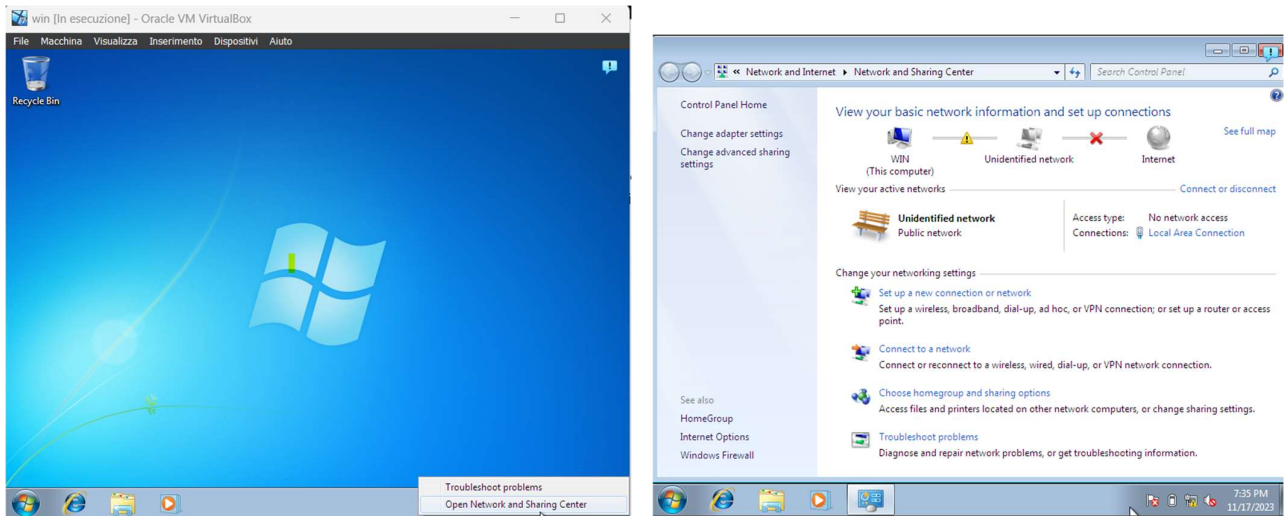


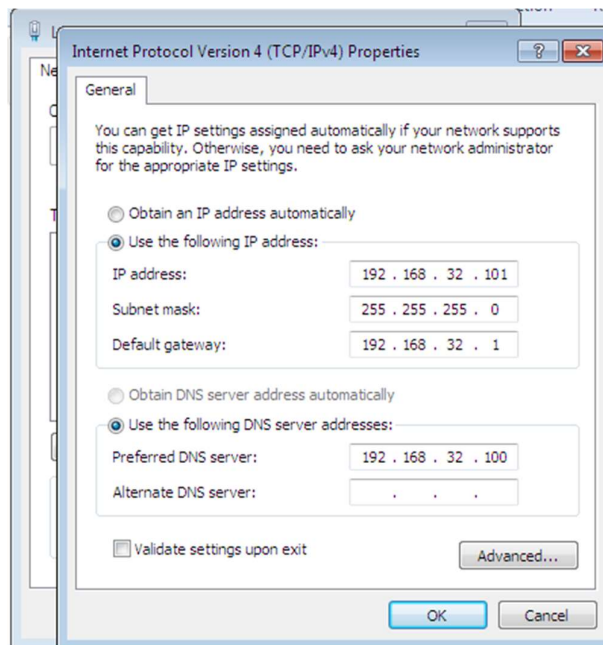
Esercizio W4D4

Per lo svolgimento di questo esercizio come primo passaggio provvedo alla configurazione delle schede di rete delle due macchine.

Per Win7 procediamo nel seguente modo:



Imposto indirizzo IP, Subnet mask, gateway e il DNS relativo all'indirizzo IP della macchina Kali sul quale appunto oltre ai servizi di https e http avvieremo anche il servizio DNS.



