) is always between 16 bytes and 64 bytes (between 128 bit and 512 bit) long - depending on the hash method.

Message Digest (MD5)

The MD5 algorithm produces a hash value of length 128 bits (= 16 bytes).

Step 1. Append Padding Bits

The message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

Step 2. Append Length

A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low-order 64 bits of b are used. (These bits are appended as two 32-bit words and appended low-order word first in accordance with the previous conventions.) At this point the resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit) words. Let M[0... N-1] denote the words of the resulting message, where N is a multiple of 16.

Step 3. Initialize MD Buffer

A four-word buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

```
word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10
```

Step 4. Process Message in 16-Word Blocks

We first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

```
F(X,Y,Z) = XY v not(X) Z
G(X,Y,Z) = XZ v Y not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))
```

In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of v since XY and not(X)Z will never have 1,s in the same bit position.) It is interesting to note that if the bits of X, Y, and Z are independent and unbiased, then each bit of F(X,Y,Z) will be independent and unbiased. The functions G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its inputs.

This step uses a 64-element table T[1... 64] constructed from the sine function. Let T[i] denote the i-th element of the table, which is equal to the integer part of 4294967296 times abs(sin(i)), where i is in radians. The elements of the table are given in the appendix.

Do the following:

```
/* Process each 16-word block. */
For i = 0 to N/16-1 do /* Copy block i into X. */
    For j = 0 to 15 do
        Set X[j] to M[i*16+j].
    end /* of loop on j */
    /* Save A as AA, B as BB, C as CC, and D as DD. */
    AA = A
    BB = B
    CC = C
    DD = D
    /* Round 1. */
    /* Let [abcd k s i] denote the operation a = b + ((a + F(b,c,d) + X[k] +
T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD 0 7 1]  [DABC 1 12 2]  [CDAB 2 17 3]  [BCDA 3 22 4]
    [ABCD 4 7 5]  [DABC 5 12 6]  [CDAB 6 17 7]  [BCDA 7 22 8]
    [ABCD 8 7 9]  [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
    [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]
    /* Round 2. */
    /* Let [abcd k s i] denote the operation a = b + ((a + G(b,c,d) + X[k] +
T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
    [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
    [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
    [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]
    /* Round 3. */
    /* Let [abcd k s t] denote the operation a = b + ((a + H(b,c,d) + X[k] +
T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
    [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
    [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
    [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
    /* Round 4. */
    /* Let [abcd k s t] denote the operation a = b + ((a + I(b,c,d) + X[k] +
T[i]) <<< s). */
    /* Do the following 16 operations. */
    [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
    [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
    [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
    [ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]
    /* Then perform the following additions. (That is increment each of the
four registers by the value it had before this block was started.) */
```

```
    A = A + AA
    B = B + BB
    C = C + CC
    D = D + DD
end /* of loop on i */
```

Step 5. Output

The message digest produced as output is A, B, C, D. That is, we begin with the low-order byte of A, and end with the high-order byte of D. This completes the description of MD5.

EXERCISE

6. Calculate Hash or finger print of your name using MD5, SHA-1 and SHA-256 algorithms and explain the result.

7. The hash compute for any document the output is a constant length, independent whether if the input length is 1 byte or 1 Mbyte. Explain how the hash algorithms achieve this fixed length output.

Lab Session 07

OBJECT: Data Encryption Standard (DES), Implementation of DES, 3-DES

CrypTool offers two variations of the DES cipher, electronic codebook (ECB) and cipher-block chaining (CBC), these are both known as block cipher modes of operation. DES breaks down a plaintext message into 64 bit sized blocks; these are then converted to 64 bit blocks of ciphertext. Conventionally these ciphertext blocks have no relation to one another's encoding; this is the former of the two methods, ECB. There is however a weakness with this method, the DES encodes two identical plaintext blocks as two identical ciphertext blocks, in a message where there could be large repletion of a 64 bit block, say an image, this could reveal patterns in the ciphertext message and ultimately lead to it been deciphered. CBC is a method that overcomes this by combining each block with its preceding block with an exclusive-OR logic gate. A factor that can make this cipher at times in practical is that the whole message must be transmitted error free in order for it to be comprehensibly deciphered at the receiver side. With the ECB method an error in one block will only effect the deciphering of that one block and not the entire message.

Using DES with CrypTool

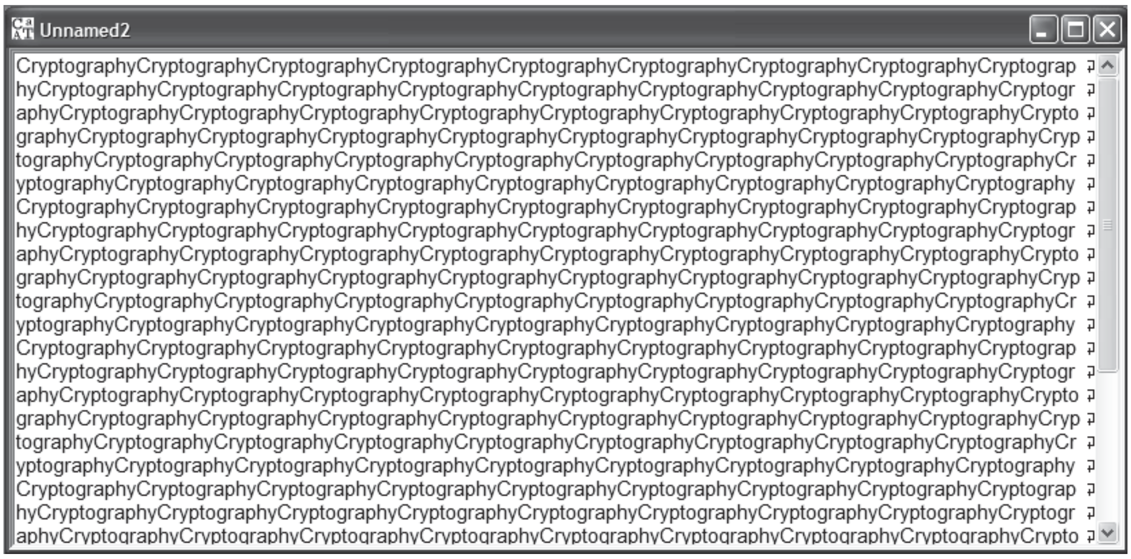
Open a new file and type a plaintext message or alternatively open the file examples/Startingexample-en.txt. Next click from the menu Encrypt/Decrypt > Symmetric (modern) > DES (ECB)… This presents a key selection window; this key must be 64 bits long, which equates to 16 hexadecimal figures. For simplicity use the default key of: 00 00 00 00 00 00 00 00

Select Encrypt and there should be presented a window showing the data encrypted in hexadecimal form and its corresponding ASCII representation. To decrypt the message again select Encrypt/Decrypt > Symmetric (modern) > DES (ECB)… Use the same key and select Decrypt, and the original message will be displayed in hexadecimal representation. Selecting View > Show as text displays it in ASCII; you may also notice some of the formatting is lost in the process or some padding is added.

Encrypt the same message using the same process as above only selecting Encrypt/Decrypt > Symmetric (modern) > DES (CBC)… instead. Comparing the two encrypted messages, you should notice they start with the same 8 bytes, but then they are not the same but both are illegible.

Comparison of ECB and CBC

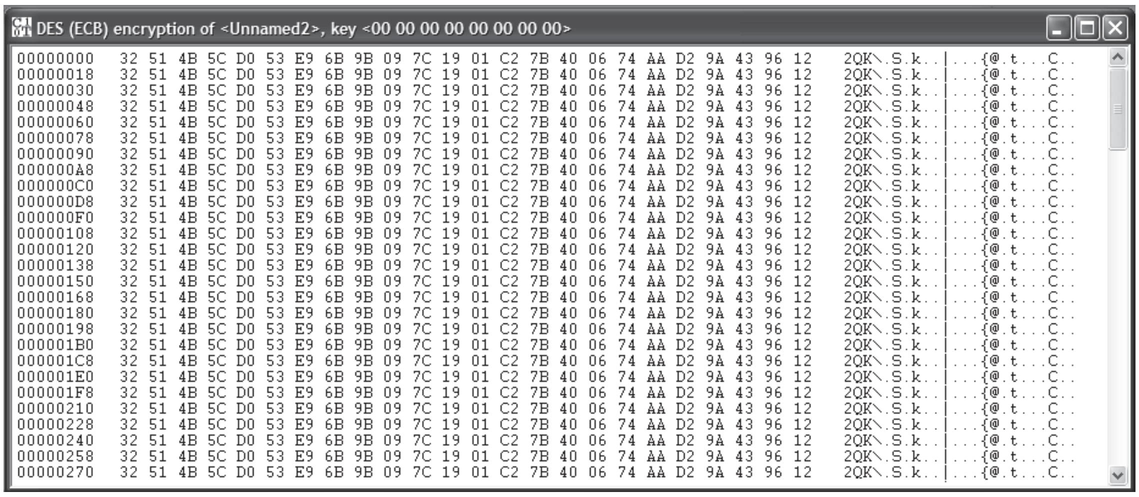
Now create a new plaintext file, choose a word and type it in, now copy and paste this word until there are about 100 repetitions. For the below example the word "Cryptography" *was* used:



Unnamed2

```
CryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptograp
hyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptogr
aphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCrypto
graphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryp
tographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCr
yptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptography
CryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptograp
hyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptogr
aphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCrypto
graphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryp
tographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCr
yptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptography
CryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptograp
hyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptogr
aphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCrypto
graphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryp
tographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCr
yptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptography
CryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptograp
hyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptogr
aphyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCryptographyCrypto
```

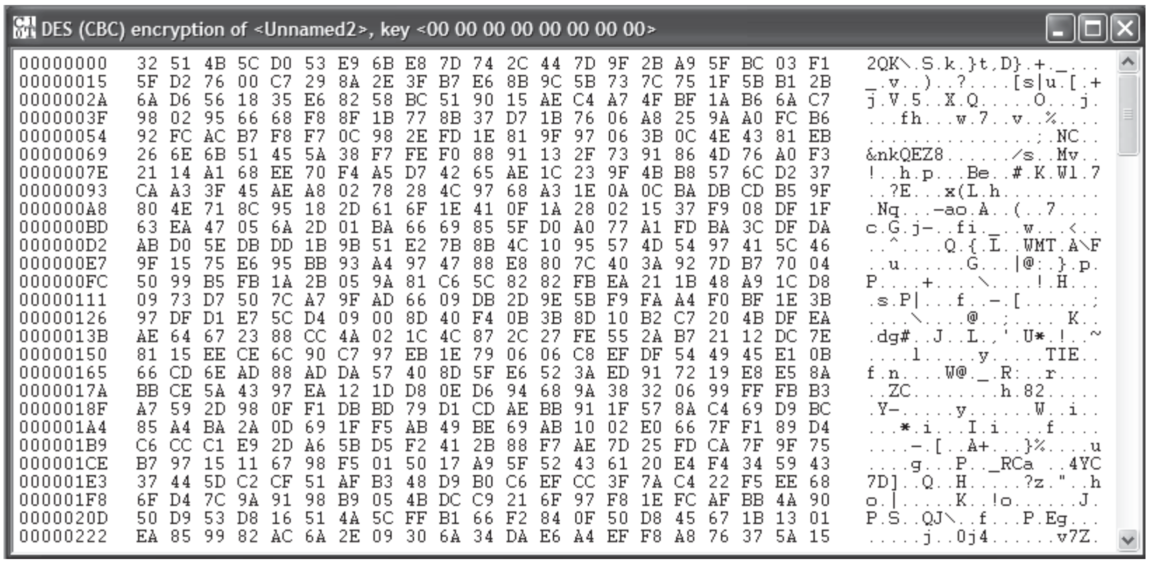
Encrypt this first with the **ECB** variant of the **DES** cipher, there should be a notable pattern present (the word is 12 bytes long, 8 bytes form a block, all 3 blocks you see the same pattern):



DES (ECB) encryption of <Unnamed2>, key <00 00 00 00 00 00 00 00>

```
00000000   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000018   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000030   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000048   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000060   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000078   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000090   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000000A8   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000000C0   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000000D8   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000000F0   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000108   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000120   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000138   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000150   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000168   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000180   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000198   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000001B0   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000001C8   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000001E0   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
000001F8   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000210   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000228   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000240   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000258   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
00000270   32 51 4B 5C D0 53 E9 6B 9B 09 7C 19 01 C2 7B 40 06 74 AA D2 9A 43 96 12   2QK\.S.k..|...{@.t...C..
```

A pattern likes this could potentially help a cryptanalyst decipher the data. Next encrypt the same message using the **CBC** method.


DES (CBC) encryption of <Unnamed2>, key <00 00 00 00 00 00 00 00>

```
00000000   32 51 4B 5C D0 53 E9 6B E8 7D 74 2C 44 7D 9F 2B A9 5F BC 03 F1   2QK\.S.k.}t,D}.+._...
00000015   5F D2 76 00 C7 29 8A 2E 3F B7 E6 8B 9C 5B 73 7C 75 1F 5B B1 2B   _.v..)..?....[s|u.[.+
0000002A   6A D6 56 18 35 E6 82 58 BC 51 90 15 AE C4 A7 4F BF 1A B6 6A C7   j.V.5..X.Q.....O...j.
0000003F   98 02 95 66 68 F8 8F 1B 77 8B 37 D7 1B 76 06 A8 25 9A A0 FC B6   ...fh...w.7..v..%....
00000054   92 FC AC B7 F8 F7 0C 98 2E FD 1E 81 9F 97 06 3B 0C 4E 43 81 EB   ...............;.NC..
00000069   26 6E 6B 51 45 5A 38 F7 FE F0 88 91 13 2F 73 91 86 4D 76 A0 F3   &nkQEZ8....../s..Mv..
0000007E   21 14 A1 68 EE 70 F4 A5 D7 42 65 AE 1C 23 9F 4B B8 57 6C D2 37   !..h.p...Be..#.K.Wl.7
00000093   CA A3 3F 45 AE A8 02 78 28 4C 97 68 A3 1E 0A 0C BA DB CD B5 9F   ..?E...x(L.h.........
000000A8   80 4E 71 8C 95 18 2D 61 6F 1E 41 0F 1A 28 02 15 37 F9 08 DF 1F   .Nq...-ao.A..(..7....
000000BD   63 EA 47 05 6A 2D 01 BA 66 69 85 5F D0 A0 77 A1 FD BA 3C DF DA   c.G.j-..fi._..w...<..
000000D2   AB D0 5E DB DD 1B 9B 51 E2 7B 8B 4C 10 95 57 4D 54 97 41 5C 46   ..^....Q.{.L..WMT.A\F
000000E7   9F 15 75 E6 95 BB 93 A4 97 47 88 E8 80 7C 40 3A 92 7D B7 70 04   ..u......G...|@:.}.p.
000000FC   50 99 B5 FB 1A 2B 05 9A 81 C6 5C 82 82 FB EA 21 1B 48 A9 1C D8   P....+....\....!.H...
00000111   09 73 D7 50 7C A7 9F AD 66 09 DB 2D 9E 5B F9 FA A4 F0 BF 1E 3B   .s.P|...f..-.[......;
00000126   97 DF D1 E7 5C D4 09 00 8D 40 F4 0B 3B 8D 10 B2 C7 20 4B DF EA   ....\....@..;.... K..
0000013B   AE 64 67 23 88 CC 4A 02 1C 4C 87 2C 27 FE 55 2A B7 21 12 DC 7E   .dg#..J..L.,'.U*.!..~
00000150   81 15 EE CE 6C 90 C7 97 EB 1E 79 06 06 C8 EF DF 54 49 45 E1 0B   ....l.....y.....TIE..
00000165   66 CD 6E AD 88 AD DA 57 40 8D 5F E6 52 3A ED 91 72 19 E8 E5 8A   f.n....W@._.R:..r....
0000017A   BB CE 5A 43 97 EA 12 1D D8 0E D6 94 68 9A 38 32 06 99 FF FB B3   ..ZC........h.82.....
0000018F   A7 59 2D 98 0F F1 DB BD 79 D1 CD AE BB 91 1F 57 8A C4 69 D9 BC   .Y-.....y......W..i..
000001A4   85 A4 BA 2A 0D 69 1F F5 AB 49 BE 69 AB 10 02 E0 66 7F F1 89 D4   ...*.i...I.i....f....
000001B9   C6 CC C1 E9 2D A6 5B D5 F2 41 2B 88 F7 AE 7D 25 FD CA 7F 9F 75   ....-.[..A+...}%....u
000001CE   B7 97 15 11 67 98 F5 01 50 17 A9 5F 52 43 61 20 E4 F4 34 59 43   ....g...P.._RCa ..4YC
000001E3   37 44 5D C2 CF 51 AF B3 48 D9 B0 C6 EF CC 3F 7A C4 22 F5 EE 68   7D]..Q..H.....?z."..h
000001F8   6F D4 7C 9A 91 98 B9 05 4B DC C9 21 6F 97 F8 1E FC AF BB 4A 90   o.|.....K..!o......J.
0000020D   50 D9 53 D8 16 51 4A 5C FF B1 66 F2 84 0F 50 D8 45 67 1B 13 01   P.S..QJ\..f...P.Eg...
00000222   EA 85 99 82 AC 6A 2E 09 30 6A 34 DA E6 A4 EF F8 A8 76 37 5A 15   .....j..0j4......v7Z.
```

As shown with the above example there should be no obvious pattern present making this a much stronger form of DES encryption.

