Crittografia applicata: PKCS, X.509 e PGP

Luca Veltri

(mail.to: luca.veltri@unipr.it)

Corso di Sicurezza nelle reti di telecomunicazioni, a.a. 2009/2010

http://www.tlc.unipr.it/veltri

Lo standard PKCS

- Public-key Cryptography Standard (PKCS)
- It is based on RSA public-key cryptography
- Developed by RSADSI (RSA Data Security Inc.)
- PKCS is actually a set of standards
- PKCS describes the syntax for messages in an abstract manner, and gives complete details about algorithms
  Defines encoding for
  - **RSA public/private key,**
  - **signature,**
  - **short RSA-encrypted message (typically a secret key),**
  - **etc**

PKCS Goals

- To maintain compatibility with PEM (the Internet Privacy-Enhanced Mail protocols, described in RFCs 1421–1424) wherever possible
- To extend beyond PEM in being able to:
  - **handle arbitrary binary data (not just ASCII data),**
  - **handle a richer set of attributes and features in (extended) certificates and enveloped data**
- To define a standard suitable for incorporation in future OSI standards
  - **The standards are based on the use of OSI standard ASN.1 (Abstract Syntax Notation One) and BER (Basic Encoding Rules) to describe and represent data**

Lo standard PKCS

PKCS: Public-Key Cryptography Standards

- PKCS #1:RSA Cryptography Standard
- PKCS #3:Diffie-Hellman Key Agreement Standard
- PKCS #5:Password-Based Cryptography Standard
- PKCS #6:Extended-Certificate Syntax Standard
- PKCS #7:Cryptographic Message Syntax Standard
- PKCS #8:Private-Key Information Syntax Standard
- PKCS #9:Selected Attribute Types
- PKCS #10:Certification Request Syntax Standard
- PKCS #11:Cryptographic Token Interface Standard
- PKCS #12:Personal Information Exchange Syntax Standard

ASN.1, BER and DER

- ASN.1 (Abstract Syntax Notation One, defined in X.208) is the OSI's method of specifying abstract objects
- ASN.1 is a flexible notation that allows one to define a variety data types
  - **from simple types such as integers and bit strings to structured types such as sets and sequences, as well as complex types defined in terms of others**
- One set of rules for representing such objects as strings of ones and zeros is called the BER (Basic Encoding Rules, defined in X.209)
- BER describes how to represent or encode values of each ASN.1 type as a string of eight-bit octets
- There is generally more than one way to BER-encode a given value
- Another set of rules, called the Distinguished Encoding Rules (DER), which is a subset of BER, gives a unique encoding to each ASN.1 value

Certificati e Certification Authority

Key Distribution

- Public key cryptography solves a major problem with symmetric algorithms
  - **I can encrypt messages to you with your public key**
  - **You can verify my signatures using my public key**
  - **Requires no pre-established relationship between us**
- But how do you get my public key?
  - **And how do you know it is my public key?**
- PGP and X.509/PKI
  - **PGP uses a web of of trust**
  - **X.509/PKI uses hierarchical CAs**

Public Key Infrastructure (PKI)

- System for publishing and verifying the public key values used in public key cryptography
  - ***Certification*** **binds a public-key value to an individual, organization or other entity**
    - Performed by Certification Authorities (CAs)
  - ***Validation*** **is the process of verifying that a certification is still valid**
    - Performed by end users or systems

Certificato digitale

- I certificati sono documenti digitali che attestano la corrispondenza di una chiave pubblica con un individuo/organizzazione
- Nella forma più semplice contengono
  - **una chiave pubblica e il nome dell'individuo/organizzazione,**
  - **il numero di serie del certificato e le date di emissione/scadenza**
  - **il nome di chi ha emesso il certificato (Certification Authority)**
  - **la firma digitale di chi ha emesso il certificato (viene utilizzata la chiave privata del soggetto/organizzazione che emette il certificato)**
- Il formato maggiormente diffuso e accettato è basato sullo standard ITU-T X.509