Per quanto riguarda Kali, apriremo da terminale il file interfaces per configurare la scheda di rete:

```
kali@kali: /etc/network
File Actions Edit View Help
(kali@kali)~$ cd /etc/network/
(kali@kali)~/etc/network$ ls -lrt
total 36
drwxr-xr-x 2 root root 4096 Jan 24 2023 interfaces.d
drwxr-xr-x 2 root root 4096 Aug 21 14:53 if-down.d
drwxr-xr-x 2 root root 4096 Aug 21 14:58 if-up.d
drwxr-xr-x 2 root root 4096 Aug 21 14:58 if-pre-up.d
drwxr-xr-x 2 root root 4096 Aug 21 14:58 if-post-down.d
-rw-rw-r-- 1 root root 247 Oct 27 13:52 interfaces.save
-rw-rw-r-- 1 root root 346 Oct 27 14:16 interfaces.save.1
-rw-r--r-- 1 root root 345 Oct 27 14:21 interfacesc
-rw-rw-r-- 1 root root 386 Nov 16 16:55 interfaces
(kali@kali)~/etc/network$ sudo nano interfaces
[sudo] password for kali:
```

```
GNU nano 7.2 interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.32.100
netmask 255.255.255.0
network 192.168.32.0
broadcast 192.168.32.255
gateway 192.168.32.1
```

Procediamo poi con la configurazione di inetsim strumento progettato per simulare diversi servizi di rete tra cui appunto quelli richiesti dalla traccia.

```
kali@kali: /etc/inetsim
File Actions Edit View Help
(kali@kali)~$ cd /etc/inetsim
(kali@kali)~/etc/inetsim$ ls -lrt
total 44
-rw-r--r-- 1 root root 41715 Nov 16 17:24 inetsim.conf
(kali@kali)~/etc/inetsim$ sudo nano inetsim.conf
[sudo] password for kali:
```

Successivamente procedo a commentare tutti i servizi di rete non richiesti lasciando attivi solo quello di https, http e dns.

```
kali@kali: /etc/inetsim
File Actions Edit View Help
GNU nano 7.2 inetsim.conf
# start_service
#
# The services to start
# Syntax: start_service <service name>
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
#start_service smtp
#start_service smtps
#start_service pop3
#start_service pop3s
```

Imposto indirizzo di bind e configuro il dns:

```
#####
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 192.168.32.100
#####
```

```
#####
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.32.100
#####
```

```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
#dns_static epicode.internal 192.168.32.100
#####
```

Avvio inetsim:

```
(kali㉿kali)-[~]
$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 17154) ==
Session ID: 17154
Listening on: 192.168.32.100
Real Date/Time: 2023-11-17 13:57:51
Fake Date/Time: 2023-11-17 13:57:51 (Delta: 0 seconds)
Forking services ...
* dns_53_tcp_udp - started (PID 17164)
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
* http_80_tcp - started (PID 17165)
* https_443_tcp - started (PID 17166)
done.
Simulation running.
```

Da Win7 Provo ad preventivamente ad accedere alla risorsa tramite ping e poi con il web browser:

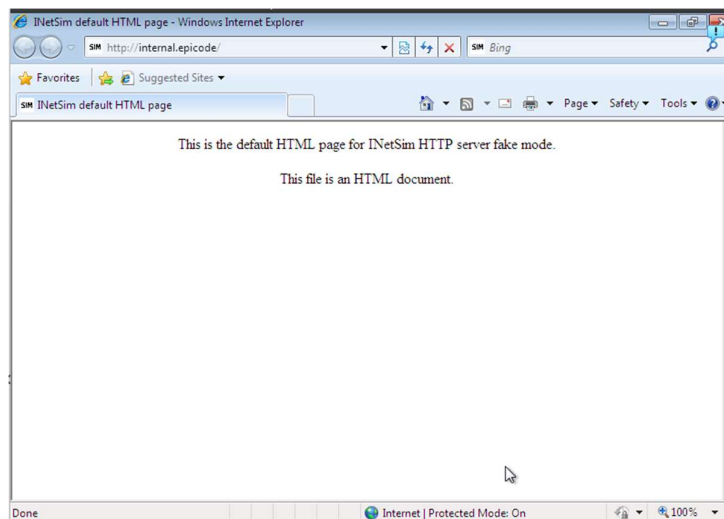
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\vboxuser>ping epicode.internal

