Sicurezza

Goal	**Threat**
Confidenzialità dei dati	Divulgazione
Integrità dei dati	Modifiche accidentali o intenzionali
Disponibilità dei servizi	Denial of service

Gli intrusi: vari gradi di pericolosità

- "Innocenti" ficcanaso
- Script kiddies
- Crackers per diletto
- Assalti a scopo di lucro
- Spionaggio industriale
- Spionaggio militare

Difendere la propria email dal fratellino piccolo *non è la stessa cosa* che difenderla da un intruso competente e determinato

(per un fratellino di pericolosità media...)

Riservatezza

Può essere garantita solo tramite crittografia

Crittografia debole: proteggo i miei file dal mio fratellino (forse)

Crittografia forte: proteggo i miei file dalla CIA

Gli algoritmi crittografici forti sono pochi

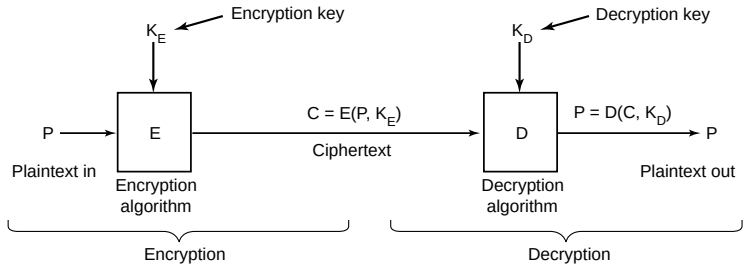
Non fidatevi della crittografia fatta in casa!

Crittografia: che cos'è

Convertire un messaggio *in chiaro* (plaintext) in un messaggio *in cifra* (ciphertext)

Si basa su funzioni matematiche note

Il segreto sta nei parametri degli algoritmi, detti *chiavi*



Un semplice algoritmo di cifratura

Monoalphabetic substitution cipher

Ad ogni lettera faccio corrispondere una lettera

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

Ciao mamma → EOQGDQDDQ

Si risolve facilmente con l'analisi della frequenza delle lettere

(Vedi "Lo scarabeo d'oro" di Edgar Allan Poe)

La macchina da cifra "Enigma"

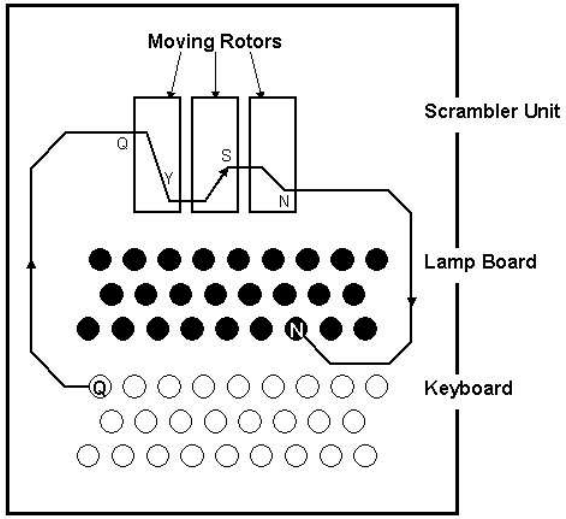
Inventata per il commercio nel 1923

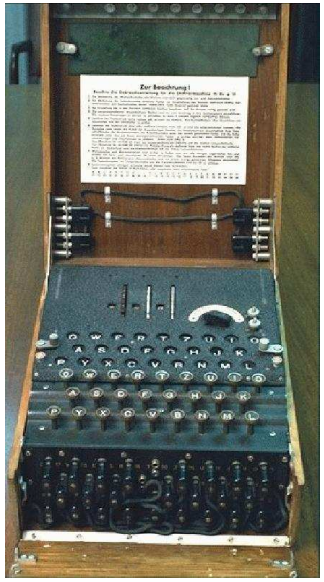
Input: tastiera

Output: display con 26 lampadine

La funzione di sostituzione cambia ad ogni pressione di tasto

Usata in Germania durante la Seconda Guerra Mondiale





Il problema fondamentale della crittografia tradizionale

La chiave per crittare è la stessa usata per decrittare

Per potere comunicare in cifra le due parti devono prima scambiarsi la chiave

Il costo della sicurezza delle chiavi è preponderante

Il problema dello scambio delle chiavi rende impossibile l'uso della crittografia fra sconosciuti
$\Rightarrow$ la crittografia non è di alcuna utilità in Internet

... o no?