Where Certificates Are Deployed

- Web transactions
  - **Transport Layer Security (TLS)**
    - Old version called Secure Sockets Layer (SSL)
- Virtual Private Networks
  - **IPSEC using Internet Key Exchange (IKE)**
- Secure messaging
  - **S/MIME, Pretty Good Privacy (PGP)**
- Anywhere strong authentication and/or encryption is required

X.509

X.509 PKI

- PKIX Working Group dell'IETF definisce una PKI come
  - **L'insieme di hardware, software, persone e procedure che sono necessarie per la creazione, la gestione, la memorizzazione, la distribuzione e la revoca di certificati digitali impiegati in un sistema di crittografia a chiave pubblico-privata**
- IETF, RFC 2459: "Internet X.509 Public Key Infrastructure Certificate and CRL Profile"
  - **descrive il formato dei certificati X.509 v3, le estensioni di un certificato, il formato delle CRL, altro**
- IETF, RFC 2527: "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework"
  - **Fornisce indicazioni (framework) per la stesura del Certification Practice Statements (CPS) a cura di una Autorità di certificazione**

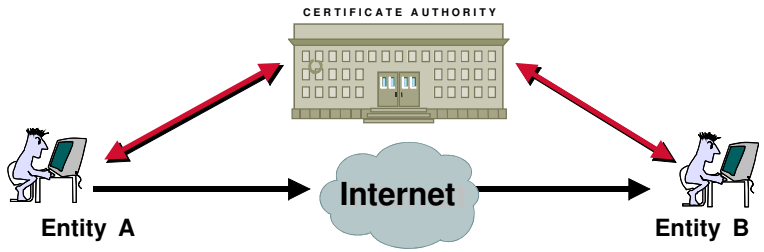
X.509 History

- ITU-T X.509 (formerly CCITT X.509) or ISO/IEC/ITU 9594-8, which was first published in 1988 as part of the X.500 Directory recommendations, defines a standard certificate format
- The certificate format in the 1988 standard is called the version 1 (v1) format
- When X.500 was revised in 1993, two more fields were added, resulting in the version 2 (v2) format
- The Internet Privacy Enhanced Mail (PEM) RFCs, published in 1993, include specifications for a PKI based on X.509 v1 certificates
- The experience gained in attempts to deploy PEM RFCs made it clear that the v1 and v2 certificate formats are deficient in several respects
- ISO/IEC/ITU and ANSI X9 developed the X.509 version 3 (v3) certificate format (June 1996)
- The v3 format extends the v2 format by adding provision for additional extension fields

X.509

- While trust can be assigned to individual keys, the power of the X.509 model comes from its default arrangement of delegating the trust decision to the certification authority
- It does this by assuming trust is inherited from the signing key
- Vendors of X.509 products generally include a set of root certificates that the product will trust "out of the box"
  - **therefore automatically validate other certificates presented to the product**
  - **X.509 encryption and signature capabilities are built into many web browsers and mail programs**
  - **for example the secure HTTP protocol (HTTPS) used for web-based ordering and on-line banking uses X.509**

Certificate Authority

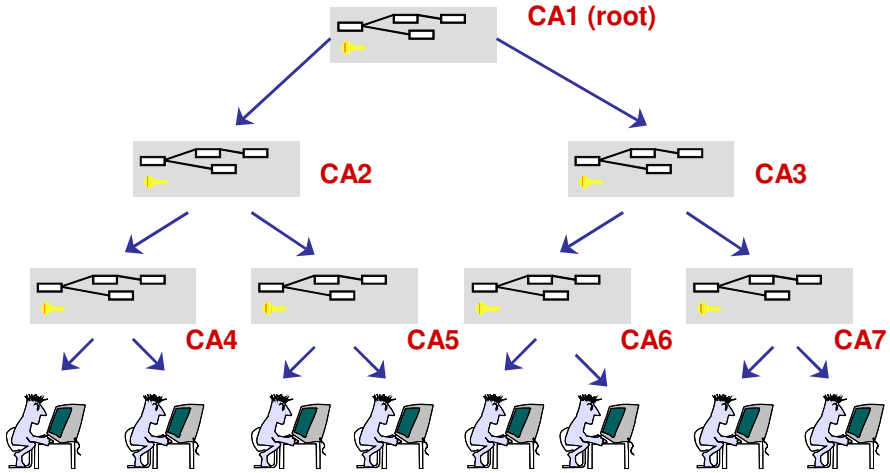


- La Certificate Authority (CA) svolge la funzione di certificare le chiavi pubbliche
  - **garantisce "la connessione" tra le chiavi e l'entità a cui si riferiscono**
- Questa operazione di certificazione avviene attraverso l'emissione di un certificato digitale

Certificate Authority (cont.)