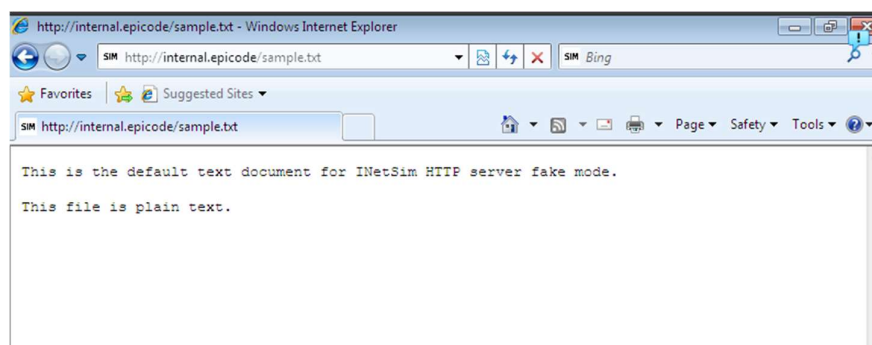
Pinging epicode.internal [192.168.32.100] with 32 bytes of data:
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.32.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\vboxuser>
```



Su Win7 procedo a richiedere il file sample.txt tramite http e intercetto la comunicazione con Wireshark su Kali sulla eth0:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_47:b0:59	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000017838	PcsCompu_cb:7e:f5	PcsCompu_47:b0:59	ARP	42	192.168.32.100 is at 08:00:27:cb:7e:f5
3	0.000022257	192.168.32.101	192.168.32.101	TCP	60	49174 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000750743	192.168.32.100	192.168.32.101	TCP	60	80 → 49174 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.001904924	192.168.32.101	192.168.32.100	TCP	60	49174 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.003220029	192.168.32.101	192.168.32.100	HTTP	454	GET /sample.txt HTTP/1.1
7	0.003240259	192.168.32.100	192.168.32.101	TCP	54	80 → 49174 [ACK] Seq=1 Ack=401 Win=64120 Len=0
8	0.0032433850	192.168.32.100	192.168.32.101	TCP	204	80 → 49174 [PSH, ACK] Seq=1 Ack=401 Win=64120 Len=150 [TCP segment of a reassembled PDU]
9	0.003253347	192.168.32.100	192.168.32.101	HTTP	151	HTTP/1.1 200 OK (text/plain)
10	0.003270157	192.168.32.101	192.168.32.100	TCP	60	49174 → 80 [ACK] Seq=401 Ack=249 Win=65452 Len=0
11	0.003270386	192.168.32.101	192.168.32.100	TCP	60	49174 → 80 [FIN, ACK] Seq=401 Ack=249 Win=65452 Len=0
12	0.003511228	192.168.32.100	192.168.32.101	TCP	54	80 → 49174 [ACK] Seq=249 Ack=402 Win=64120 Len=0
13	5.207215956	PcsCompu_cb:7e:f5	PcsCompu_47:b0:59	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
14	5.208052093	PcsCompu_47:b0:59	PcsCompu_cb:7e:f5	ARP	60	192.168.32.101 is at 08:00:27:47:b0:59

<pre> Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0 Ethernet II, Src: PcsCompu_47:b0:59 (08:00:27:47:b0:59), Dst: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5) Destination: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5) Source: PcsCompu_47:b0:59 (08:00:27:47:b0:59) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100 Transmission Control Protocol, Src Port: 49174, Dst Port: 80, Seq: 0, Len: 0 </pre>	<pre> 0000 08 00 27 cb 7e f5 08 00 27 47 b0 59 08 00 45 00 ...G.Y.E 0010 00 34 01 d9 40 00 80 06 36 da c0 a8 20 65 c0 a8 4 @ 6 e 0020 20 64 c0 16 00 50 62 5f cc 83 00 00 00 00 02 d _Pb_ 0030 20 00 0d b2 00 00 02 04 05 b4 01 03 03 02 01 01 ... 0040 04 02 </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Dalla schermata sopra possiamo vedere il protocollo ARP, il processo di scambio di pacchetti SYN, SYN-ACK e ACK (three way handshake) , fondamentale nel protocollo TCP/IP per stabilire una connessione tra due dispositivi su una rete e l'utilizzo della porta 80 (http). Possiamo successivamente evincere gli indirizzi MAC sorgente e destinatario:

- Source: 08:00:27:47:b0:59
- Destination: 08:00:27:cb:7e:f5

Ed infine essendo una richiesta http possiamo vedere in chiaro il contenuto del file richiesto:

No.	Time	Source	Destination	Protocol	Length	Info
34	0.372150306	192.168.32.101	192.168.32.100	TCP	66	49175 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM
35	0.37222224	192.168.32.100	192.168.32.101	TCP	66	80 → 49175 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
36	0.373193330	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
37	0.373193607	192.168.32.101	192.168.32.100	HTTP	284	GET /fwlink/?LinkId=121721&clcid=0x409&arch=x86&eng=0.0.0.0&asdelta=0.0.0.0&prod=925A3ACA-C353-458A-AC8B-AT7E5B378092 HTTP/1.1
38	0.373420840	192.168.32.100	192.168.32.101	TCP	64	80 → 49175 [ACK] Seq=1 Ack=231 Win=64128 Len=0
39	0.373863779	192.168.32.100	192.168.32.101	TCP	204	80 → 49175 [PSH, ACK] Seq=1 Ack=231 Win=64128 Len=150 [TCP segment of a reassembled PDU]
40	0.390745338	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
41	0.391050717	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [ACK] Seq=231 Ack=410 Win=65280 Len=0
42	0.391050954	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [FIN, ACK] Seq=231 Ack=410 Win=65280 Len=0
43	0.391091763	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=410 Ack=232 Win=64128 Len=0
44	0.504130354	PcsCompu_cb:7e:f5	PcsCompu_cb:7e:f5	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