EXERCISE

8. What are ECB and CBC and their purpose? How do they differ?

____

____

____

9. For each of the following say whether ECB or CBC would be most appropriate and give a brief explanation as to why.
   - An online bank statement
   - An encrypted VoIP session
   - Viewing of a website using TCP/IP

____

____

____

10. Why are the following keys considered to be weak keys of DES. Think about applying these keys to cryptool preferably trying to encrypt text with these keys twice.
    - K1= 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
    - K2= F E F E F E F E F E F E F E F E
    - K3= 1 F 1 F 1 F 1 F 0 E 0 E 0 E 0 E
    - K4= E 0 E 0 E 0 E 0 F 1 F 1 F 1 F 1

____

____

____

Lab Session 08

OBJECT: Implementation of Diffie-Hellman Key Exchange Method

The symmetric encryption methods require that the keys are securely transported between sender and receiver. Each pair of sender and receiver shares a common secret key. For technical reasons and to evade the huge efforts for mass direct key exchange of the different keys, other methods have been invented, which depend on a public key agreement. The session key is generated interactively and unknown by others for each session.

Besides the asymmetric encryption methods multiple methods for key exchange exist, which make use of public and private keys too. The most famous and even today implemented method was developed by Diffie and Hellman.

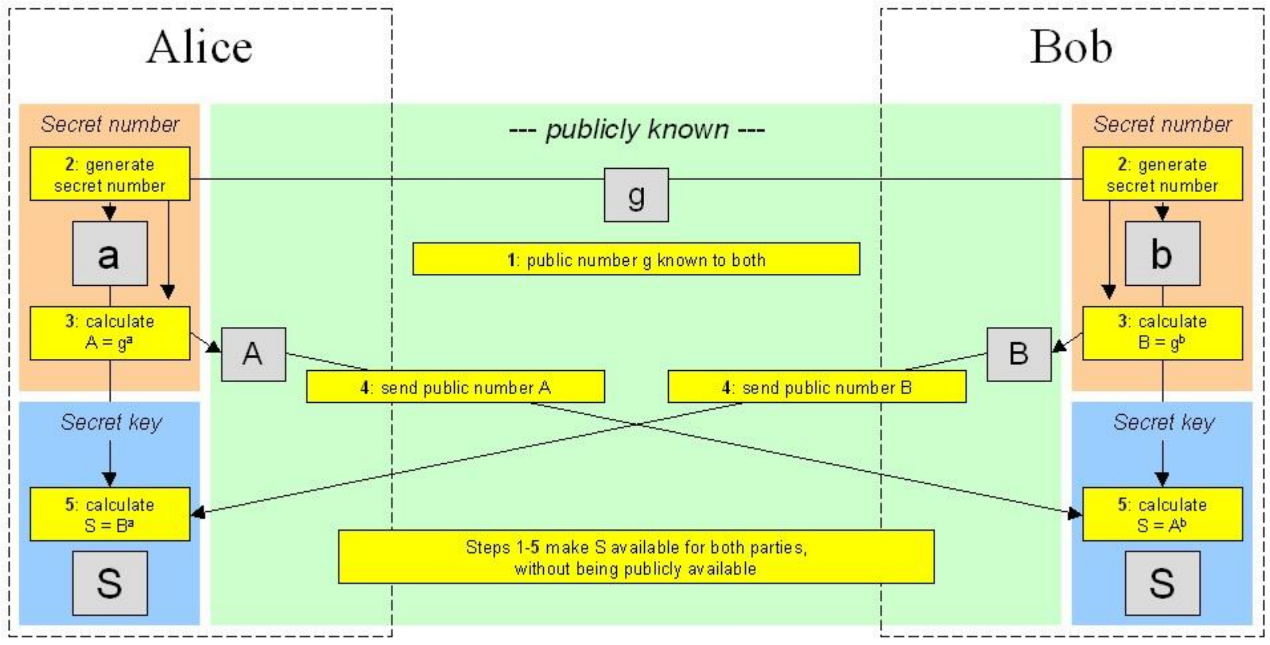
Diffie-Hellman Key Exchange Protocol

The Diffie-Hellman key exchange protocol was developed by Whitfield Diffie, Martin E. Hellman and Ralph Merkle in 1976.

CrypTool contains a visualization of the Diffie-Hellman key exchange protocol. You can access it via the menu item Individual Procedures / Diffie-Hellman Demonstration.

Despite communicating via a non-secure line, the protocol enables two or more parties to negotiate on a secret key which is solely known to the participating parties.

For better comprehension the two participating parties will be described as Alice and Bob. The following drawing is to give a quick overview of the Diffie-Hellman key exchange protocol:



The secret and common key, often described as *session key*, depends on the following parameters:

- Public parameters (p and g)
- Secret of Alice
- Secret of Bob
- Shared key of Alice
- Shared key of Bob

The protocol is based on the discrete logarithm problem, that is making use of a so-called **one-way-function** property: Whereas the computation of a mathematical one-way-function is of low complexity, the inverse operation is extremely hard to accomplish.

The creation of the session key at the end of the protocol is based on such a one-way-function: Even if a potential attacker has knowledge of the shared keys of both Alice and Bob, it is hardly possible for him to compute the session key; depending on the key length, it can easily turn out as impossible for the attacker to calculate the session key.

Increasing the prime module p directly affects the complexity to compute the session key. Anyhow, the additional expenses to compute the exponential function are bearable for Alice and Bob.

Contrary to the situation of Alice and Bob, the attacker needs to compute the discrete logarithm, which is the inversion of the exponential function. This effort increases much more than the effort necessary for the two communication partners.

For this reason the safety of the Diffie-Hellman key exchange protocol depends essentially on the size of the prime module p. In practical prime module numbers are considered safe if their bit length has at least 1024 bit, which complies with a 300-digit decimal number.

Besides the prime module p, the secret numbers of Alice and Bob need to be of a certain length in order to prevent the attacker from correctly guessing the session key by chance. Thus, in practice secrets should be at least 80 bit long (25-digit decimal number).

Remark 1:

The key exchange protocol according to Diffie-Hellman is **NOT an encryption method**; it is merely used to agree on a secret and common key.

Remark 2:

The Diffie-Hellman protocol is vulnerable against "Man-In-The-Middle" attacks:

The attacker attempts to personate as Alice for Bob and at the same time to personate as Bob against Alice (the attacker stands in the middle of the communication between Alice and Bob). If he is successful then he obtains via the Diffie-Hellman protocol a common secret key with Alice and a second common secret key with Bob.

EXERCISE

11. Calculate the Symmetric Key using Diffie-Hellman Key Exchange method while considering the following values g=4, p=13 and a=10, b=8. Verify the Cryptool generated key with the manual calculations.

12. How Man in the Middle Attach is possible in Diffie-Hellman Key Exchange method.

Lab Session 09

OBJECT: RSA (Rivest–Shamir–Adleman) Implementation

The most well-known asymmetric cryptosystem is the RSA algorithm, named after its authors, Ron Rivest, Adi Shamir and Leonard Adleman. A special characteristic of the RSA cryptosystem is that extensive calculations are necessary to generate the RSA key before RSA encryption or decryption can take place. First of all the RSA parameters p, q, N and the Euler number phi(N) are calculated:

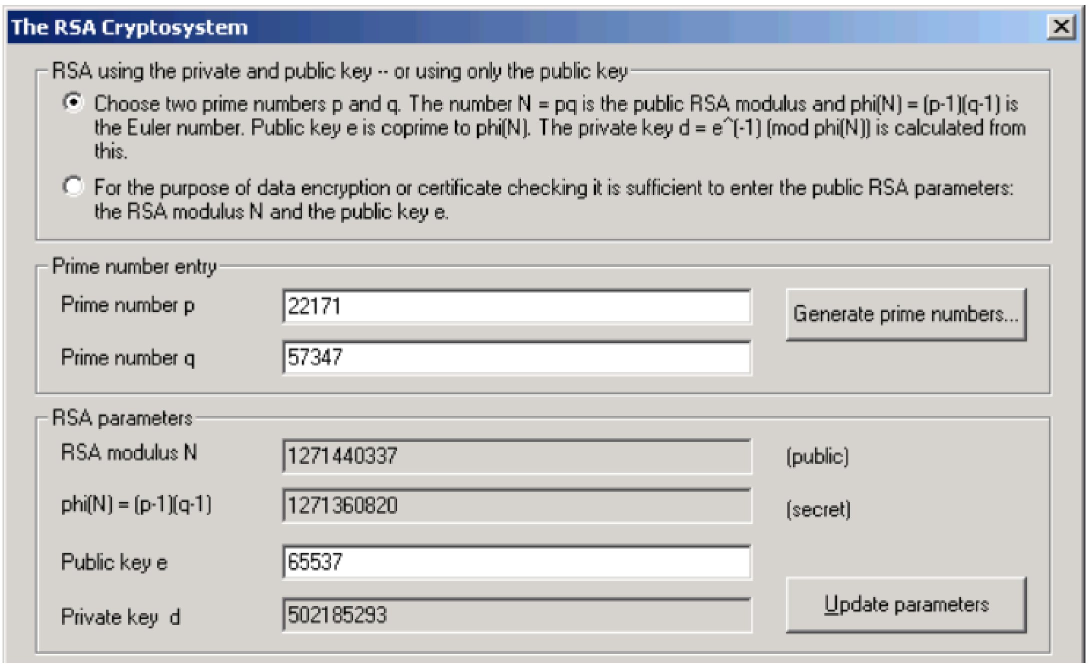
Choose two different prime numbers p and q at random and calculate the so-called RSA modulus N = pq. The Euler number phi(N) = (p-1)*(q-1) is calculated from the prime factors p and q.

In a second step, the public RSA exponent e is determined and from this together with phi(N) the secret RSA exponent d is calculated:

Choose the number e: 1 < e < phi(N), with the property that e is relatively prime to phi(N)

An especially popular value for e is 2^16+1 = 65537, as in most cases this is co-prime to phi(N) and is especially well-suited for square and multiply exponentiation resulting in a very fast public key operation.

The secret exponent d = e^(-1) mod phi(N) is calculated from the public exponent e as the multiplicative inverse modulo phi(N).


The RSA Cryptosystem

RSA using the private and public key -- or using only the public key

[✓ Choose two prime numbers p and q. The number N = pq is the public RSA modulus and phi(N) = (p-1)(q-1) is the Euler number. Public key e is coprime to phi(N). The private key d = e^(-1) (mod phi(N)) is calculated from this.]

For the purpose of data encryption or certificate checking it is sufficient to enter the public RSA parameters: the RSA modulus N and the public key e.

Prime number entry

Prime number p: 22171

Prime number q: 57347

RSA parameters

RSA modulus N: 1271440337 (public)

phi(N) = (p-1)(q-1): 1271360820 (secret)

Public key e: 65537

Private key d: 502185293

In the dialog "The RSA Cryptosystem" you can enter the prime numbers p and q directly. If you do not enter any prime numbers for p or q, an error message will be displayed. By clicking on the Generate prime numbers button, you can search for random prime numbers from within specified number intervals. The Update parameters button is used to calculate the secret exponent d from e and phi(N) as described above. If e is not co-prime to phi(N), an error message will be displayed, asking you to choose a new number for the public exponent e.

After successful generation of the RSA key, the asymmetric RSA key pair is displayed:

(N,e) is the public key and

(N,d) is the secret key.

An RSA key bit length is calculated from the number of bits of the binary representation of the RSA modulus N.

After the key has been generated, anyone can encrypt a message with the RSA algorithm using the public key (N,e), but only the owner of the secret key (N,d) can decrypt the message again with the RSA algorithm.

RSA encryption of messages

After the RSA key generation, the modulus N and the RSA key e can be published. Anyone could encrypt a message to the owner of the secret RSA key or check a digital signature.

If you generate the RSA key by your own in the RSA demo dialog the public RSA parameter appears already in the dialog. In the other case you enter the public RSA parameter. The message can now be encrypted.

To this end, enter the "plaintext" message M that is to be encrypted. For example:

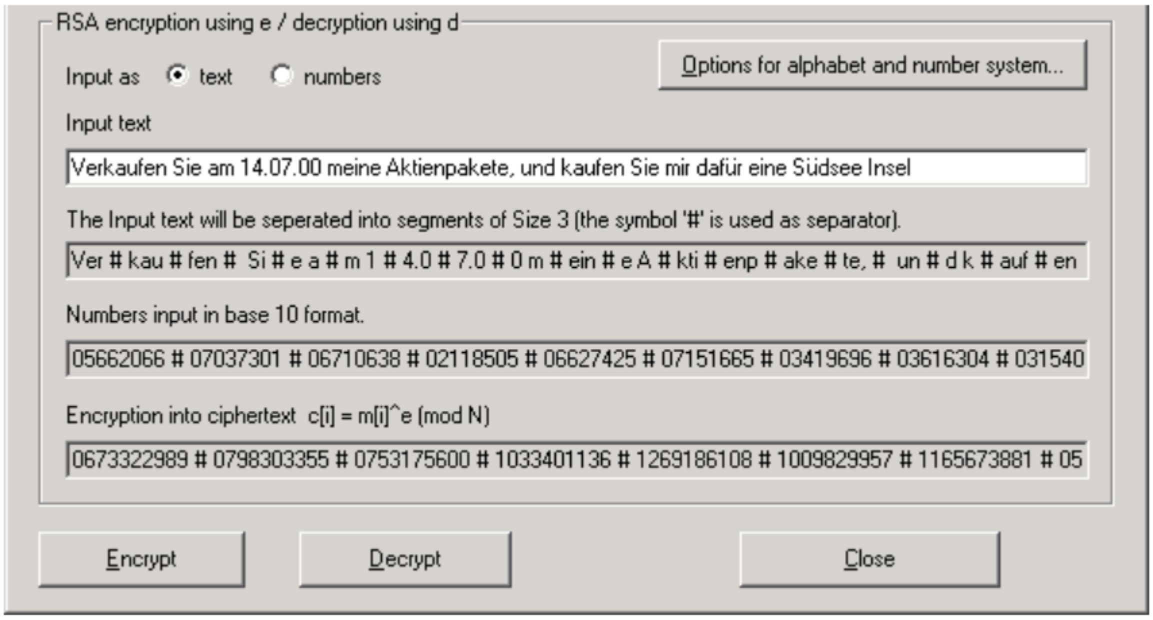
"Verkaufen Sie am 14.07.00 meine Aktienpakete, und kaufen Sie mir dafür eine Südsee Insel" [Sell my shares on 14 July 2000 and use the proceeds to buy me a South Sea Island].