- L'entità certificata può essere:
  - **una persona**
  - **una organizzazione**
  - **un ruolo all'interno di una organizzazione**
  - **uno pseudonimo**
  - **un sistema hardware**
  - **codice software**
- Alcune Autorità garantiscono la certificazione solo ad un sottoinsieme di entità
- Ad una Autorità di certificazione possono essere riposti livelli di fiducia diversi

Certification Authority (cont.)



Types of Certificates

- CA signed certificate
  - **la firma digitale del certificato viene apposta da una Certification Authority**
- Self signed certificate
  - **la firma digitale del certificato viene apposta dal proprietario della chiave pubblica**

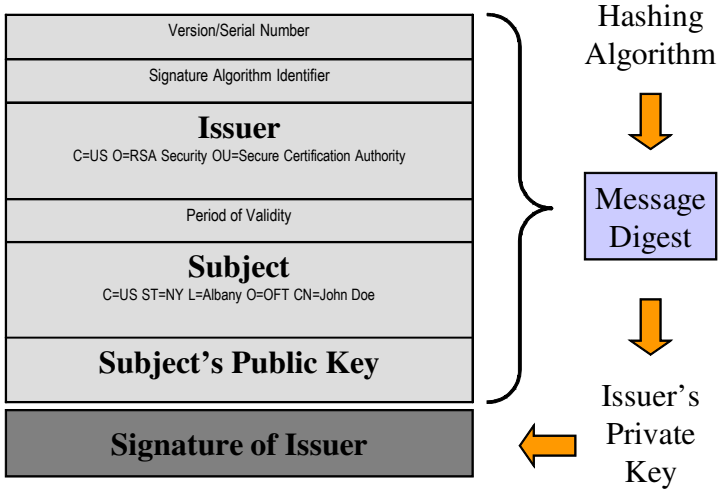
Types of Certificates

- Root Certificates
  - **Self-signed by a Certification Authority**
- CA Certificates
  - **For verifying signatures on issued certificates**
- End systems certificates
  - **e.g.**
    - Server Certificates
      - For use by SSL/TLS servers
    - Software Signing Certificates
      - For signing executable code

Certificato digitale X.509

- Contiene:
  - **una chiave pubblica**
  - **il riferimento (Distinguished Name) ad informazioni che associano la chiave pubblica ad il suo proprietario (nome, e-mail, il nome della Società, telefono) od al dispositivo (Security gateway) che ne fa uso (Indirizzo IP, …)**
  - **il riferimento (Distinguished Name) ad informazioni circa l'organo che ha emesso il certificato (nome, e-mail, telefono)**
  - **un *serial number* che identifica univocamente il certificato**
  - **un indicatore del livello di trust garantito**
  - **la data di emissione del certificato**
  - **la data di scadenza del certificato**
  - **la firma digitale dell'organo che ha emesso il certificato**