45	0.505184577	PcsCompu_cb:7e:f5	PcsCompu_cb:7e:f5	ARP	60	192.168.32.101 is at 08:00:27:47:b0:59
46	159.544577318	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
47	160.543806120	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
48	162.546712937	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
49	166.552534804	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
50	174.553806219	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
51	199.557116304	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
52	222.56321711	fe80::18e5:20cf:692::ff02::1:2	ff02::1:2	DHCPv6	145	Solicit XID: 0x37a72e CID: 000100012cd3718000002747b059
[Stream index: 1]						
[Conversation completeness: Complete, WITH_DATA (31)]						
[TCP Segment Len: 256]						
Sequence Number: 151 (relative sequence number)						
Sequence Number (raw): 1968156018						
[Next Sequence Number: 410 (relative sequence number)]						
Acknowledgment Number: 231 (relative ack number)						
Acknowledgment Number (raw): 19339820						
0101 = Header Length: 20 bytes (5)						
Flags: 0x019 (FIN, PSH, ACK)						
Window: 59						
[calculated window size: 64128]						
[Window size scaling factor: 128]						
Checksum: 0xc36 [unverified]						
[Checksum Status: Unverified]						
Urgent Pointer: 0						
[Timestamps]						
[SEQ/ACK analysis]						
TCP payload (256 bytes)						
TCP segment data (256 bytes)						
12 Reassembled TCP Segments (408 bytes): #39(150), #48(250)						
A data segment used in reassembly of a lower-level protocol (tcp.segment_data), 258 bytes						
Frame (312 bytes) Reassembled TCP (408 bytes)						
Packets: 52 - Displayed: 52 (100.0%)						
Profile: Default						

Ora procedo ad analizzare lo scambio di pacchetti per richiedere lo stesso file tramite https:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49210 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000040898	192.168.32.101	192.168.32.101	TCP	66	443 → 49210 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000993918	192.168.32.101	192.168.32.100	TCP	60	49210 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.001017252	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
5	0.001357429	192.168.32.100	192.168.32.101	TCP	54	443 → 49210 [ACK] Seq=1 Ack=102 Win=64128 Len=0
6	0.046144387	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.061308512	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
8	0.061407079	192.168.32.100	192.168.32.101	TCP	54	443 → 49210 [ACK] Seq=1320 Ack=296 Win=64128 Len=0
9	0.062770922	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
10	0.078905891	PcsCompu_cb:7e:f5	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
11	0.259802854	192.168.32.101	192.168.32.100	TCP	60	49210 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0
12	0.911083749	PcsCompu_cb:7e:f5	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
13	2.402936831	PcsCompu_cb:7e:f5	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
14	3.210135298	fe80::18e5:20cf:692::ff02::1:3	ff02::1:3	LLMNR	84	Standard query 0x9eb9 A wpad
15	3.210990996	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x9eb9 A wpad
16	3.304739906	fe80::18e5:20cf:692::ff02::1:3	ff02::1:3	LLMNR	84	Standard query 0x9eb9 A wpad
17	3.306085701	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x9eb9 A wpad
18	3.507671772	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPA0-000
19	4.256293698	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPA0-000
Frame 4: 215 bytes on wire (1720 bits): 215 bytes captured (1720 bits) on interface eth0, id 0						
Ethernet II, Src: PcsCompu_cb:7e:f5 (08:00:27:47:b0:59), Dst: PcsCompu_cb:7e:f5 (08:00:27:47:b0:59)						
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100						
Transmission Control Protocol, Src Port: 49210, Dst Port: 443, Seq: 1, Ack: 1, Len: 161						
Transport Layer Security						
Packets: 78 - Displayed: 78 (100.0%)						
Profile: Default						

Dalla schermata possiamo evincere l'utilizzo della porta 443 (HTTPS) e l'handshake SSL/TLS che includono informazioni sui certificati, le chiavi crittografiche e altri parametri di sicurezza. Inoltre non abbiamo modo di poter visualizzare in chiaro come prima il file sample.txt dato che stiamo utilizzando un protocollo che crittografa i dati durante la trasmissione.