When you click on **Encrypt**, first of all the message is converted to numbers. For this purpose, message M is split up into blocks of k characters (M = M[1] # M[2] # ... # M[j]), whereby k*8 is always smaller than the bit length of RSA modulus N. The message blocks (M[i], i = 1, 2, ..., j) are converted for encryption into numbers (m[i], for i = 1, 2, ..., j) and encrypted with the public RSA key (N,e) according to the following formula:

c[i] = m[i]^e (mod N) for i = 1, 2, ..., j .

Because the public key (N,e) is available to everyone and the secret key (N,d) remains confidential, we can establish that

Anyone can encrypt a message with the public key. But only the owner of the secret key can decrypt the message again.


RSA encryption using e / decryption using d
Input as
text
numbers
Options for alphabet and number system...
Encrypt
Decrypt
Close

Input text

```
Verkaufen Sie am 14.07.00 meine Aktienpakete, und kaufen Sie mir dafür eine Südsee Insel
```

The Input text will be seperated into segments of Size 3 (the symbol '#' is used as separator).

```
Ver # kau # fen #  Si # e a # m 1 # 4.0 # 7.0 # 0 m # ein # e A # kti # enp # ake # te, #  un # d k # auf # en
```

Numbers input in base 10 format.

```
05662066 # 07037301 # 06710638 # 02118505 # 06627425 # 07151665 # 03419696 # 03616304 # 031540
```

Encryption into ciphertext c[i] = m[i]^e (mod N)

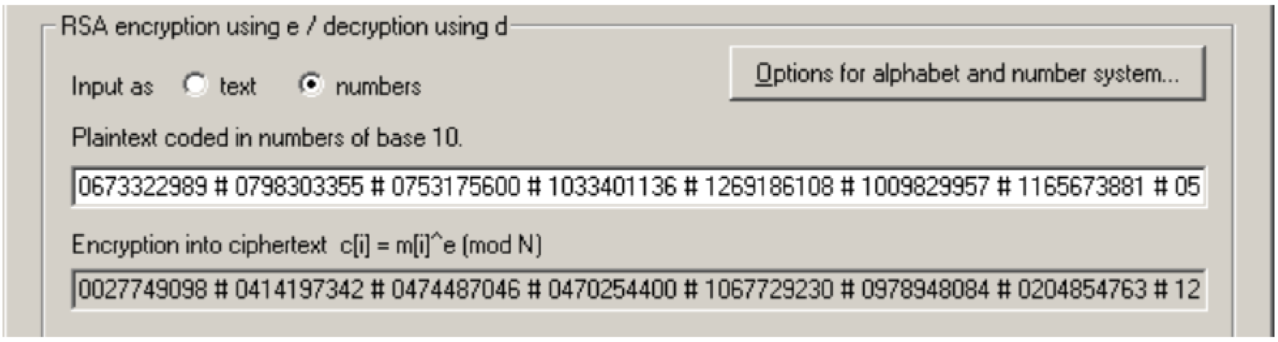
```
0673322989 # 0798303355 # 0753175600 # 1033401136 # 1269186108 # 1009829957 # 1165673881 # 05
```

The conversion of the plaintext M to numbers m = m[1] # m[2] # .... # m[j] and the encrypted version c = c[1] # c[2] # .... # c[j] are displayed in the two lines below the data input field.

The character # does not have any significance here: it is simply a visual separator between different number blocks to be encrypted or decrypted.

You can specify the block length k by clicking on Options for the alphabet and the number system. In the dialog accessed by this option you can also determine the basis for representation of numbers in RSA encryption and you can experiment with the RSA variant from the story Dialogue of the Sisters [see c,t 25/99].

Instead of entering text you can also specify in the options that you wish to directly encrypt a sequence of plaintext numbers m = m[1] # m[2] # ... # m[j]:


RSA encryption using e / decryption using d
Input as
text
numbers
Options for alphabet and number system...

Plaintext coded in numbers of base 10.

```
0673322989 # 0798303355 # 0753175600 # 1033401136 # 1269186108 # 1009829957 # 1165673881 # 05
```

Encryption into ciphertext c[i] = m[i]^e (mod N)

```
0027749098 # 0414197342 # 0474487046 # 0470254400 # 1067729230 # 0978948084 # 0204854763 # 12
```

RSA decryption of a message

Only if you are the owner of the secret RSA key d you are able to decrypt messages which are encrypted with the corresponding public key. If you just have access to the public RSA parameter N and e you can try to break the RSA key (compute the private key, works only for short keys) via the factorization attack.

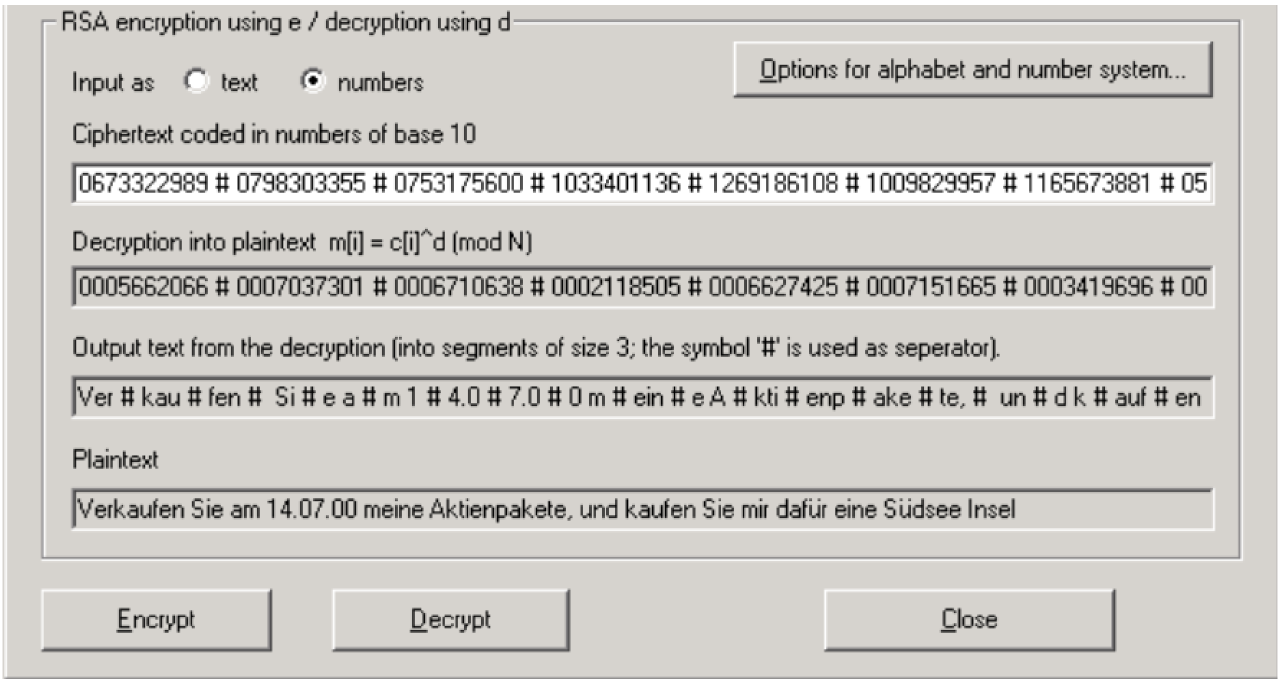
To decrypt RSA-encrypted messages, enter the numeric code of the encrypted message. When you click on **Decrypt**, the entered ciphertext

c = c[1] # c[2] # .... # c[j]

is decrypted using the secret key (N,d) according to the following formula:

m[i] = c[i]^d mod N for i = 1, ..., j .

The numbers m = m[1] # m[2] # .... # m[j] are then transformed to the original text message M undoing the block splitting:


RSA encryption using e / decryption using d
Input as text [✓ numbers]
Options for alphabet and number system...
Encrypt
Decrypt
Close

Ciphertext coded in numbers of base 10

0673322989 # 0798303355 # 0753175600 # 1033401136 # 1269186108 # 1009829957 # 1165673881 # 05

Decryption into plaintext m[i] = c[i]^d (mod N)

0005662066 # 0007037301 # 0006710638 # 0002118505 # 0006627425 # 0007151665 # 0003419696 # 00

Output text from the decryption (into segments of size 3; the symbol '#' is used as seperator).

Ver # kau # fen # Si # e a # m 1 # 4.0 # 7.0 # 0 m # ein # e A # kti # enp # ake # te, # un # d k # auf # en

Plaintext

Verkaufen Sie am 14.07.00 meine Aktienpakete, und kaufen Sie mir dafür eine Südsee Insel

EXERCISE

Perform RSA encryption and decryption. The parameters used here are small. Verify your results with cryptool?

- Choose two distinct prime numbers, such as P=61 q=53
- Compute $n = pq$ giving n=?
- Compute the totient of the product as $\varphi(n) = (p - 1)(q - 1)$
- Choose any number $1 < e < 3120$ that is co-prime to 3120. Choosing a prime number for e leaves us only to check that e is not a divisor of 3120.
- Compute d, the modular multiplicative inverse of e (mod $\varphi(n)$) yielding d=?

Lab Session 10

OBJECT: Attack on RSA encryption with short RSA modulus

The analysis is performed in two stages: first of all the prime factorization of the RSA modulus is calculated using factorization, and then in the second stage the secret key for encryption of the message is determined. After this, the cipher text can be decrypted with the cracked secret key.

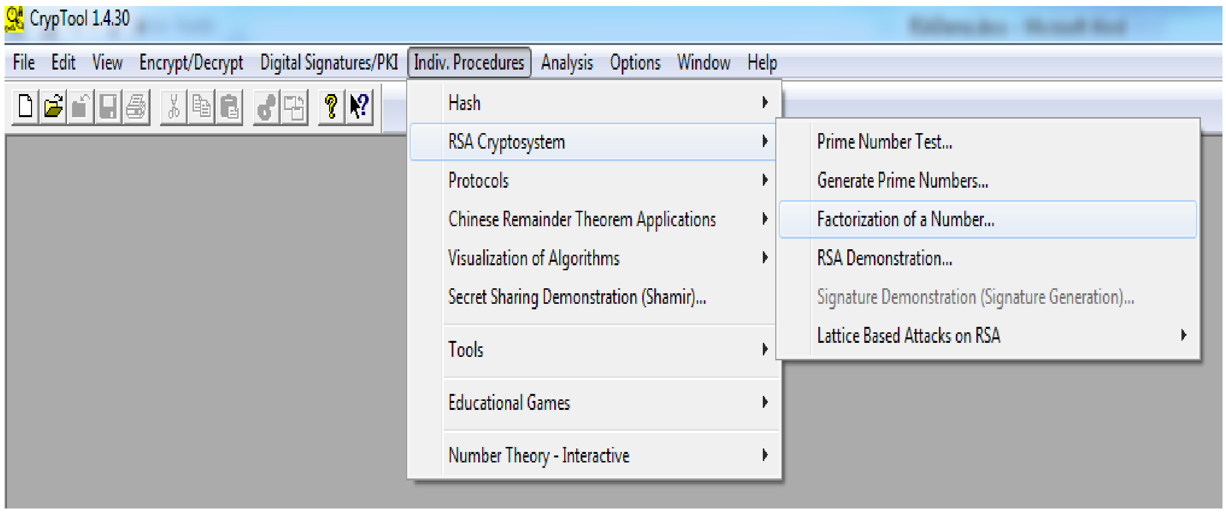
We will figure out plaintext given

RSA modulus n = 63978486879527143858831415041
Public exponent e = 17579

Cipher text = 45411667895024938209259253423, 16597091621432020076311552201, 46468979279750354732637631044, 32870167545903741339819671379

1. Factorization of the RSA modulus with the aid of prime factorization.

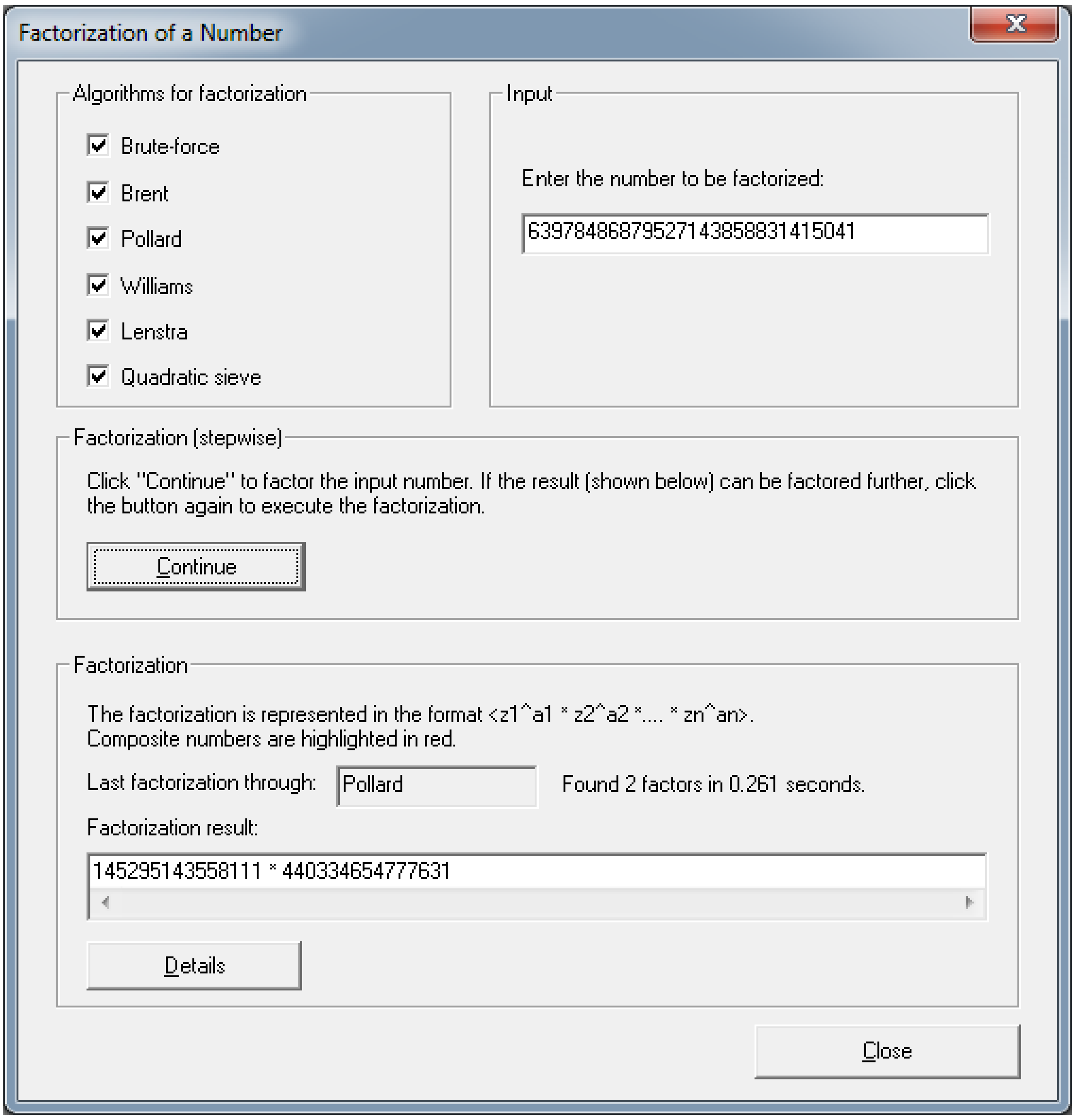
To break down the natural number, select menu **sequence Indiv. Procedure/RSA Cryptosystem / Factorization of a Number**.