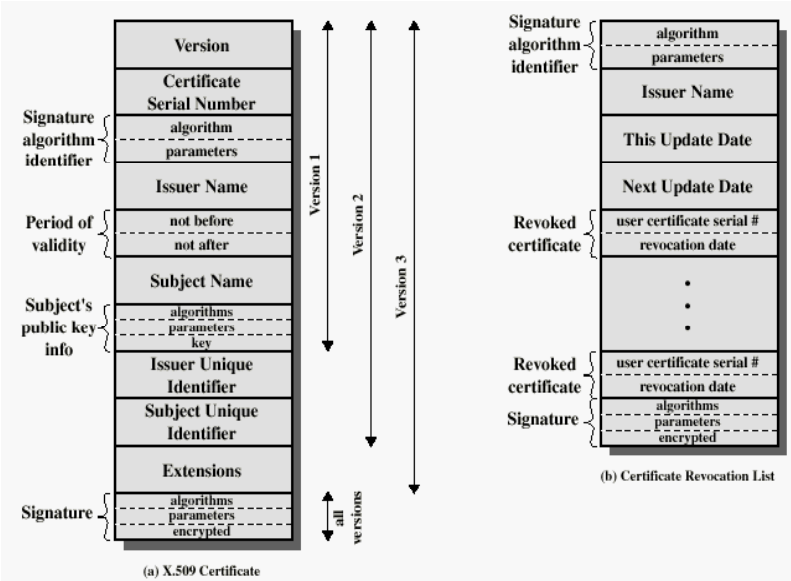
Certificate signature



Example of PEM-encoded certificate

```
-----BEGIN CERTIFICATE-----
MIIC7jCCAlegAwIBAgIBATANBgkqhkiG9w0BAQQFADCBqTELMAkGA1UEBhMCWFkx
FTATBgNVBAgTDFNuYWtlIERlc2VydDETMBEGA1UEBxMKU25ha2UgVG93bjEXMBUG
A1UEChMOU25ha2UgT2lsLCBMdGQxHjAcBgNVBAsTFUNlcnRpZmljYXRlIEF1dGhv
cml0eTEVMBMGA1UEAxMMU25ha2UgT2lsIENBMR4wHAYJKoZIhvcNAQkBFg9jYUBz
bmFrZW9pbC5kb20wHhcNOTgxMDIxMDg1ODM2WhcNOTkxMDIxMDg1ODM2WjCBpzEL
MAkGA1UEBhMCWFkxFTATBgNVBAgTDFNuYWtlIERlc2VydDETMBEGA1UEBxMKU25h
a2UgVG93bjEXMBUGA1UEChMOU25ha2UgT2lsLCBMdGQxFzAVBgNVBAsTDldlYnNl
cnZlciBUZWFtMRkwFwYDVQQDExB3d3cuc25ha2VvaWwuZG9tMR8wHQYJKoZIhvcN
AQkBFhB3d3dAc25ha2VvaWwuZG9tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDH9Ge/s2zcH+da+rPTx/DPRp3xGjHZ4GG6pCmvADIEtBtKBFAcZ64n+Dy7Np8b
vKR+yy5DGQiijsH1D/j8HlGE+q4TZ8OFk7BNBFazHxFbYI4OKMiCxdKzdiflyfaa
lWoANFlAzlSdbxeGVHoT0K+gT5w3UxwZKv2DLbCTzLZyPwIDAQABoyYwJDAPBgNV
HRMECDAGAQH/AgEAMBEGCWCGSAGG+EIBAQQEAwIAQDANBgkqhkiG9w0BAQQFAAOB
gQAZUIHAL4D09oE6Lv2k56Gp38OBDuILvwLg1v1KL8mQR+KFjghCrtpqaztZqcDt
2q2QoyulCgSzHbEGmi0EsdkPfg6mp0penssIFePYNI+/8u9HT4LuKMJX15hxBam7
dUHzICxBVC1lnHyYGjDuAMhe396lYAn8bCld1/L4NMGBCQ==
-----END CERTIFICATE-----
```


X.509 Formats



Certification Paths and Trust

- A user of a security service requiring knowledge of a public key generally needs to obtain and validate the certificate containing the required public key
- If the user does not already hold an assured copy of the public key of the CA that signed the certificate, the CA's name, and related information (such as the validity period or name constraints), then it might need an additional certificate to obtain that public key
- In general, a chain of multiple certificates may be needed, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs

Certificate Revocation

- When a certificate is issued, it is expected to be in use for its entire validity period (certificates have a period of validity)
- However, various circumstances may cause a certificate to become invalid (revoked) prior to the expiration of the validity period, e.g.
  - **change of name, change of association between subject and CA (e.g., an employee terminates employment with an organization)**
  - **user's private key is assumed (or suspected) to be compromised**
  - **user is no longer certified by this CA**
  - **CA's certificate is assumed to be compromised**
- X.509 defines one method of certificate revocation
- This method involves each CA periodically issuing a signed data structure called a certificate revocation list (CRL)
- users should check certs with CA's CRL