Crittografia a chiave pubblica

Una chiave serve a crittare, l'altra per decrittare

Non è possibile ricavare l'una dall'altra

Whitfield Diffie e Martin Hellman, 1976

Basato sulla difficoltà di invertire certe funzioni

Funzioni difficili da invertire

Esempio: "se mi dici quanto fa 1654176541*1987612761 guadagni 100 euro"

un bambino con una calcolatrice può fare questo conto in meno di un ora

Funzioni difficili da invertire

Esempio: "se mi dici quali sono i fattori primi di 3287862401838439701 guadagni 100 euro"

la maggior parte degli adulti non sono in grado di risolvere questo problema, non importa quanto tempo gli lasciamo

Crittografia a chiave pubblica

Bob manda la sua chiave pubblica ad Alice (in chiaro)

Alice critta un messaggio per Bob con la chiave pubblica di Bob

Bob riceve il messaggio crittato e lo decritta con la sua chiave privata

L'attaccante intercetta il messaggio, ma non può decrittarlo perché ha solo la chiave pubblica!

Ma: l'attaccante può intercettare il messaggio, e fare arrivare a Bob un messaggio diverso (man-in-the-middle attack)

Firme digitali

Bob vuole essere certo che i messaggi di Alice siano tali

Alice critta il messaggio M con la sua chiave privata: ottiene $Apr(M)$

Alice manda M e $Apr(M)$ a Bob

Bob decritta $Apr(M)$ con la chiave pubblica di Alice: ottiene $Apu(Apr(M))$

Se $M = Apu(Apr(M))$ allora il messaggio proviene proprio da Alice, e non è stato manipolato!

Ma: siamo certi che la copia di Bob della chiave pubblica di Alice sia autentica?

Un messaggio firmato con PGP

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Un esempio di narrativa cypherpunk è "Cryptonomicon" di Neil Stephenson
-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 6.0.2i

iQA/AwUBOPKqkRLxzDGSvFPxEQI9sQCgi0+MSTQnDNPkQ6ZgFdSn2sJflb4AoINa
ndz5LJmqG3wgT/oATh1pejPR
=M9ra
-----END PGP SIGNATURE-----
```

Web of Trust

Alice vuole certificare che la sua chiave pubblica è veramente sua

Alice chiede a Dave di firmare la chiave pubblica di Alice

Dave usa la chiave privata di Dave per firmare la chiave pubblica di Alice; quindi *Dave certifica che la chiave è autentica*

Bob ha una copia della chiave pubblica di Dave di cui è certo

Bob ha una copia della chiave pubblica di Alice di cui è meno certo

Bob verifica che la firma di Dave sulla sua copia della chiave di Alice è autentica,

Ora Bob si fida un po' di più della sua copia della chiave di Alice

Certification authorities (CA)

Meccanismo opposto al Web of Trust

Le CA Rilasciano *certificati digitali*

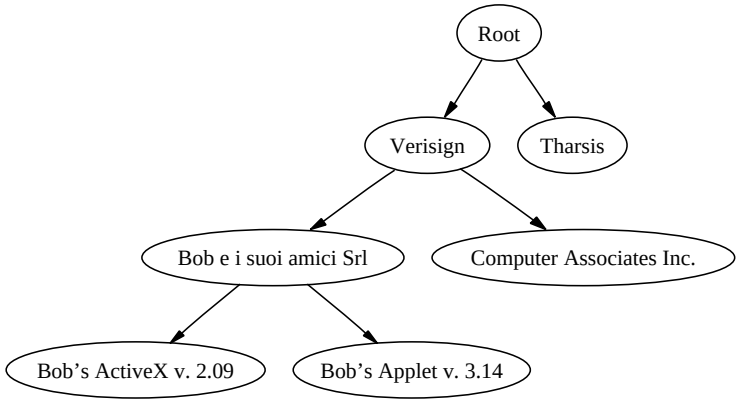
Un certificato digitale si compone di:
- una chiave pubblica
- informazioni, ad es. nome e cognome del proprietario
- una firma digitale

Esiste una *root authority*, e CA intermedie

Standard X.509

Usato per firmare codice (.EXE, .CAB, ActiveX, Applets,. . .), email, e comunicazioni di rete (siti web)

Gerarchia di certification authorities



RSA (Rivest, Shamir, Adelman) è il più usato algoritmo di crittografia a chiave pubblica

Si basa sul fatto che non si conoscono algoritmi efficienti per fattorizzare numeri grandi

siano n, m due numeri primi grandi ($> 10^{100}$)

ti passo $n * m$; prova a scomporlo in fattori primi!

I passi necessari per ottenere un certificato digitale per un sito web

a) generare una coppia di chiavi
b) generare un "certificate signing request" (csr)
c) mandare il csr a una certification authority
d) tirare fuori la carta di credito e pagare
e) copiare il certificato e la chiave privata nei file di configurazione del webserver

Come creare una coppia di chiavi con openssl(1)