CrypTool 1.4.30
File
Edit
View
Encrypt/Decrypt
Digital Signatures/PKI
Indiv. Procedures
Analysis
Options
Window
Help
Hash
RSA Cryptosystem
Protocols
Chinese Remainder Theorem Applications
Visualization of Algorithms
Secret Sharing Demonstration (Shamir)...
Tools
Educational Games
Number Theory - Interactive
Prime Number Test...
Generate Prime Numbers...
Factorization of a Number...
RSA Demonstration...
Signature Demonstration (Signature Generation)...
Lattice Based Attacks on RSA

2. The two components of the public key is

RSA modulus n = 63978486879527143858831415041
Public exponent e = 17579

Enter n=**63978486879527143858831415041** as input and click **Continue**.


Factorization of a Number

Algorithms for factorization
- [✓] Brute-force
- [✓] Brent
- [✓] Pollard
- [✓] Williams
- [✓] Lenstra
- [✓] Quadratic sieve

Input

Enter the number to be factorized: 63978486879527143858831415041

Factorization (stepwise)

Click "Continue" to factor the input number. If the result (shown below) can be factored further, click the button again to execute the factorization.

Factorization

The factorization is represented in the format <z1^a1 * z2^a2 *.... * zn^an>.
Composite numbers are highlighted in red.

Last factorization through: Pollard   Found 2 factors in 0.261 seconds.

Factorization result:

145295143558111 * 440334654777631

It is interesting to see which procedure broke down the RSA modulus the fastest.
2. Calculate the secret key **d** from the prime factorization of n and the public key **e**:

With the knowledge of the prime factors p = 145295143558111 and q = 440334654777631 and the public key e = 17579, we are in a position to decrypt the ciphertext.

3. Open the next dialog box via menu selection **Indiv. Procedure/RSA Cryptosystem/RSA Demonstration**:.

4. Enter **p = 145295143558111 and q = 440334654777631** and the public key **e = 17579**.

5. Click on **Alphabet and number system options** and make the following settings:

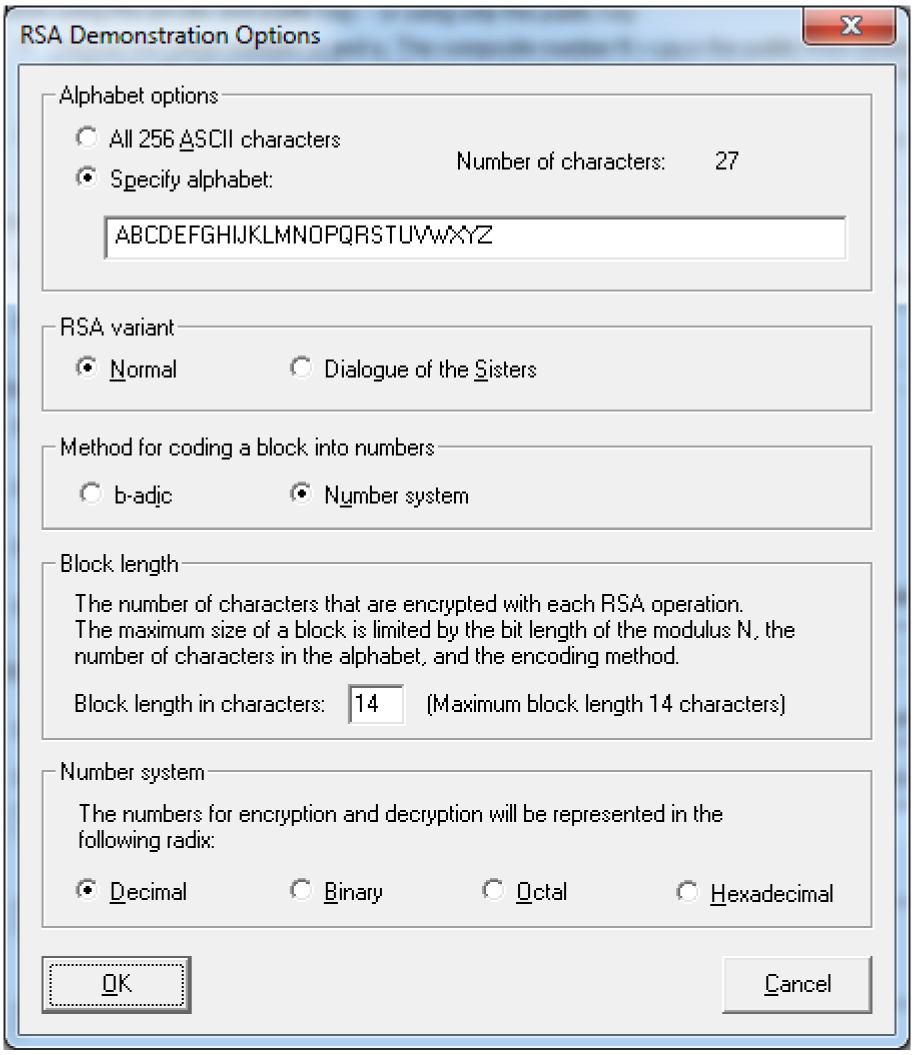
Alphabet options: **Specify alphabet**

RSA variant: **Normal**

Method for coding a block into number: **Number system**

Block length: **14**

Number system: **Decimal**


RSA Demonstration Options

Alphabet options

( ) All 256 ASCII characters

(●) Specify alphabet:

Number of characters: 27

ABCDEFGHIJKLMNOPQRSTUVWXYZ

RSA variant

(●) Normal ( ) Dialogue of the Sisters

Method for coding a block into numbers

( ) b-adic (●) Number system

Block length

The number of characters that are encrypted with each RSA operation.
The maximum size of a block is limited by the bit length of the modulus N, the number of characters in the alphabet, and the encoding method.

Block length in characters: 14 (Maximum block length 14 characters)

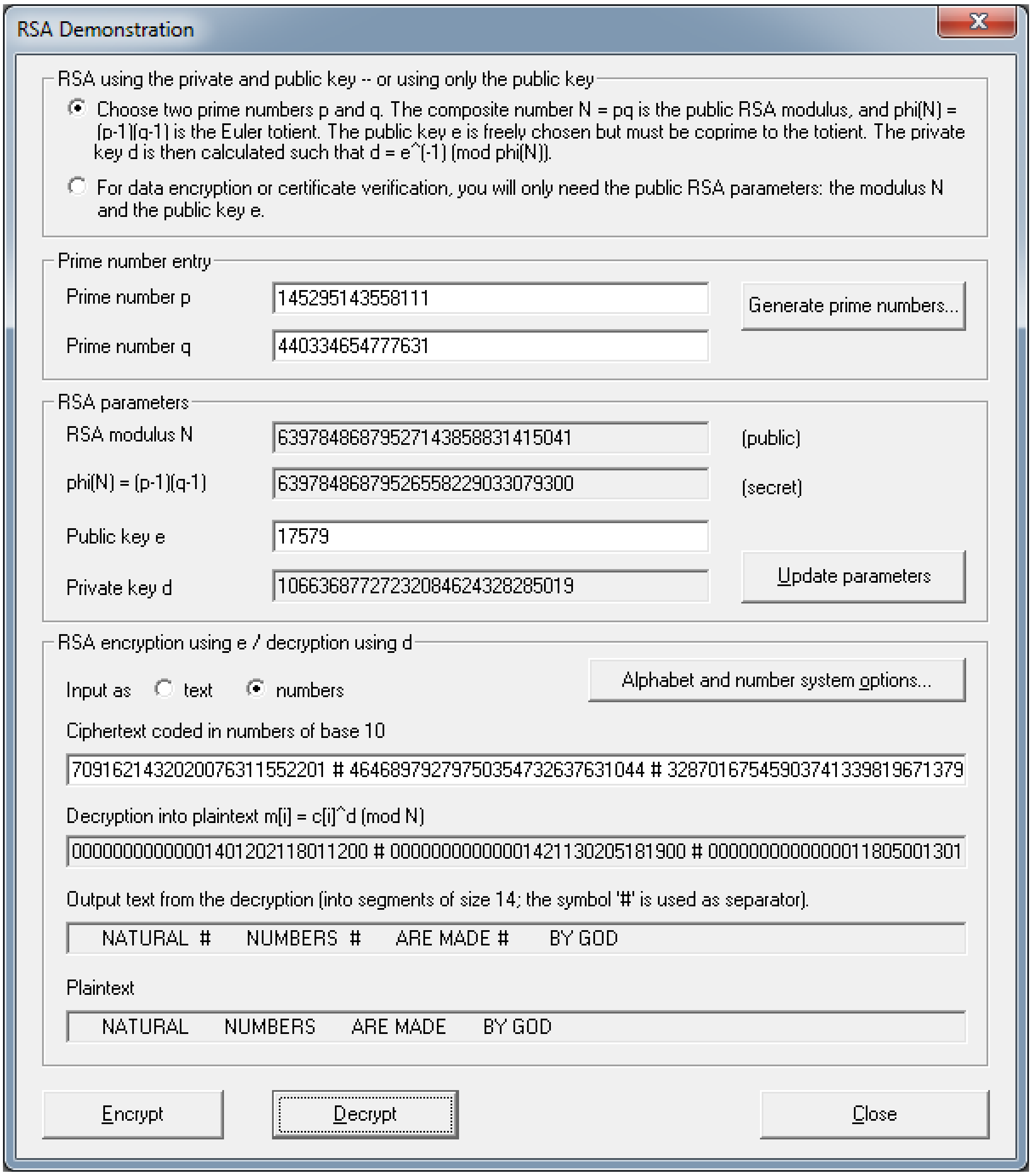
Number system

The numbers for encryption and decryption will be represented in the following radix:

(●) Decimal ( ) Binary ( ) Octal ( ) Hexadecimal

6. Enter the following cipher text in the input text field. And click **Decrypt** button.

45411667895024938209259253423,
16597091621432020076311552201,
46468979279750354732637631044,
32870167545903741339819671379



RSA Demonstration

RSA using the private and public key -- or using only the public key

[✓] Choose two prime numbers p and q. The composite number N = pq is the public RSA modulus, and phi(N) = (p-1)(q-1) is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that d = e^(-1) (mod phi(N)).

[ ] For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p: 145295143558111

Prime number q: 440334654777631

RSA parameters

RSA modulus N: 63978486879527143858831415041 (public)

phi(N) = (p-1)(q-1): 63978486879526558229033079300 (secret)

Public key e: 17579

Private key d: 10663687727232084624328285019

RSA encryption using e / decryption using d

Input as [ ] text [✓ numbers]

Ciphertext coded in numbers of base 10

```
70916214320200763115522201 # 4646897927975035473263763 1044 # 32870167545903741339819671379
```

Decryption into plaintext m[i] = c[i]^d (mod N)

```
00000000000001401202118011200 # 00000000000001421130205181900 # 0000000000000011805001301
```

Output text from the decryption (into segments of size 14; the symbol '#' is used as separator).

```
NATURAL #     NUMBERS #     ARE MADE #     BY GOD
```

Plaintext

```
NATURAL     NUMBERS     ARE MADE     BY GOD
```

Generate prime numbers...
Update parameters
Alphabet and number system options...
Encrypt
Decrypt
Close

Check your results: "**NATURAL NUMBERS ARE MADE BY GOD**"

EXERCISE

In RSA, practical difficulty of factoring the product of two large prime numbers is known as the factoring problem. This is what RSA is based on. The prime factors must be kept secret. If the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.

If we know N = 63978486879527143858831415041 (95 bit, 29 decimal digits) and then try this number N = 351573870816322547022741576341143304183 (129 bit, 39 digit). Find the factors using cryptool by going in to Indiv. Procedures -> RSA Cryptosystem -> Factorization of a number. Then enter the number and find the factors. What does this tell you about the difficulty level of finding the factors in both cases? What are the factors in both cases? What algorithm was used last to factorize in both cases?

Lab Session 11

OBJECT: Network Packet Capturing & Analysis through Ethereal.

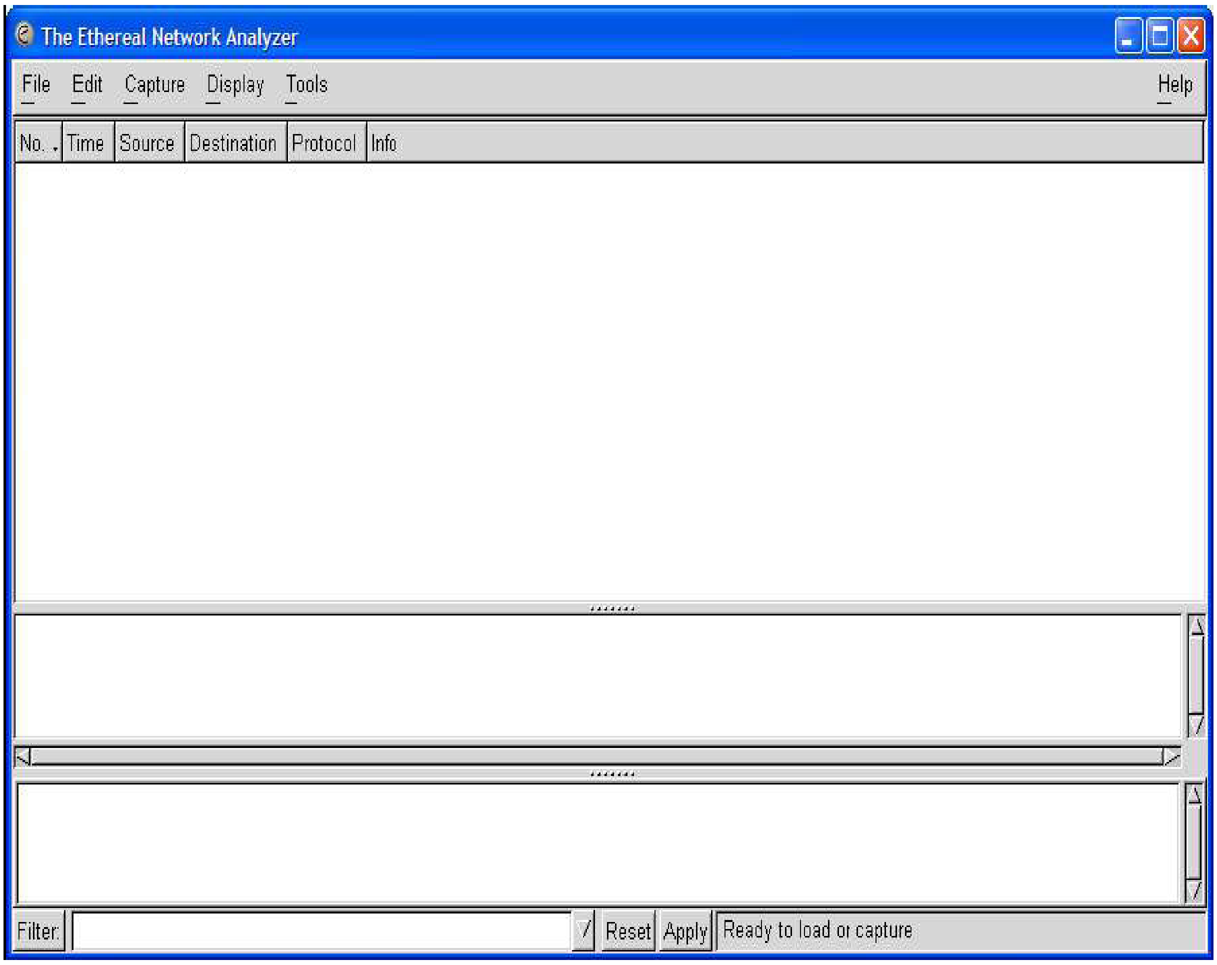
THEORY

Introduction to Ethereal

Ethereal is a very powerful well featured packet analyzer. It captures packets and decodes them into their component parts for analysis. The range of decodes is very large, if you've heard of a protocol there's a good chance that Ethereal has a decoder for it, and if you haven't there's still likely to be a decoder for it. Ethereal is available for use on UNIX systems and for Microsoft Windows. Ethereal uses the same capture and filtering mechanism as tcpdump and can read files captured by tcpdump.