Certificate Revocation List (CRL)

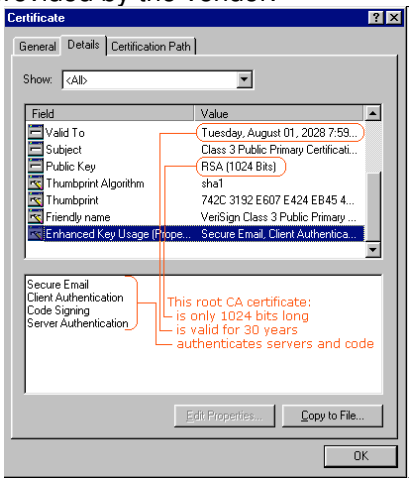
- A CRL is a time stamped list identifying revoked certificates which is signed by a CA and made freely available in a public repository
- When a system uses a certificate, that system not only checks the certificate signature and validity but also acquires a suitably-recent CRL and checks that the certificate serial number is not on that CRL
- A CA issues a new CRL on a regular periodic basis (e.g., hourly, daily, or weekly)
- An entry is added to the CRL as part of the next update following notification of revocation
  - **An entry may be removed from the CRL after appearing on one regularly scheduled CRL issued beyond the revoked certificate's validity period**
- CRLs may be distributed by exactly the same means as certificates themselves, namely, via untrusted communications and server systems
- One limitation of the CRL revocation method, is that the time granularity of revocation is limited to the CRL issue period

How are CAs secure?

- Because they automatically validate other keys, certification authority signing keys are far more valuable than simple e-commerce keys or personal keys
- Compromising one of these trusted certification authority keys would permit an attacker
  - **to fabricate e-commerce certificates, signature certificates and so forth certificates that would be indistinguishable from legitimate certificates and that would be automatically trusted by your browser, e-mail application, or other X.509-enabled application**
- Certification authority key signing keys have long lifetimes (ten to 30 years) and so are generally 2048 bits to protect them during that lifetime

How are CAs secure?

- It is distressing to see a key such as the following in the trusted certification authority key store provided by the vendor:



How are CAs secure?

- The work to factor a 1024 bit RSA key like this is about the same as to brute force a 74 bit key
- The cost to do so would be about 86 million dollars, while the machine to do so in ten years could be built for about 30 million dollars
- So here we have a hundreds of millions of dollars of value protected by a lock that can be broken in a timely fashion (that is, while the key is valid) with a 30 million dollar tool. This is not the way it is supposed to work
- Note:
  - **Internet Explorer 5.5 is distributed with 106 trusted root certificates, of which over half have only 1024 bit keys — and at least one of which has only a 1000 bit key**

Memorizzazione delle chiavi

- Chiave privata: viene mantenuta memorizzata in maniera crittata (algoritmo simmetrico) su hard disk, su smartcard o PMCIA card. L'accesso è protetto da password o passphrase utente
- Chiave pubblica: viene mantenuta all'interno del certificato. I certificati possono essere memorizzati all'interno di un dispositivo fisico (Security Gateway) o all'interno di un repository accessibile

Problems with X.509 PKI: "Which directory?"

- The biggest problem is reflected in the simple phrase "fetches a certificate from a repository" Since the concept of a global distributed directory (or even a less ambitious local directory) was never realized, there 's no clear idea where to fetch a certificate from, and if you have a certificate there 's no clear idea where to fetch its CRL from
- The solution which was adopted, and which works reasonably well in practice, was to include any certificates which might be needed wherever they might be needed
  - **for example, an S/MIME signature usually includes with it all the certificates needed to verify it, and an SSL server 's communication to the client usually includes with it the certificates needed to protect those communications**

Problems with X.509 PKI: "Which John Smith?"

