WEB SECURITY

UNIT - I

Outline

- The Web Security Problem
- Risk Analysis and Best Practices
- Cryptography and the Web
    - Cryptography and Web Security
    - Working Cryptographic Systems and Protocols
    - Legal Restrictions on Cryptography
- Digital Identification I: Passwords, Biometrics, and Digital Signatures
    - Physical Identification
    - Using Public Keys for Identification
    - Real-World Public Key Examples
- Digital Identification II: Digital Certificates, CAs, and PKI
    - Understanding Digital Certificates with PGP
    - Certification Authorities: Third-Party Registrars
    - Public Key Infrastructure
    - Open Policy Issues

1. The Web Security Problem

A computer is secure if you can **depend on it and its software to behave as you expect**. The goal of computer security is to **minimize surprise** and ensure **predictable behavior**. Web security is a set of **procedures, practices, and technologies** that assure reliable and predictable operation of:

- web servers
- web browsers
- programs communicating with web servers
- Internet infrastructure

The **scale and complexity of the Web** make web security more complex than Internet security in general.

Today's web security problem has three primary facets

Securing the web server and the data that is on it

- Ensure continued operation of the server
- Prevent unauthorized modification of information
- Distribute information only to authorized individuals

Securing information that travels between the web server and the user

- Protect usernames, passwords, financial information, and browsing data
- Prevent information from being read, modified, or destroyed by third parties
- Protect information flowing from servers to users
- Ensure the communication link cannot be easily disrupted

Securing the end user's computer and other devices

- End user systems must be reasonably secured

- Platforms should be free of viruses and hostile software
- Protect user privacy and personal information

Securing the Web Server

Securing the web server is a **three-part process**:

1. Secure the **computer itself**
2. Secure **programs that provide web service**
3. Examine **interactions between the operating system and web service**

Web servers run on **multi-purpose operating systems** such as Unix or Windows NT, increasing complexity.

Server Security Examples

- Poorly written scripts may allow modification of configuration files and excess privileges
- Secure web servers may still be compromised through insecure database servers with default accounts

Securing the Underlying Computer System

- Examine hardware, operating system, and add-on programs
- Ensure users have **necessary privileges and nothing more**
- Prevent Internet users from breaking in and gaining control

Securing the Computer's Web Service

- Understand web server operation and configuration
- Verify correct privilege and authorization levels
- Examine scripts (CGI, Perl, ASP, VBScript, C) for vulnerabilities

Simplification of services

- Minimize services running on the web server host
- Place different services (mail server, web server) on separate computers
- Remove unnecessary defaults and options
- Reduce complexity to reduce interactions and abuse

Restricting Access to the Web Server

- Locate server in a **secure physical location**
- Limit users who can log into the system
- Use secure remote access methods such as **SSH, SecureID, or S/Key**

Policing copyright

- Prevent unauthorized copying and redistribution of web content
- Technical solutions cannot fully prevent copying once data is viewed
- Copy protection systems can be subverted
- **Digital watermarking** embeds hidden identification in data
- Watermarks identify true owner and original recipient

Securing Information in Transit

- Prevent eavesdropping and alteration of data
- Protection methods:
    - Physical security
    - Information hiding
    - **Encryption** (only practical large-scale solution)

Encryption prevents decoding and detects alteration.

Secure Sockets Layer (SSL)

- Developed by **Netscape Communications**
- Enables encrypted information transfer over the Internet
- Allows secure transmission of credit card numbers
- Credited with enabling electronic commerce
- SSL protects data in transit, not data at endpoints

Denial-of-Service Attacks

- Disruptions caused by physical or logical network events
- February 2000 attacks used repeated web page requests
- No practical individual defense
- Redundancy and backup systems reduce impact
- Legal system needed for deterrence

Securing the User's Computer

- Early web security focused on browser vulnerabilities
- Few losses caused by browser flaws
- Major damage caused by **viruses and worms**, especially via email
- Education alone is insufficient due to system complexity

User Education vs Technology

- Users cannot reliably make security decisions
- Worms like **ILOVEYOU** exploited trust relationships
- Emails appeared to come from known contacts
- Led to file deletion and rapid propagation

2. Risk Analysis and Best Practices

Security is often viewed as a **process designed to prevent something from happening**. People approach computer security by:

- identifying **risks**
- formulating strategies to **minimize or mitigate risks**

Risk Analysis

- A technique involving:
    - gauging the **likelihood of each risk**
    - evaluating the **potential for damage**
    - addressing risks in a **systematic order**
- Has a long history in **public safety and civil engineering**
- Example: **suspension bridge construction**
    - stress analysis
    - probability of failure
    - projected destruction
    - cost-effective design and maintenance decisions

Limitations of Risk Analysis in Computer Security

- Difficult to:
    - gauge likelihood of attacks
    - calculate potential damage

    - estimate long-term security trends
- Few **statistical or scientific studies**
- People **badly estimate risk** based on personal experience
- Questions such as:
    - likelihood of privilege escalation
    - effect of new vulnerabilities over time
    - impact of system maintenance
    - damages of successful penetration remain unanswered

What Is a "Secure Web Server?"