PROCEDURE

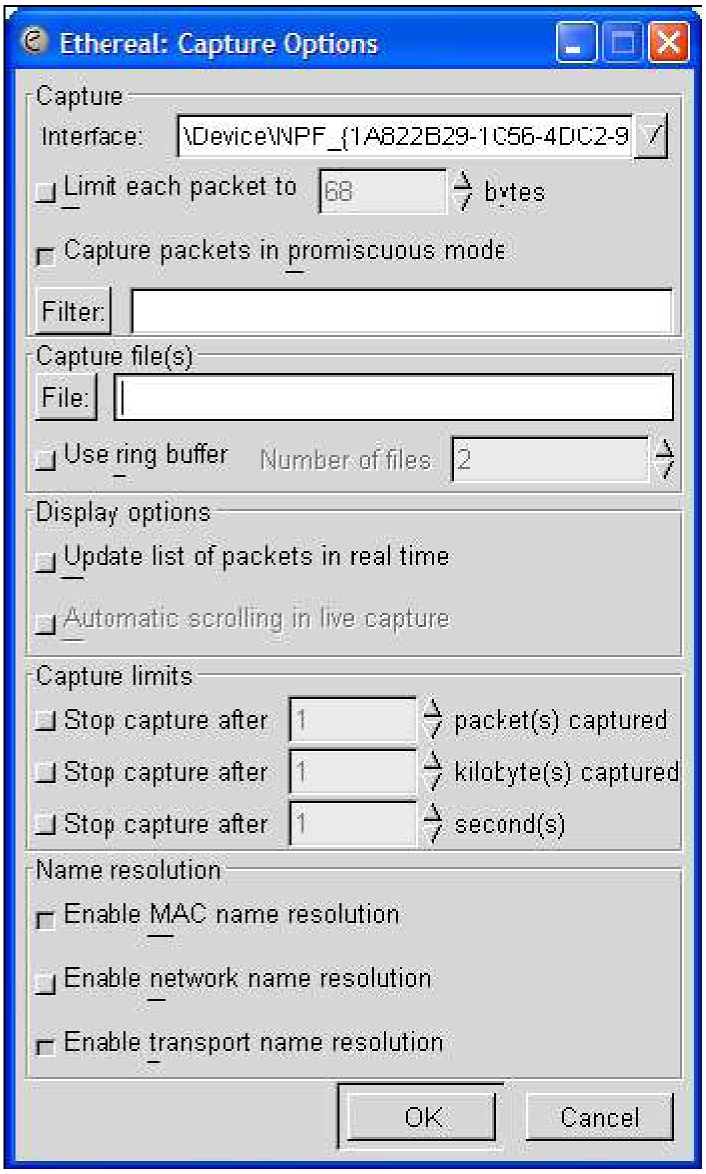
Using Ethereal
Step 1 In Windows click Start, All Programs, OPENXTRA BASICS.
Step 2 Select Ethereal.



Ethereal displays a large windows consisting of three panes. To begin with the panes are blank.

Capturing packets

Click on Capture, Start. The Capture Options box appears.


Ethereal: Capture Options

Capture
Interface: \Device\NPF_{1A822B29-1C56-4DC2-9
Limit each packet to 68 bytes
Capture packets in promiscuous mode
Filter:

Capture file(s)
File:
Use ring buffer Number of files 2

Display options
Update list of packets in real time
Automatic scrolling in live capture

Capture limits
Stop capture after 1 packet(s) captured
Stop capture after 1 kilobyte(s) captured
Stop capture after 1 second(s)

Name resolution
Enable MAC name resolution
Enable network name resolution
Enable transport name resolution

If the square button is raised the option is Off, if the square button is pressed the option is On. Options are grouped into different functions.

Capture

Limit each packet to allows you to restrict allows you to specify, in bytes,how much of each packet to collect. This is useful if you are interested in the header information only, and if you want to keep the file sizes small.

Capture packets in promiscuous mode. If you want to capture everything that your machine can see click this option. If you only want to see packets in and out of your machine leave this option unselected.

Filter allows you to enter an existing capture filter.

Capture file(s)

File allows you to save the captured packets in a named file.Use ring buffer allows you to specify a number of files to use for the capture. In a Ring Buffer when one file is full a new one starts. When the specified number of files are all full capture begins to overwrite the files in sequence. This function is useful if you want to capture continuously but do not want to fill your hard disk.

Note that when Use Ring Buffer is pressed the Capture limits option, Stop capture after xx kilobyte(s) changes to Rotate capture every xx kilobytes.

Display options

Update list of packets in real time. Use this option if you want to see the list of packets as they are captured. Automatic scrolling in live capture. Select this if you want the packet list to scroll.

Capture limits

These options limit the number of packets you can capture. There are three options, limit by number of packets, by an amount of disk space, or by time. All can be enabled at once, the first option to be matched will cause the capture to stop.

If Use ring buffer is pressed Rotate capture every allows you to specify the file size in kilobytes.

To capture continuously switch all the options Off.

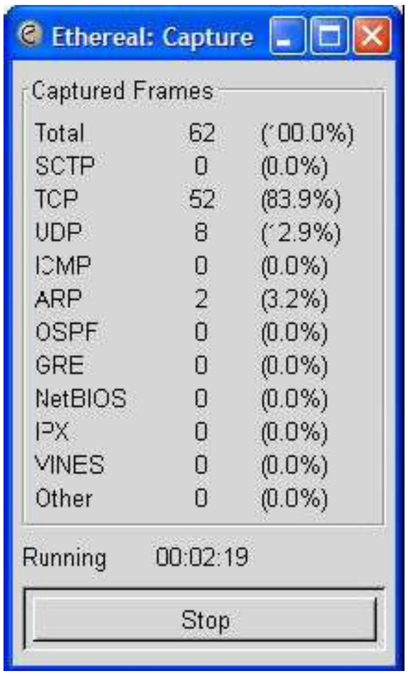
Name resolution

Enable name resolution. If you want the addresses to be resolved into names select this option.

Enabling name resolution can slow Ethereal down increasing the risk of dropping packets.

Click OK when you have set the options you require.

As packet capture proceeds a breakdown of the running totals are displayed in the capture window.

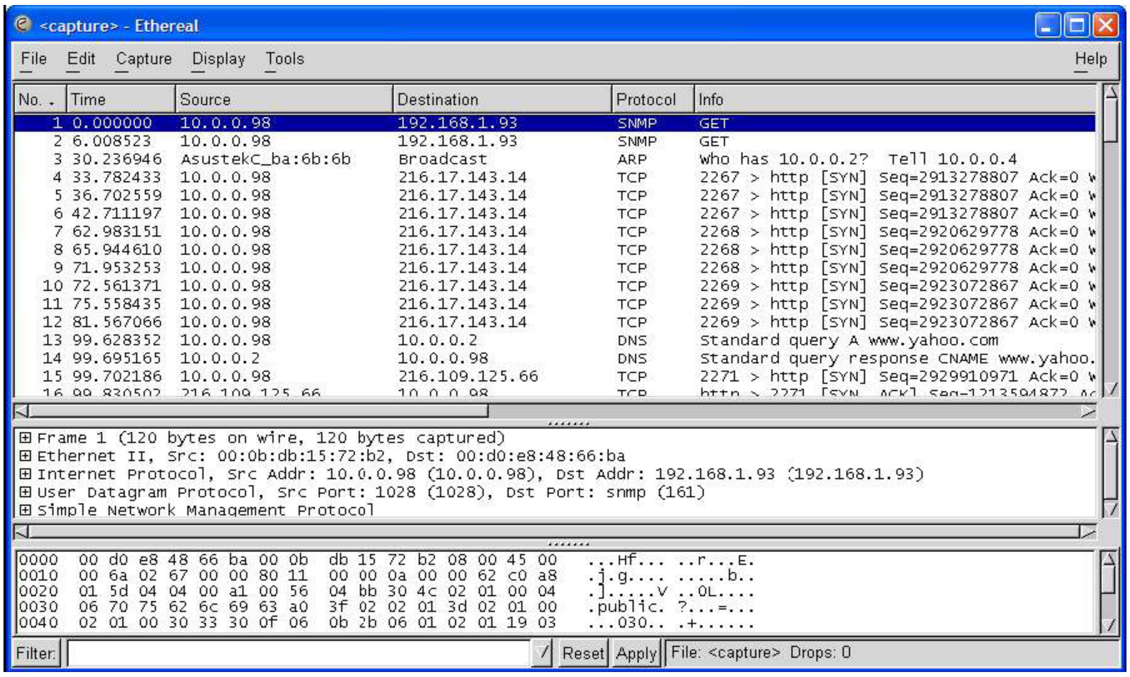


Captured Frames

Total	62	(100.0%)
SCTP	0	(0.0%)
TCP	52	(83.9%)
UDP	8	(12.9%)
ICMP	0	(0.0%)
ARP	2	(3.2%)
OSPF	0	(0.0%)
GRE	0	(0.0%)
NetBIOS	0	(0.0%)
IPX	0	(0.0%)
VINES	0	(0.0%)
Other	0	(0.0%)

Running 00:02:19

If you have set a value in the Capture limits options capturing will continue until that value is reached. If not you can press the Stop button at any time. When you stop capturing the opening screen now shows the captured packets.

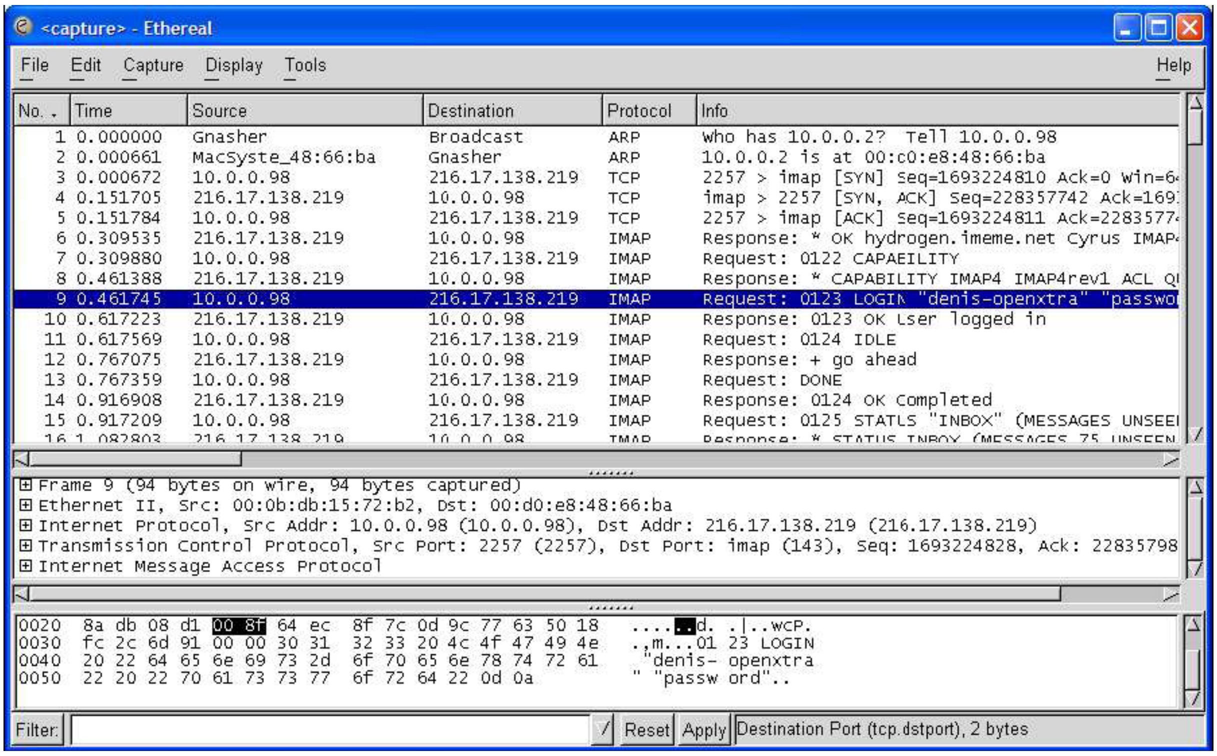


```
No.  Time       Source             Destination      Protocol  Info
  1 0.000000   10.0.0.98          192.168.1.93     SNMP      GET
  2 6.008523   10.0.0.98          192.168.1.93     SNMP      GET
  3 30.236946  AsustekC_ba:6b:6b  Broadcast        ARP       Who has 10.0.0.2?  Tell 10.0.0.4
  4 33.782433  10.0.0.98          216.17.143.14    TCP       2267 > http [SYN] Seq=2913278807 Ack=0 W
  5 36.702559  10.0.0.98          216.17.143.14    TCP       2267 > http [SYN] Seq=2913278807 Ack=0 W
  6 42.711197  10.0.0.98          216.17.143.14    TCP       2267 > http [SYN] Seq=2913278807 Ack=0 W
  7 62.983151  10.0.0.98          216.17.143.14    TCP       2268 > http [SYN] Seq=2920629778 Ack=0 W
  8 65.944610  10.0.0.98          216.17.143.14    TCP       2268 > http [SYN] Seq=2920629778 Ack=0 W
  9 71.953253  10.0.0.98          216.17.143.14    TCP       2268 > http [SYN] Seq=2920629778 Ack=0 W
 10 72.561371  10.0.0.98          216.17.143.14    TCP       2269 > http [SYN] Seq=2923072867 Ack=0 W
 11 75.558435  10.0.0.98          216.17.143.14    TCP       2269 > http [SYN] Seq=2923072867 Ack=0 W
 12 81.567066  10.0.0.98          216.17.143.14    TCP       2269 > http [SYN] Seq=2923072867 Ack=0 W
 13 99.628352  10.0.0.98          10.0.0.2         DNS       Standard query A www.yahoo.com
 14 99.695165  10.0.0.2           10.0.0.98        DNS       Standard query response CNAME www.yahoo.
 15 99.702186  10.0.0.98          216.109.125.66   TCP       2271 > http [SYN] Seq=2929910971 Ack=0 W
 16 99.830502  216.109.125.66     10.0.0.98        TCP       http > 2271 [SYN, ACK] Seq=1213594872 Ac

⊞ Frame 1 (120 bytes on wire, 120 bytes captured)
⊞ Ethernet II, Src: 00:0b:db:15:72:b2, Dst: 00:d0:e8:48:66:ba
⊞ Internet Protocol, Src Addr: 10.0.0.98 (10.0.0.98), Dst Addr: 192.168.1.93 (192.168.1.93)
⊞ User Datagram Protocol, Src Port: 1028 (1028), Dst Port: snmp (161)
⊞ Simple Network Management Protocol

0000   00 d0 e8 48 66 ba 00 0b  db 15 72 b2 08 00 45 00   ...Hf... ..r...E.
0010   00 6a 02 67 00 00 80 11  00 00 0a 00 00 62 c0 a8   .j.g.... .....b..
0020   01 5d 04 04 00 a1 00 56  04 bb 30 4c 02 01 00 04   .].....V ..0L....
0030   06 70 75 62 6c 69 63 a0  3f 02 02 01 3d 02 01 00   .public. ?...=...
0040   02 01 00 30 33 30 0f 06  0b 2b 06 01 02 01 19 03   ...030.. .+......

File: <capture>  Drops: 0
```

Scroll through the list to find the packet you are interested in and click on it to see the details.