- Even if the user knows which directory to look in, there 's no way to determine which DN should be used to find a certificate, or which of a number of identical names you 're searching on belongs to the person whose key you 're interested in
- PGP solved (?) the problem in a simple manner: users were allowed to choose any kind of identifier they wanted for certificates, which generally consisted of an email address

Problems with X.509 PKI: CRL

- An entity which doesn't have a current CRL is expected to fetch the current one and use that to check the validity of certificates
- In practice this rarely occurs because users and/or applications don't know where to go for a CRL, or it takes so long to fetch
- In order to guarantee timely status updates, it's necessary to issue CRLs as frequently as possible, however the more often a CRL is issued the higher the load on the server which holds the CRL, on the network over which it is transmitted, and on the client which fetches it
- creating and distributing them requires processing time, one or more servers, and significant amounts of network bandwidth (CRLs can become quite large, and many clients can fetch these large CRLs)
- This problem is addressed by protocols such as OCSP (Online Certificate Status Protocol) or the SCVP (Simple Certificate Validation Protocol)

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) - History

- It started out as a single public domain implementation
- Author of PGP is Philip Zimmerman of Guerrilla Freeware
- It was the author's intention that it be distributed widely
- Selected best available crypto algs to use, integrated into a single program
- Both RSADSI (RSA Data Security Inc.) and government authorities caused PGP to start its life as contraband
  - **by enforcing respectively the RSA patent, and export control of dangerous technologies like nuclear weapons and the ability to encrypt mail**
- PGP has been legal and freely available in many other countries because the RSA patent is U.S. only
  - **other governments have different policies about the export, import, and use of privacy protection technology**
- Originally free, now have commercial versions available also
  - **different platforms (Windows, Unix/Linux, Macintoch, etc.)**

GNU Privacy Guard (GnuPG)

- Free implementation of the OpenPGP standard as defined by RFC4880
  - **RFC4880 describes format and methods needed to read, check, generate, and write conforming packets crossing any network**
- Command line tool with features for easy integration with other applications
  - **freely available also frontend applications and libraries**