The term **secure web server** means different things to different groups:

Vendor Perspective

- A server implementing **cryptographic protocols**
- Prevents **eavesdropping** during data transfer

User Perspective

- Safeguards **personal information**
- Supports **privacy**
- Does not download **viruses or rogue programs**

Organizational Perspective

- Resistant to **Internet attacks** and **corporate insiders**

Comprehensive View

A secure web server is:

- **Reliable**
- **Mirrored or backed up**
- Quickly **reconstituted after failure**
- **Expandable** to handle large traffic

Cryptographically Enabled Web Server

- Vendors often equate security with **SSL**
- SSL allows secure exchange of information in transit
- Encryption is a **prerequisite for commerce**
- Cryptography:
    - is **useful**
    - is **not sufficient**
    - does **not protect stored data**
- Many attacks stole data **after decryption**
- This book uses the term **cryptographically enabled web server**
- Web security requires **more than cryptographic protection**

Best Practices (Due Care)

- Emerged due to difficulty of risk analysis
- Consists of:
    - recommendations
    - procedures
    - policies
- Generally accepted by **security practitioners**
- Aim:

    - reasonable security
    - risk mitigation
    - reasonable cost
- Act as **"rules of thumb"**

Problems with Best Practices

- No single set applies to all web sites
- Different sites require different security levels
- Following best practices does **not guarantee security**
- Requires:
    - monitoring for new attacks
    - timely patching
- Vulnerable to **novel, unpublished attacks**
- "Best" techniques may not be:
    - appropriate
    - cost-effective
- Many organizations follow only **minimum standards**

Recommended Approach

- Use a **combination of risk analysis and best practices**
- Start with best practices
- Evaluate:
    - risks
    - trade-offs
- Choose **reasonable solutions** for specific environments
- Web servers should:
    - be hosted on **isolated machines**
    - use **minimally required functionality**
- Operators should:
    - remain vigilant
    - apply patches
    - prepare for the unexpected
- Requires:
    - solid understanding of the Web
    - knowledge of failure scenarios

3. Cryptography and Web

3.1. Cryptography and Web Security

Cryptography is the **fundamental technology** used to protect information as it travels over the Internet.

Encryption protects:

- web transactions
- email
- newsgroups

- chat
- web conferencing
- telephone calls

Without encryption, anyone with **physical access to the wires** can eavesdrop. With encryption, messages can be protected such that they are **computationally infeasible to decipher**.

Cryptography is also used to:

- control access to computer systems
- sign digital messages
- enable anonymous digital money
- facilitate online voting

Roles for Cryptography

Security professionals identify **five roles** of cryptography:

Authentication

- Uses **digital signatures**
- Identifies transaction participants or message authors
- Used with or instead of passwords and biometrics

Authorization

- Determines if an authenticated individual is allowed to perform an action
- Uses cryptographic techniques to distribute **authorized user lists**

Confidentiality

- Scrambles data in transit and storage
- Prevents eavesdropping
- Often called "privacy," though professionals distinguish the terms

Integrity

- Verifies messages have not been modified
- Uses **digitally signed message digest codes**

Nonrepudiation

- Uses cryptographic receipts
- Prevents sender from denying message transmission
- True nonrepudiation is **not possible**
- Public key cryptography proves key usage, not user intent

Figures:

Nonrepudiation limitations are discussed conceptually; cryptographic proof of intent is not achievable due to malware, coercion, or misuse.

3.2 Working Cryptographic Systems and Protocols

A **cryptographic system** consists of software and hardware used to encrypt/decrypt data. A **cryptographic protocol** defines how information moves within the system.

Example:

- Web browser + web server using **SSL**

Protocols fall into two categories:

1. **Offline encryption systems**

2. **Online cryptographic protocols**

Offline Encryption Systems

Used for encrypting stored or emailed messages. Examples are listed in **Table 4-1**.

Table 4-1. Cryptographic protocols for offline communications

Protocol	What does it do?	Widely deployed?	Programs and systems	URL
PGP/OpenPGP	Encryption and digital signatures for email and electronic documents	Yes	PGP (Network Associates)	*http://www.pgp.com/*
			Hushmail (Hush Communications)	*http://www.hushmail.com/*
			Veridis	*http://www.veridis.com/*
			Highware	*http://www.highware.com/*
			GNU Privacy Guard (GNU)	*http://www.gnupg.org/*
S/MIME	Encryption and digital signatures for email	No	Netscape Communicator (Netscape Communications)	*http://netscape.com/*
			Outlook (Microsoft)	
			Outlook Express (Microsoft)	*http://microsoft.com/*

PGP/OpenPGP

- **PGP (Pretty Good Privacy)** protects email and files
- **OpenPGP (RFC 2440)** defines standards
- Provides:
  - confidentiality
  - integrity
  - nonrepudiation
- Developed by **Phil Zimmermann**
- Available as:
  - command-line program
  - integrated application
- Supports email clients via plug-ins

PGP keys:

- encryption keys
- signing keys (older versions used a single key)

Each key includes:

- person's name
- mathematical key (**Figure 4-1**)