You can also resize the windows if required by pointing your cursor at the bars between panes. Click and hold the left mouse button, and drag the bar to the required position.

Opening an existing file
Click on File, Open. Select a file and press Enter.
Ethereal can read packet capture files from tcpdump using the -w option, and uses the same filtering system and syntax.

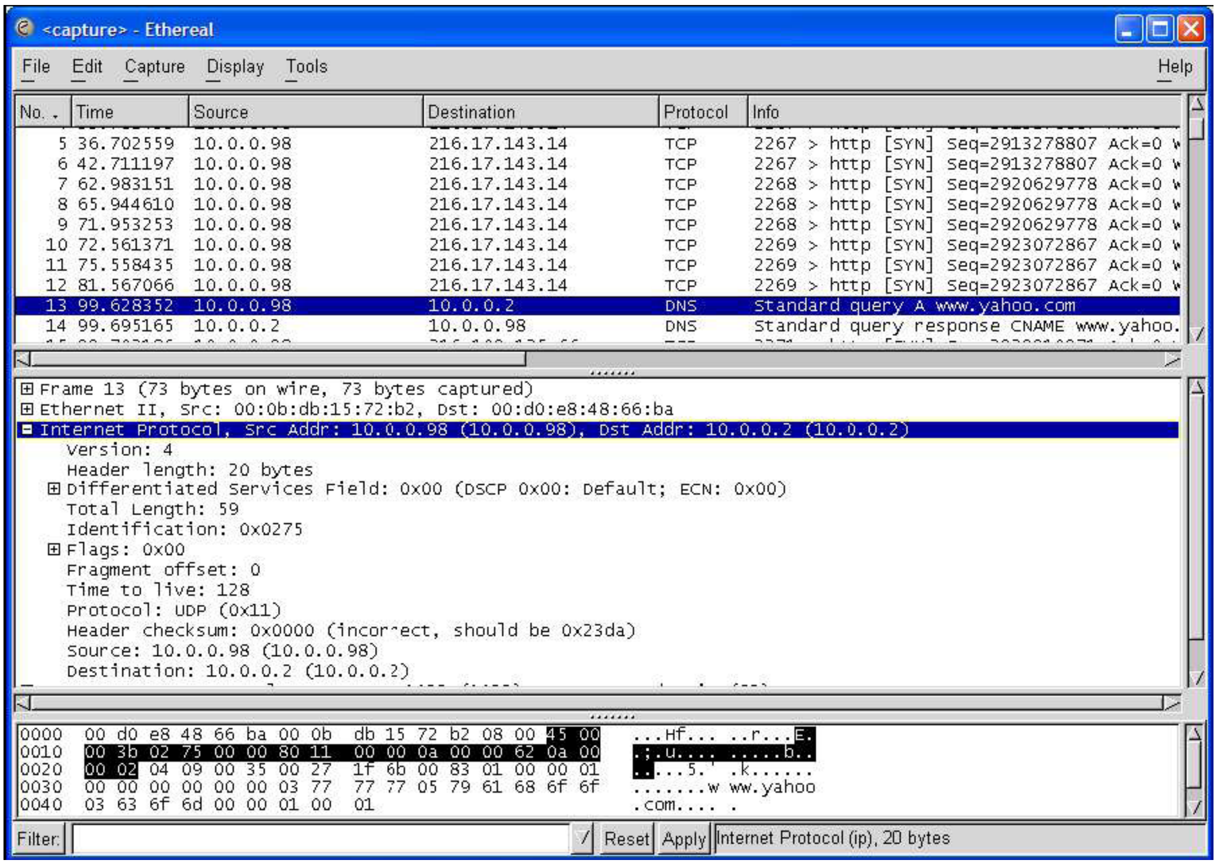


```
No. .  Time      Source              Destination       Protocol  Info
     1 0.000000  Gnasher             Broadcast         ARP       who has 10.0.0.2?  Tell 10.0.0.98
     2 0.000661  MacSyste_48:66:ba   Gnasher           ARP       10.0.0.2 is at 00:c0:e8:48:66:ba
     3 0.000672  10.0.0.98           216.17.138.219    TCP       2257 > imap [SYN] Seq=1693224810 Ack=0 Win=6
     4 0.151705  216.17.138.219      10.0.0.98         TCP       imap > 2257 [SYN, ACK] Seq=228357742 Ack=169
     5 0.151784  10.0.0.98           216.17.138.219    TCP       2257 > imap [ACK] Seq=1693224811 Ack=2283577
     6 0.309535  216.17.138.219      10.0.0.98         IMAP      Response: * OK hydrogen.imeme.net Cyrus IMAP
     7 0.309880  10.0.0.98           216.17.138.219    IMAP      Request: 0122 CAPAEILITY
     8 0.461388  216.17.138.219      10.0.0.98         IMAP      Response: * CAPABILITY IMAP4 IMAP4rev1 ACL Q
     9 0.461745  10.0.0.98           216.17.138.219    IMAP      Request: 0123 LOGIN "denis-openxtra" "passwo
    10 0.617223  216.17.138.219      10.0.0.98         IMAP      Response: 0123 OK User logged in
    11 0.617569  10.0.0.98           216.17.138.219    IMAP      Request: 0124 IDLE
    12 0.767075  216.17.138.219      10.0.0.98         IMAP      Response: + go ahead
    13 0.767359  10.0.0.98           216.17.138.219    IMAP      Request: DONE
    14 0.916908  216.17.138.219      10.0.0.98         IMAP      Response: 0124 OK Completed
    15 0.917209  10.0.0.98           216.17.138.219    IMAP      Request: 0125 STATUS "INBOX" (MESSAGES UNSEE
    16 1.082803  216.17.138.219      10.0.0.98         IMAP      Response: * STATUS INBOX (MESSAGES 75 UNSEEN

⊞ Frame 9 (94 bytes on wire, 94 bytes captured)
⊞ Ethernet II, Src: 00:0b:db:15:72:b2, Dst: 00:d0:e8:48:66:ba
⊞ Internet Protocol, Src Addr: 10.0.0.98 (10.0.0.98), Dst Addr: 216.17.138.219 (216.17.138.219)
⊞ Transmission Control Protocol, Src Port: 2257 (2257), Dst Port: imap (143), Seq: 1693224828, Ack: 22835798
⊞ Internet Message Access Protocol

0020  8a db 08 d1 00 8f 64 ec  8f 7c 0d 9c 77 63 50 18   ......d. .|..wcP.
0030  fc 2c 6d 91 00 00 30 31  32 33 20 4c 4f 47 49 4e   .,m...01 23 LOGIN
0040  20 22 64 65 6e 69 73 2d  6f 70 65 6e 78 74 72 61    "denis- openxtra
0050  22 20 22 70 61 73 73 77  6f 72 64 22 0d 0a         " "passw ord"..
```

When a file is open the top pane shows a list of the packets, with times, addresses, protocols and summary information, the central pane shows a detailed breakdown of the highlighted packet protocols, and the bottom pane shows the raw hex and ASCII data in the packet.

Displaying multiple packets Click Display, Show Packet in New Window, to view a packet in a separate window. Using this allows you to view details of several packets
at once.

Viewing protocol details The central pane shows the protocols in the highlighted packet. Click the plus or minus sign to expand or collapse the protocols. *Display has an option to expand and collapse all protocols with a single key press.*



```
No. . Time       Source      Destination    Protocol Info
  5 36.702559  10.0.0.98   216.17.143.14  TCP      2267 > http [SYN] Seq=2913278807 Ack=0 W
  6 42.711197  10.0.0.98   216.17.143.14  TCP      2267 > http [SYN] Seq=2913278807 Ack=0 W
  7 62.983151  10.0.0.98   216.17.143.14  TCP      2268 > http [SYN] Seq=2920629778 Ack=0 W
  8 65.944610  10.0.0.98   216.17.143.14  TCP      2268 > http [SYN] Seq=2920629778 Ack=0 W
  9 71.953253  10.0.0.98   216.17.143.14  TCP      2268 > http [SYN] Seq=2920629778 Ack=0 W
 10 72.561371  10.0.0.98   216.17.143.14  TCP      2269 > http [SYN] Seq=2923072867 Ack=0 W
 11 75.558435  10.0.0.98   216.17.143.14  TCP      2269 > http [SYN] Seq=2923072867 Ack=0 W
 12 81.567066  10.0.0.98   216.17.143.14  TCP      2269 > http [SYN] Seq=2923072867 Ack=0 W
 13 99.628352  10.0.0.98   10.0.0.2       DNS      Standard query A www.yahoo.com
 14 99.695165  10.0.0.2    10.0.0.98      DNS      Standard query response CNAME www.yahoo.
```

```
⊞ Frame 13 (73 bytes on wire, 73 bytes captured)
⊞ Ethernet II, Src: 00:0b:db:15:72:b2, Dst: 00:d0:e8:48:66:ba
⊟ Internet Protocol, Src Addr: 10.0.0.98 (10.0.0.98), Dst Addr: 10.0.0.2 (10.0.0.2)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 59
    Identification: 0x0275
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
    Header checksum: 0x0000 (incorrect, should be 0x23da)
    Source: 10.0.0.98 (10.0.0.98)
    Destination: 10.0.0.2 (10.0.0.2)
```

```
0000  00 d0 e8 48 66 ba 00 0b  db 15 72 b2 08 00 45 00   ...Hf... ..r...E.
0010  00 3b 02 75 00 00 80 11  00 00 0a 00 00 62 0a 00   .;.u.... .....b..
0020  00 02 04 09 00 35 00 27  1f 6b 00 83 01 00 00 01   .....5.' .k......
0030  00 00 00 00 00 00 03 77  77 77 05 79 61 68 6f 6f   .......w ww.yahoo
0040  03 63 6f 6d 00 00 01 00  01                        .com.... .
```

Display options

The display options in *Ethereal* are very powerful. Options... allows you to specify the format of the Time field and the type of Name resolution required. The box at the bottom of the opening screen allows you to specify display filters. Display filters can be very complex, but the good news is that you do not have to know a great deal about the syntax to make useful filters. By far the easiest way to build display filters is from inside the protocol
details view. Display Match, Selected

This very useful for viewing filtered selections from the full packet trace.There are a number of powerful options.

Step 1 Click on a field inside the central protocol pane.Not all selections make sensible filters but with a bit of trial and error you will quickly learn what makes works and what does not.
Step 2 Click Display, Match, Selection.

A filtered list of packets appears.
Other options allow you to reverse the filter, and perform other boolean operations.
The Filter details appear at the bottom of the screen.To clear the Filter press the Reset button. You can save and name your Filters using the options under Filter.*An alternative way to filter a file is to simply type the name of the protocol in the Filter box and click on Apply.*

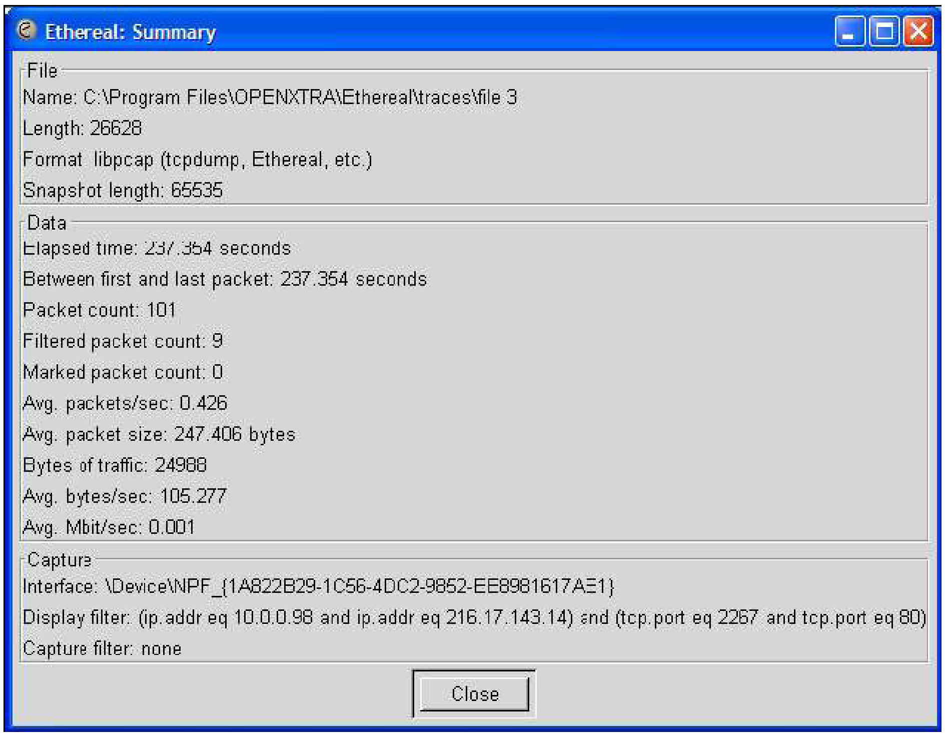
Making Display filters by hand
Step 1 Click Edit, Display Filters...
Step 2 Type a name for the filter in the Filter name box.
Step 3 Add a Filter string.
You can use the Add Expression button to make this easier.Click it, go to the protocol you are interested in, click that and select a field. Type a value.
Step 4 Click New.

Editing Display Filters

Step 1 Click Edit, Display Filters...
Step 2 Click the filter you want to edit.
Step 3 Edit the Filter string.
You can use the Add Expression button to make this easier.
Click it, go to the protocol you are interested in, click that
and select a field. Type a value.
Step 4 Click Change.
Step 5 Click Close.

Tools options

The Tools options allow you to perform advanced analysis on the packets in the file. This section describes some of the simpler options.
Summary
Click Tools, Summary, for a breakdown of the data that you have collected.