```
$ openssl genrsa -out privkey.pem 2048
```

- "genrsa" è il sottocomando per generare una chiave DSA
- "-out privkey.pem": output nel file privkey.pem
- "2048" è la lunghezza della chiave in bit

Il risultato è il file privkey.pem:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAusWBMi7nObsMLAcQ3TIcQPVKn2aWSvsVylS3RgBmWKodyLoW
IdEBVLAvijZO7k2IuAp29v7ITY4oM45YiGgJy7RTUrjPRuWRXNDQvfTlQ3ILR11n
[ ... ]
Suosilu6wy2JwLVRSv7js/Ufy3DxBOzlxlqhoo9U3YouZDRA8ES8B3qZ2pQyPivs
yUM6T+cbkH5e/p30/ZH9RSwO7mWbHjYZTNSjq4DXskFVhU7XCmDRjQ==
-----END RSA PRIVATE KEY-----
```

Questo file contiene sia la chiave privata che quella pubblica

Come generare la "certificate signing request"

```
$ openssl req -new -key privkey.pem -out cert.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IT
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:Varese
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Foo Bar Srl
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:www.foobar.it
Email Address []:vaccari@foobar.it

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:foobarbaz
An optional company name []:
$
```


Come decodificare i campi del file cert.csr

```
$ openssl req -text -nocut < cert.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=IT, L=Varese, O=Foo Bar Srl, CN=www.foobar.it/emailAddress=vaccari@foobar.it
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:ba:c5:81:32:2e:e7:39:bb:0c:2c:07:10:dd:32:
                    1c:40:f5:4a:9f:66:96:4a:fb:15:ca:54:b7:46:00:
[...]
                    36:0d:90:17:29:26:2f:06:94:2a:e1:22:46:72:f8:
                    d0:cd
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :foobarbaz
    Signature Algorithm: md5WithRSAEncryption
        15:e2:b8:b4:41:14:bc:a9:62:90:a5:6b:21:5a:47:f1:12:bd:
        21:d6:ab:b3:d0:a1:c9:e7:65:30:fd:2f:cd:b0:e9:bc:55:e0:
[...]
        40:b9:88:c5:3d:b3:37:1a:ab:ec:bf:0c:2b:f4:20:b3:f5:0f:
        bf:8e:69:d3
```

Come creare un certificato autofirmato

- per test
- per siti web non commerciali
- per mettere su una propria certification authority

```
$ openssl req -new -x509 -key privkey.pem -out cacert.pem -days 1095
```


Come vedere i campi di un certificato

```
$ openssl x509 -text -noout < cacert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=IT, L=Varese, O=Internet Widgits Pty Ltd, CN=www.foobar.it/emailAddress=vaccari@foobar
        Validity
            Not Before: May 31 11:25:39 2004 GMT
            Not After : May 31 11:25:39 2007 GMT
        Subject: C=IT, L=Varese, O=Internet Widgits Pty Ltd, CN=www.foobar.it/emailAddress=vaccari@foobar
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:ba:c5:81:32:2e:e7:39:bb:0c:2c:07:10:dd:32:
[...]
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                8C:9A:38:FF:F7:9D:BB:AE:5E:36:5A:A2:B7:60:1F:5C:6E:04:70:00
            X509v3 Authority Key Identifier:
                keyid:8C:9A:38:FF:F7:9D:BB:AE:5E:36:5A:A2:B7:60:1F:5C:6E:04:70:00
                DirName:/C=IT/L=Varese/O=Internet Widgits Pty Ltd/CN=www.foobar.it/emailAddress=vaccari@
                serial:00

            X509v3 Basic Constraints:
```



```
                CA:TRUE
    Signature Algorithm: md5WithRSAEncryption
        33:78:68:90:08:46:53:b9:bd:98:64:a8:ae:9e:98:cc:e2:df:
[...]
```