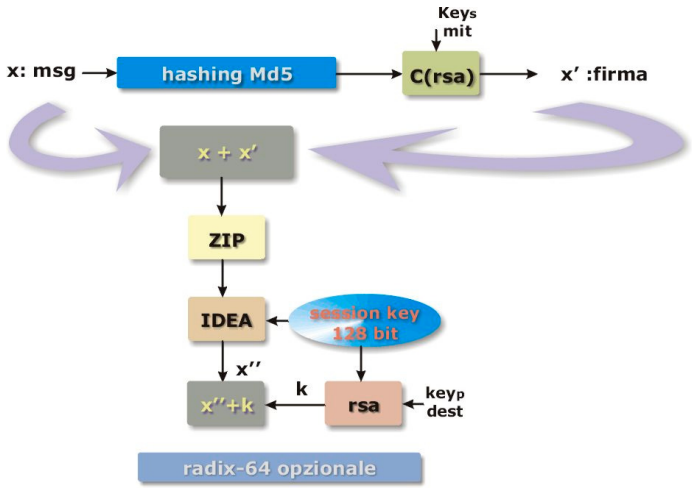
A PGP Message

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.2
hEwDFoFLYIYi6t0BAf0TFjuS2Sp9obCam/Wf1BGsF970sqD0uU9RHNIl4xNny+Sp
U4cwVwDy0HD+CZvdUH+9QAavZwoidlt+AnFqx4uDpgAAAfIxpJ6R06Tou+q6dDTV
ycUXCJKi+DWH/Sfyl+BVo036IFsngt2tE9eoKhZtGzOO0JrC3mlMY/zJFR2FT2kL
uBvGeHSkjC4RlVkoXcbEvIKSyauOBBv4ve1Ze69LVONgBxif8TurmuQpyr0/1+zD
JPC1E9M7o5pckQJnuikg8GFc+mkTwiLgIqau518dPUAjwBZQO4wUI0Kj0DKXMgRQ
wd1CMD5EWLjBXvCsNhTCRC/lzVHX1szUjAjIV8m/hMBDVc+VYvBP8KzjBntwFAIf
ql4HnwqdSRebCDODnEKxScrjtIUX9MpTgjoZ49Tg73R1aUl0p4+ALZDu10X6OdgI
JQBMJEsLl8CIo0eKW1NwKTNJBb3nJWpE7LF3X4k5F/PVobwKMsuLZJTOWy/iMCpW
KZJ63B/A0gbHW1ysJpbbNXDyNkHjyQ/N0TkJTxa4FWnrEAzbjfqyDQLH8KDbEA+g
19Wl7y31dziJYjYsGwPZBGzV0vOLh1lkBNw3wHhf2hQLQ6kL5rK4bASMrISha4oe
aFUD30AdcGzl4RgU2U7eElCwaZPucHSq0EM2RCFZs3Rt4fRwcRlEPySEbI9eK9MJ
MNvXRruxPb7wWA1/UHzhJi6GQkT+G4W1EMeo9euwZsfSYsBGiqcz8IIMbG7PfeYH
T+SXD6A=
=f/zW
-----END PGP MESSAGE-----
```


PGP Overview

- PGP performs authentication, encryption, compression
- Someone wishing to send a secure mail message could first transform the file to be mailed using PGP, and then mail the transformed file using a traditional e-mail program
- Similarly, if one were to receive a PGP encrypted mail message, one could treat the received message as a file and feed it to PGP to process

PGP Overview



The PGP Process

- When you use PGP to encrypt a mail message and send it to a friend, the following occurs:
  - **1. PGP creates a random session key for the message**
  - **2. PGP uses the IDEA algorithm to encrypt the message with the session key**
  - **3. PGP uses the RSA algorithm to encrypt the session key with the recipient's public key**
  - **4. PGP bundles the encrypted message and the session key together and prepares the message for mailing**
- PGP handles session keys automatically, without any intervention on your part

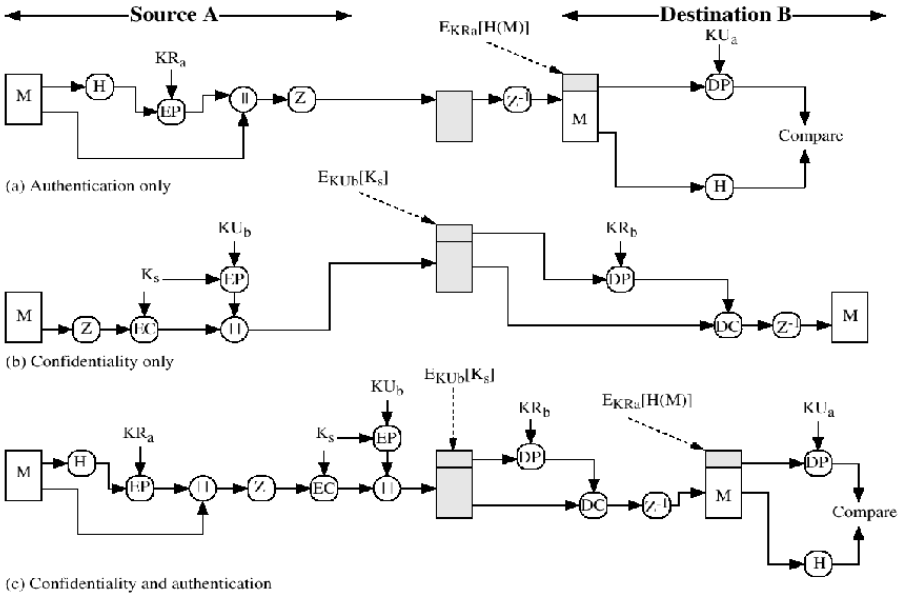
PGP Operations

Consist of five services:

- **Authentication**
- **Confidentiality**
- **Compression**
- **Text compatibility**
- **Segmentation**

PGP: algoritmi utilizzati

- Generatore di numeri casuali
- Algoritmo simmetrico IDEA, 3DES, etc.
- Algoritmo asimmetrico RSA
- Algoritmo per la compressione dati ZIP
- Algoritmo per la selezione del messaggio MD5 o SHA-1
- Algoritmo per la conversione in formato RADIX-64



PGP Operation – Authentication

- sender creates a message
- SHA-1 used to generate 160-bit hash code of message
- hash code is encrypted with RSA using the sender's private key, and result is attached to message
- receiver uses RSA with sender's public key to decrypt and recover hash code
- receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic

PGP Operation – Confidentiality

- sender generates message and random 128-bit (or 168 or 192) number to be used as session key for this message only
- message is encrypted, using CAST-128 / IDEA/3DES with session key
- session key is encrypted using RSA with recipient's public key, then attached to message
- receiver uses RSA with its private key to decrypt and recover session key
- session key is used to decrypt message

PGP Operation – Confidentiality & Authentication

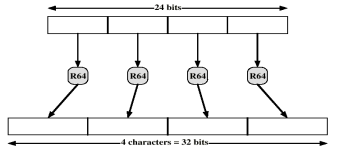
- uses both services on same message
  - **create signature & attach to message**
  - **encrypt both message & signature**
  - **attach RSA encrypted session key**

PGP Operation – Compression

- by default PGP compresses message after signing but before encrypting
  - **so can store uncompressed message & signature for later verification**
  - **& because compression is non deterministic**
- uses ZIP compression algorithm

PGP Operation – Email Compatibility

- when using PGP will have binary data to send (encrypted message etc)
- however email was designed only for text
- hence PGP must encode raw binary data into printable ASCII characters
- the scheme used is radix-64 conversion
  - **maps 3 bytes to 4 printable chars**
  - **radix-64 expands the message by 33%**



- PGP also segments messages if too big
  - **Often restricted to a maximum message length of 50,000 octets**
  - **Longer messages must be broken up into segments**
  - **PGP automatically subdivides a message that is to large**
  - **The receiver strip of all e-mail headers and reassemble the block**

Key Certificates

- PGP keeps each public key in a key certificate. Each key certificate contains:
  - **The public key itself**
  - **One or more user IDs for the key's creator (usually that person's name and e-mail address)**
  - **The date that the key was created**
  - **Optionally, a list of digital signatures on the key, provided by people who attest to the key's accuracy**

PGP Digital Signatures