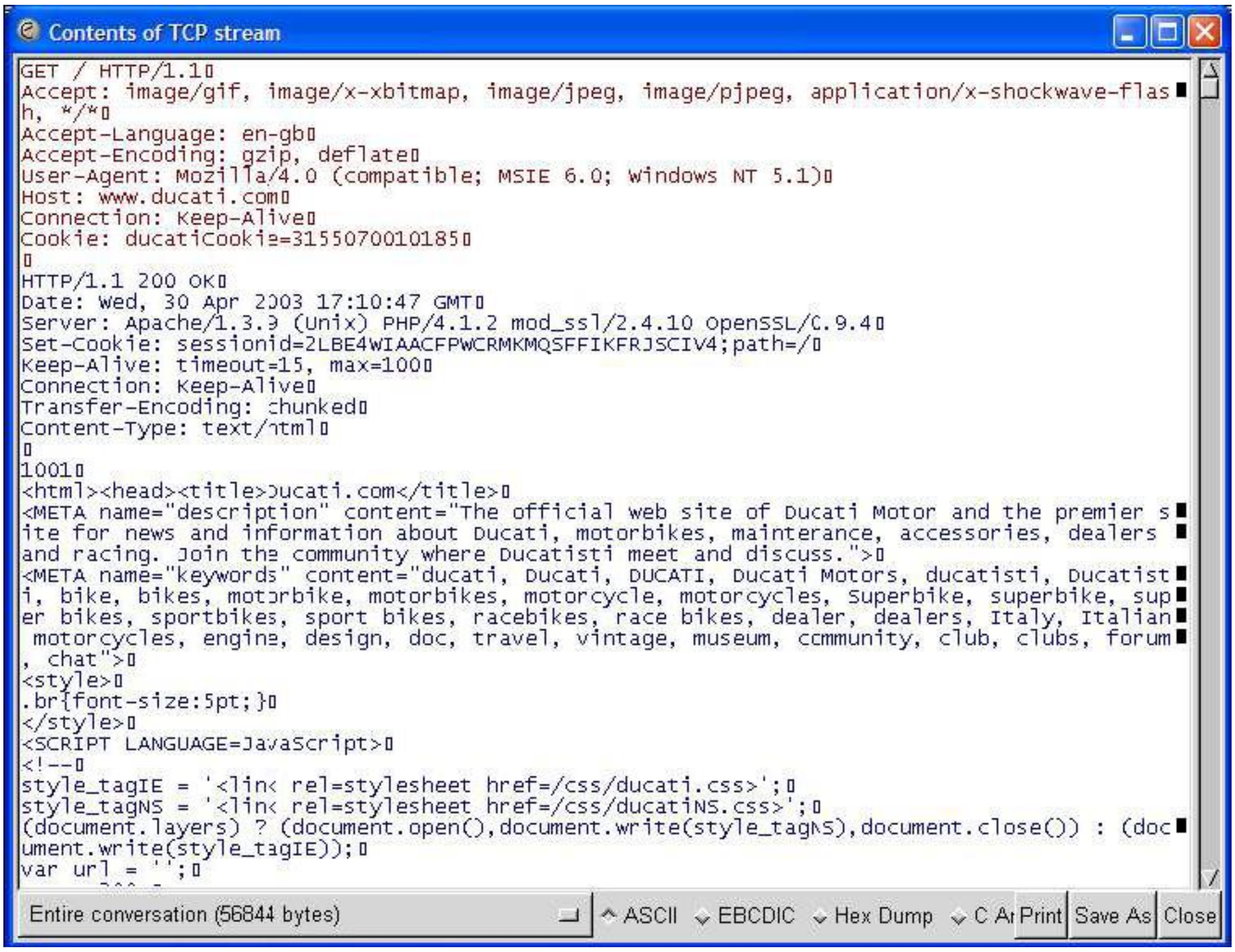
File
Name: C:\Program Files\OPENXTRA\Ethereal\traces\file 3
Length: 26628
Format libpcap (tcpdump, Ethereal, etc.)
Snapshot length: 65535

Data
Elapsed time: 237.354 seconds
Between first and last packet: 237.354 seconds
Packet count: 101
Filtered packet count: 9
Marked packet count: 0
Avg. packets/sec: 0.426
Avg. packet size: 247.406 bytes
Bytes of traffic: 24988
Avg. bytes/sec: 105.277
Avg. Mbit/sec: 0.001

Capture
Interface: \Device\NPF_{1A822B29-1C56-4DC2-9852-EE8981617AE1}
Display filter: (ip.addr eq 10.0.0.98 and ip.addr eq 216.17.143.14) and (tcp.port eq 2267 and tcp.port eq 80)
Capture filter: none

Follow TCP Stream

Click Tools, Follow TCP Stream. This option tracks the information in a TCP conversation and makes it easy to follow the traffic between two endpoints.



```
GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flas
h, */*
Accept-Language: en-gb
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: www.ducati.com
Connection: Keep-Alive
Cookie: ducatiCookie=3155070010185

HTTP/1.1 200 OK
Date: Wed, 30 Apr 2003 17:10:47 GMT
Server: Apache/1.3.9 (Unix) PHP/4.1.2 mod_ssl/2.4.10 OpenSSL/0.9.4
Set-Cookie: sessionid=2LBE4WIAACFPWCRMKMQSFFIKFRJSCIV4;path=/
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html

1001
<html><head><title>Ducati.com</title>
<META name="description" content="The official web site of Ducati Motor and the premier s
ite for news and information about Ducati, motorbikes, maintenance, accessories, dealers
and racing. Join the community where Ducatisti meet and discuss.">
<META name="keywords" content="ducati, Ducati, DUCATI, Ducati Motors, ducatisti, Ducatist
i, bike, bikes, motorbike, motorbikes, motorcycle, motorcycles, superbike, superbike, sup
er bikes, sportbikes, sport bikes, racebikes, race bikes, dealer, dealers, Italy, Italian
 motorcycles, engine, design, doc, travel, vintage, museum, community, club, clubs, forum
, chat">
<style>
.br{font-size:5pt;}
</style>
<SCRIPT LANGUAGE=JavaScript>
<!--
style_tagIE = '<link rel=stylesheet href=/css/ducati.css>';
style_tagNS = '<link rel=stylesheet href=/css/ducatiNS.css>';
(document.layers) ? (document.open(),document.write(style_tagNS),document.close()) : (doc
ument.write(style_tagIE));
var url = '';
```

EXERCISE

1. The 48-bit destination address in Ethernet frame is made of all Fs. Why?

2. The 48-bit Target MAC address in ARP packet is all 0s. Why?

3. Write down the importance of TCP stream Capturing in ETHEREAL .With the help of Real World example.

Lab Session 12

OBJECT: Folder Security Implementation Through BATCH Programming

THEORY

In DOS, OS/2, and Microsoft Windows, a batch file is a text file containing a series of commands intended to be executed by the command interpreter. When a batch file is run, the shell program (usually COMMAND.COM or cmd.exe) reads the file and executes its commands, normally line-by-line. Batch files are useful for running a sequence of executables automatically and are often used by system administrators to automate tedious processes.

DOS batch files have the filename extension .bat (or .BAT because file names are case insensitive in DOS, Windows and OS/2). There are a number of subtle differences between the .bat and .cmd extensions which affect the error levels returned, among other things; it is a common myth that .bat are run by COMMAND.COM while .cmd are run by CMD.EXE - this is not true and is not the reason for the differences. Also, the Windows 9x family only recognizes the .bat extension.

In this lab session we are going to implement folder security through Batch Programming

PROCEDURE

1. Create a New Folder and named it “test”.

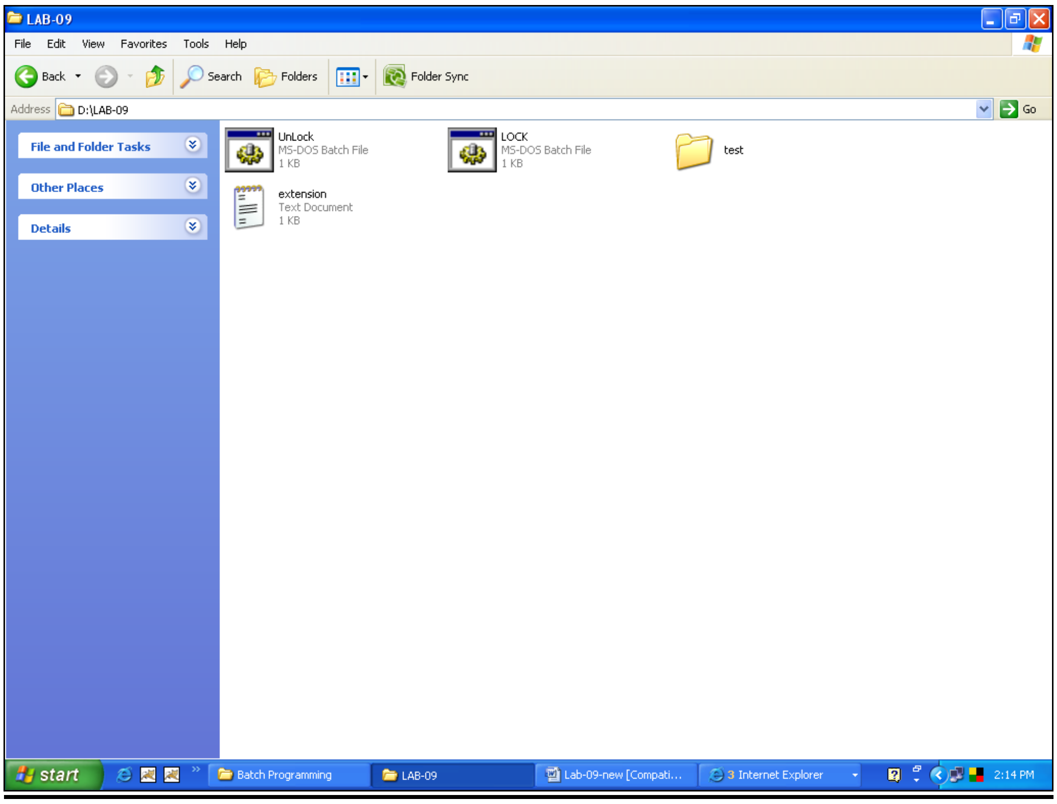
2. Move your files that you want to secure in to that folder.

3. Write the following script in a notepad to secure your folder and save it with the name “LOCK.bat”

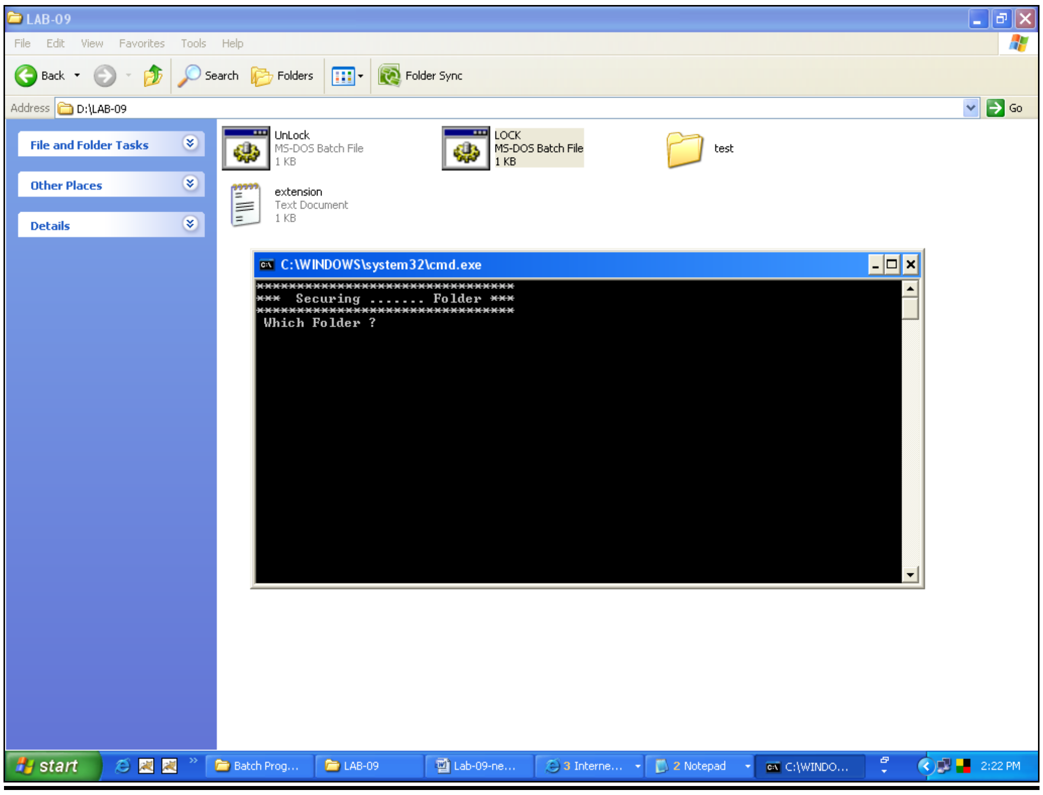
```
@echo off
echo ********************************
echo ***  Securing ....... Folder ***
echo ********************************


set /p folder= Which Folder ?

ren %folder% %folder.{20D04FE0-3AEA-1069-A2D8-08002B30309D}

echo Done
```

4. Write the following script in a notepad to UN-Lock your folder and save it with the name "UNLOCK.bat"

```
@echo off

set /p folder= Which Folder To be UnLock ?
ren %folder%.{20D04FE0-3AEA-1069-A2D8-08002B30309D} test
```

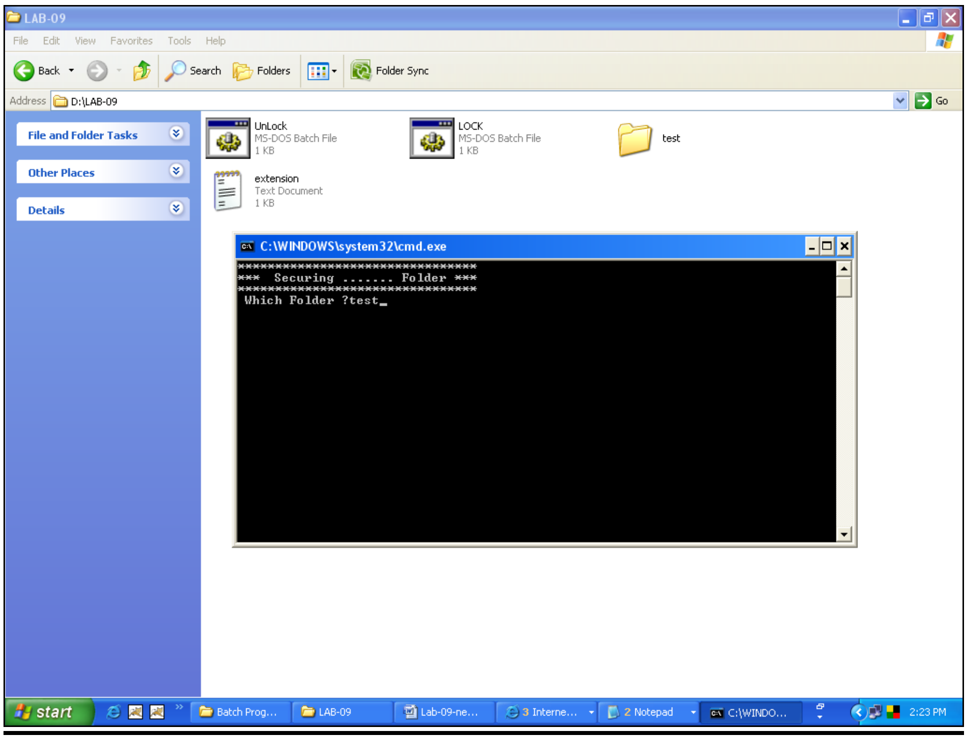


5. Now Double Click on LOCK.bat file


LAB-09
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Address D:\LAB-09
File and Folder Tasks
Other Places
Details
UnLock MS-DOS Batch File 1 KB
LOCK MS-DOS Batch File 1 KB
test
extension Text Document 1 KB
C:\WINDOWS\system32\cmd.exe
start Batch Prog... LAB-09 Lab-09-ne... 3 Interne... 2 Notepad C:\WINDO... 2:22 PM

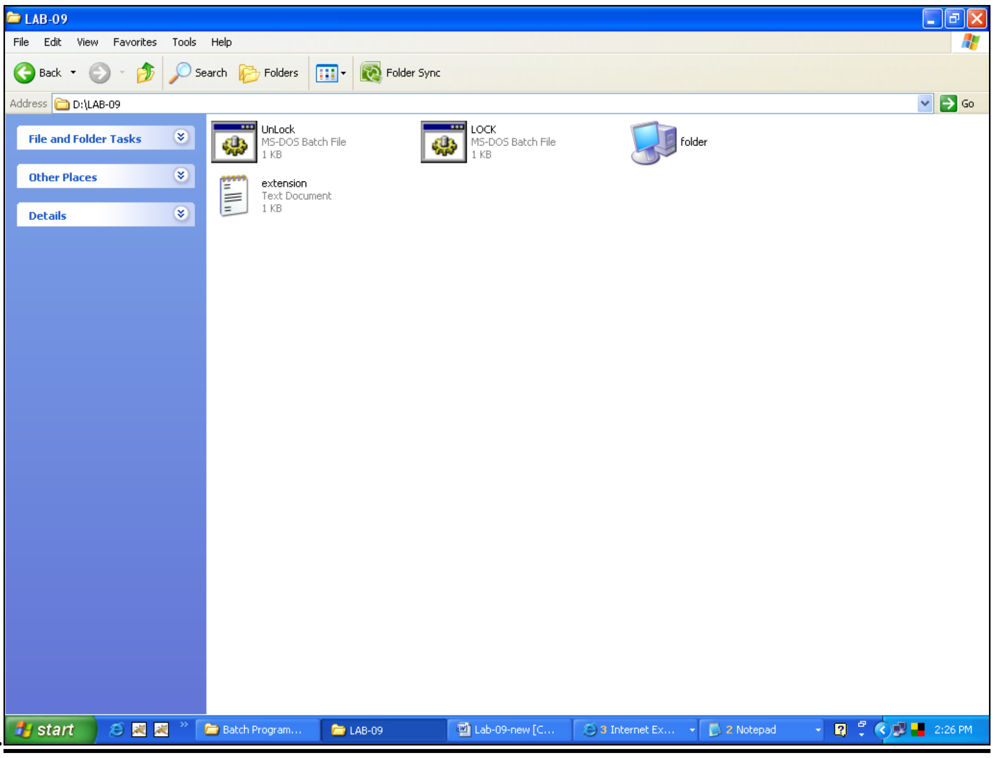
```
*********************************
***  Securing ....... Folder ***
*********************************
 Which Folder ?
```

6. Write the Name of folder that you want to secure. In our scenario it is "test" and then press Enter.


LAB-09
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Address D:\LAB-09
File and Folder Tasks
Other Places
Details
UnLock MS-DOS Batch File 1 KB
LOCK MS-DOS Batch File 1 KB
test
extension Text Document 1 KB
C:\WINDOWS\system32\cmd.exe
start Batch Prog... LAB-09 Lab-09-ne... 3 Interne... 2 Notepad C:\WINDO... 2:23 PM

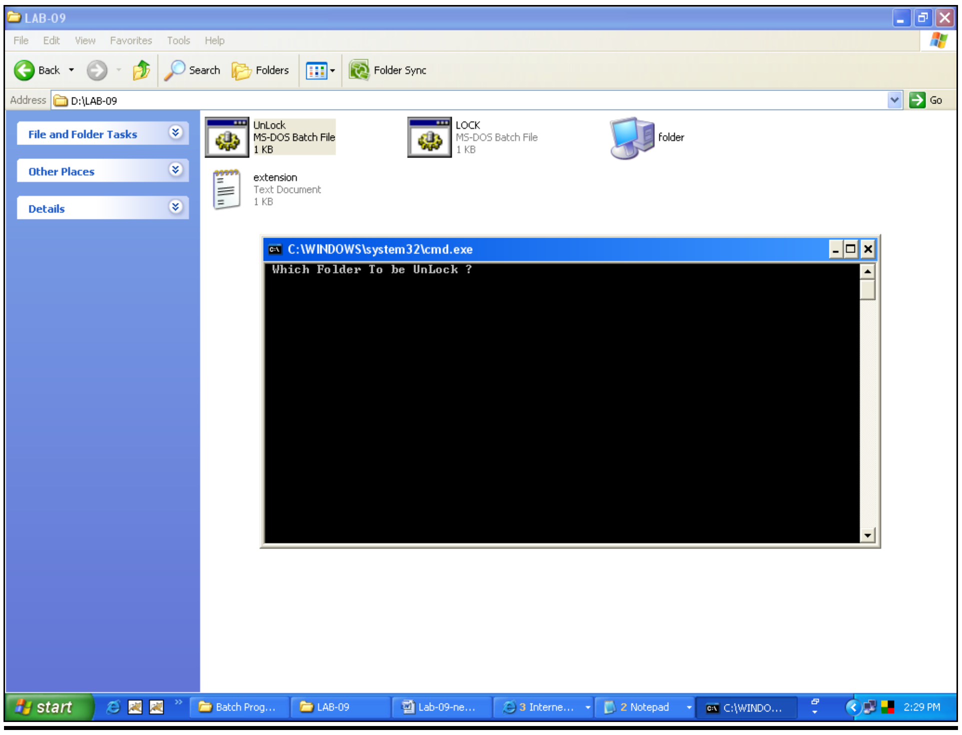
```
*********************************
***  Securing ....... Folder ***
*********************************
 Which Folder ?test_
```

7. After pressing enter your test folder is renamed as folder and representation is converted in to MY Computer.


LAB-09
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Address D:\LAB-09 Go
File and Folder Tasks
Other Places
Details
UnLock MS-DOS Batch File 1 KB
LOCK MS-DOS Batch File 1 KB
folder
extension Text Document 1 KB
start
Batch Program...
LAB-09
Lab-09-new [C...
3 Internet Ex...
2 Notepad
2:26 PM

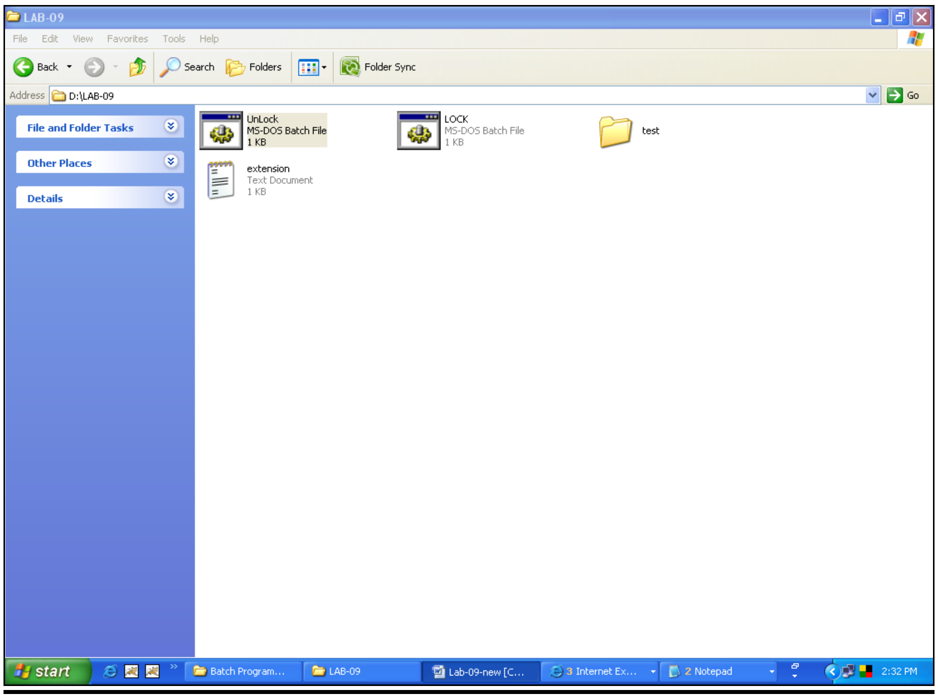
8. If you double click on “Folder” it directly open My Computer .Means the files that you put in test folder are not accessible and are being secure.

9. To retrieve your test folder double click on Unlock.bat file



```
Which Folder To be UnLock ?
```

10. Write the name of folder like in our scenario is "Folder" and then press Enter.



EXERCISE

4. In the DOS Batch Programming what does @ECHO OFF means.

5. What Command you use to take input in Batch Programming.

6. Write the command to rename NED.bat file to BCIT.bat file.

Lab Session 13

OBJECT: **Network Traffic Processing and Analysis in promiscuous mode.**

THEORY

Computers and network devices communicate with each other by sending or receiving information over tiny bundles of electronic signals called packets. This is how a packet is represented while on wire. In memory, a packet is represented by 1s and 0s.

Flow of different packets to and from different computers over the network is said to constitute network traffic. Sniffers are tools which are used to capture and process network traffic.

- Wireshark
- Ipgrab
- Ettercap
- Dsniff
- Tcpdump

All are free and open source

PROCEDURE

Apply the following commands in wireshark and observe the result.

- **$ man tshark**

  You must be root to run tshark and most other sniffers

- **$ tshark –v**

  Prints version and other information
  Can be used without root

- **$ tshark –h**

  Prints help summary
  Can be used without root

- **# tshark –D**

Lists all the interfaces available on the host

- **# tshark –i eth0**

  Capture on interface eth0

- **#tshark –V**

  Verbose output

- **#tshark –c 100**

  Stop after capturing 100 packets
  Default is infinite
  Use ctrl-c to terminate

- **#tshark –r outputfile.pcap**

  Read packets from the outputfile.pcap
  root not needed
  Useful for experimentation

- **#tshark –r outputfile.pcap > textfile.txt**

  You can create a text file of the output

- **#tshark –x**

  Also output in hex and ASCII

- **#tshark –n**

  By default names are resolved
  It generates noise on the network
  -n disables name resolution
  Avoids detection from anti sniffers

- **#tshark –p**

  Don't put the interface in prosmiscuous mode

Promiscuous mode

A number of packets arrive at an NIC, By default, the NIC chooses only those packets whose destination MAC address matches its own I.e. it captures only packets which are meant for the NIC Foreign packets are discarded.

When in promiscuous mode, the NIC captures all the packets regardless of the destination MAC address in the packet. This way it also captures which are meant for other NICs I.e. other computers. In promiscuous mode, the NIC selects and captures all the packets. None are discarded.

A sniffer running in promiscuous mode in a non-switched LAN will capture ALL the traffic of the network including those of other computers.
In a LAN using a switched hub, a packet is sent only to the NIC for which it is meant. A packet is not broadcast to all the host as in case of a non-switched hub. A sniffer running even in promiscuous mode will not be able to capture traffic of other computers unless traffic is redirected in some way.

Filter Expressions

By default a sniffer captures all the packets arriving on the interface. To capture only selective packets we use filter expressions with a sniffer. Filter expressions are characteristics are sniffers which are built using libpcap. Most of the sniffers are built using libpcap

- **# tshark host 172.16.30.1**

  This expression will captures only packets which are meant for host 72.16.30.1

- **# tshark src host 172.16.30.1**

  This expression will capture packets which originates from host 172.16.30.1

- **# tshark tcp**

  This will only capture packets with TCP header in them. Rest of the packets are not decoded. Similarly you can use ‘udp’, ‘ip’, ‘arp’, ‘icmp’, etc. instead of tcp.

- **#tshark src host 172.16.30.1 and dst port 80**

  This captures all outgoing HTTP traffic from host 172.16.30.1. An HTTP server uses port 80 to listen to HTTP traffic.A browser may use any ephemereal port above 1024 to connect to HTTP server

- **#tshark ether src host 00:0c:f1:dc:6a:a7**

  This captures packets from the host whose MAC address is as given.

- **#tshark net 172.16**

  This will capture all the traffic meant for network beginneing with 172.16
  You can also use 'src net' and 'dst net'.

- **#tshark less 500**

  This will capture all the packets whose length is less than or equal to 500 bytes.
  Similarly you can use 'greater 500'.

- **Filter expressions can be combined using and, or, not operators.**

- **#tshark "host 172.16.30.1 and (host 172.16.30.5 or host 172.16.30.10)"**

  This will capture any communication of 172.16.30.1 with either 172.16.30.5 or 172.16.30.10. Rest of the traffic will be discarded .Enclose complex filter expressions in " ".

EXERCISE

7. What is Promiscuous Mode ?

____
____
____
____

8. Write the expression that will captures only packets which are meant for host 72.16.30.1.

____
____
____

9. Describe this filter "**tshark src host 172.16.30.1 and dst port 80**" in your own words.

____
____
____
____

Lab Session 14

OBJECT: NMAP - A Stealth Port Scanner

THEORY

Nmap is a free, open-source port scanner available for both UNIX and Windows. It has an optional graphical front-end, ZENmap, and supports a wide variety of scan types, each one with different benefits and drawbacks.

The two basic scan types used most in Nmap are

1. Basic Scan Types [-sT, -sS]
2. TCP connect() scanning [-sT].

SYN scanning (also known as half-open, or stealth scanning) [-sS].

TCP connect() Scan [-sT]

These scans are so called because UNIX sockets programming uses a system call named connect()} to begin a TCP connection to a remote site. If texttt{connect()} succeeds, a connection was made. If it fails, the connection could not be made (the remote system is offline, the port is closed, or some other error occurred along the way). This allows a basic type of port scan, which attempts to connect to every port in turn, and notes whether or not the connection succeeded. Once the scan is completed, ports to which a connection could be established are listed as textit{open, the rest are said to be closed.
This method of scanning is very effective, and provides a clear picture of the ports you can and cannot access. If a connect() scan lists a port as open, you can definitely connect to it - that is what the scanning computer just did! There is, however, a major drawback to this kind of scan; it is very easy to detect on the system being scanned. If a firewall or intrusion detection system is running on the victim, attempts to connect() to every port on the system will almost always trigger a warning. Indeed, with modern firewalls, an attempt to connect to a single port which has been blocked or has not been specifically "opened" will usually result in the connection attempt being logged. Additionally, most servers will log connections and their source IP, so it would be easy to detect the source of a TCP connect() scan.

For this reason, the TCP Stealth Scan was developed.

- ☐ **SYN Stealth Scan [-sS]**

  TCP Connection -- Three way handshake
- ☐ **Flag**

  SYN (Synchronize).

  ACK (Acknowledge).

  FIN (Finished) and

  RST (Reset).

SYN or Stealth scanning makes use of this procedure by sending a SYN packet and looking at the response. If SYN/ACK is sent back, the port is open and the remote end is trying to open a TCP connection. The scanner then sends an RST to tear down the connection before it can be established fully; often preventing the connection attempt appearing in application logs. If the port is closed, an RST will be sent. If it is filtered, the SYN packet will have been dropped and no response will be sent. In this way, Nmap can detect three port states - open, closed and filtered. Filtered ports may require further probing since they could be subject to firewall rules which render them open to some IPs or conditions, and closed to others.

PROCEDURE

Modern firewalls and Intrusion Detection Systems can detect SYN scans, but in combination with other features of Nmap, it is possible to create a virtually undetectable SYN scan by altering timing and other options .Apply the following commands on NMAP and observe the results.

- ☐ **Ping Scan [-sP]**

  This scan type lists the hosts within the specified range that responded to a ping. It allows you to detect which computers are online, rather than which ports are open.

UDP Scan [-sU]

Scanning for open UDP ports is done with the -sU option. With this scan type, Nmap sends 0-byte UDP packets to each target port on the victim. Receipt of an ICMP Port Unreachable message signifies the port is closed, otherwise it is assumed open.

- ☐ **IP Protocol Scans [-sO]**

  The IP Protocol Scans attempt to determine the IP protocols supported on a target. Nmap sends a raw IP packet without any additional protocol header (see a good TCP/IP book for information about IP packets), to each protocol on the target machine. Receipt of an ICMP Protocol Unreachable message tells us the protocol is not in use, otherwise it is assumed open.

- ☐ **Version Detection [-sV]**

  Version Detection collects information about the specific service running on an open port, including the product name and version number. This information can be critical in determining an entry point for an attacker.

- ☐ **OS Fingerprinting**

  The -O option turns on Nmap's OS fingerprinting system. Used alongside the -v verbosity options, you can gain information about the remote operating system and about its TCP Sequenmce Number generation

- ☐ **IPv6**

  The -6 option enables IPv6 in Nmap (provided your OS has IPv6 support). Currently only TCP connect, and TCP connect ping scan are supported.

- **Verbose Mode**

  Highly recommended, -v

  Use -v twice for more verbosity. The option -d can also be used (once or twice) to generate more verbose output.

EXERCISE

10. What is the difference between a TCP-connect scan and a SYN scan?.

11. What is the purpose of the –sP command line switch?

12. What is the purpose of the –sS command line switch?

13. What command would you issue to scan for computers running web servers?