```
-----BEGIN PGP SIGNED MESSAGE-----
```

message:

```
Really Good Electronics - Chip Prices

1MB 2 CHIP 80 NS        $20.25
1MB 2 CHIP 70 NS        $20.75
1MB 8 CHIPP 80 NS       $18.70
1MB 8 CHIP 70 NS        $19.60
1MB FX (ANY SPEED)      $16.80

For information, call 800-RAM-GOLD
```

```
-----BEGIN PGP SIGNATURE-----
Version: 2.6
```

signature:

```
iQCVAgUBLlqEEHD7CbCQPUJlAQEMXgQAueUPPrpYeb13RZMPD4f8QmW+pQs/ay2P
vrtD+kL0zz3LczxoK3XDdvRj1eRYviXYaJhvStl3cK7+D7lnolmPHWv3DS7tBJzp
G3hJRUr6guRoekcYwXFR7OZhW9VIUHNoIG/OpK23HCatd9f+81TafeUc160k9/CM
Kj034kZlhz8=
=jRLh
-----END PGP SIGNATURE-----
```


PGP Key Management

- rather than relying on certificate authorities
- in PGP every user is own CA
  - **can sign keys for users they know directly**
- forms a “web of trust”
  - **trust keys have signed**
  - **can trust keys others have signed if have a chain of signatures to them**
- key ring includes trust indicators
- users can also revoke their keys

PGP vs. X.509 PKI

- X.509 certificates and PGP certificates (often called PGP keys, for historic reasons) differ in several syntactic ways:
  - **They use different data formats to encode the elements of the certificate**
  - **An X.509 certificate contains exactly one public key, whereas a PGP certificate commonly contains at least two public keys—one for signing and one for encrypting**
  - **An X.509 certificate contains exactly one certification, usually not a self-signed certification**
  - **On the other hand, a PGP certificate contains a collection of certifications, usually at least one self-certification and one third-party certification**
- While PGP and X.509 are syntactically different, they are semantically the same.
- This means that appropriately designed software systems can use either type of certificate in the same processes.