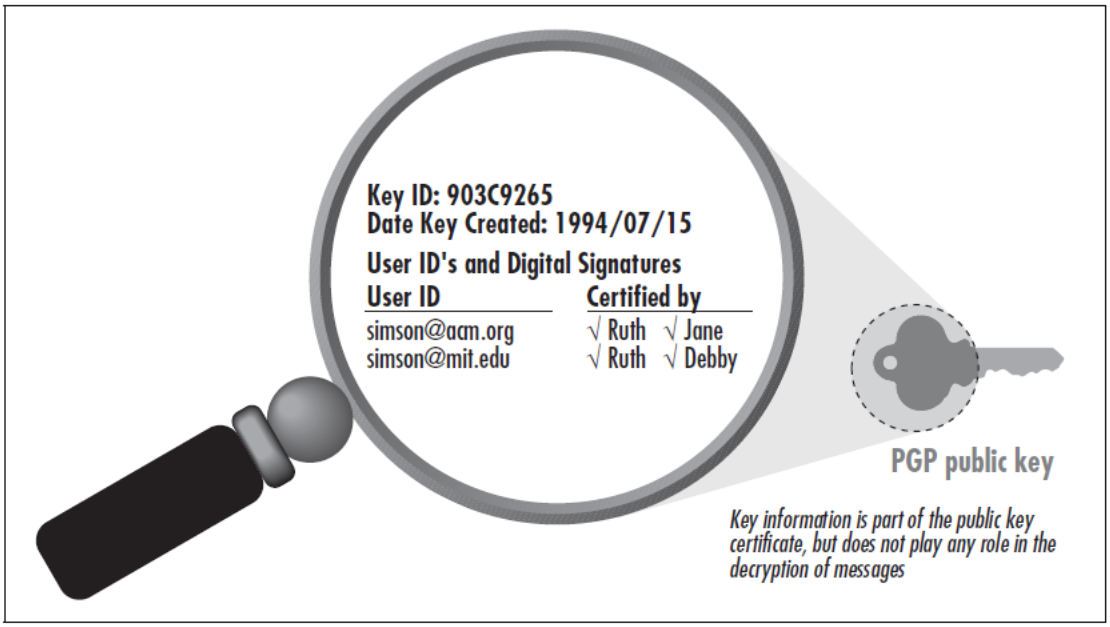

Figure 4-1. PGP keys consist of the actual public key that is used to encrypt or decrypt information, one or more email addresses, and one or more digital signatures attached to each email address.

Key trust is based on:

- direct exchange
- key signing
- certification

Limitations:

- name-based trust
- ambiguity of identities

S/MIME

- Extends **MIME**
- Allows encrypted and signed email
- Requires **S/MIME certificates**
- Certificates issued by **certification authorities**
- More complex than PGP
- Limited adoption

Online Cryptographic Protocols and Systems

Require real-time client–server interaction. Summarized in **Table 4-2**.

Table 4-2. Cryptographic protocols for online communications

Protocol	What does it do?	Widely deployed?	Programs and systems	URL
DNSSEC (Secure DNS)	Provides secure hostname to IP address resolution	No	BIND, Version 9 (Internet Software Consortium)	*http://www.ietf.org/html.charters/dnsext-charter.html*
IPsec and IPv6	Secures IP traffic	No		*http://www.ietf.org/html.charters/ipsec-charter.html*
Kerberos	Provides secure authentication and cryptographic key exchange for higher-level protocols	Somewhat	Kerberos (MIT) Windows 2000 (Microsoft)[a]	*http://web.mit.edu/kerberos/www/*
PCT (Private Communications Technology)	Provides privacy for web transactions	No	Internet Explorer (Microsoft) Internet Information Server (Microsoft)	*http://www.graphcomp.com/info/specs/ms/pct.htm*

SSL

- **Secure Sockets Layer**
- Secures bidirectional TCP/IP communication
- Used by web browsers
- Uses URL prefixes such as:
  - https:
  - snews:
- Provides:
  - confidentiality
  - integrity
  - authentication
  - nonrepudiation
- Uses **X.509 v3 certificates**
- Being replaced by **TLS**
- See **Chapter 5** and **Appendix B**

PCT

- **Private Communications Technology**
- Developed by Microsoft
- Alternative to SSL 2.0
- Declining use
- Still supported on corporate intranets

SET

- **Secure Electronic Transaction**
- Credit card payment protocol
- Merchant never sees credit card number
- Uses:
  - electronic wallet
  - merchant server
  - bank payment server
- Shown schematically in **Chapter 25**

- Limited success due to complexity

DNSSEC

- **Domain Name Service Security**
- Adds security to DNS
- Uses public key infrastructure
- Supports secure updates
- Built into **bind**
- Ideal for remote administration

IPsec and IPv6

- **IPsec** provides packet confidentiality
- Works with IPv4 and IPv6
- Used for **VPNs**
- Authentication and integrity provided by other protocols

Kerberos

- Developed at **MIT**
- Uses symmetric encryption
- Based on shared secrets
- Requires secure Kerberos server
- Supports Telnet, FTP, POP, SSH
- Difficult to administer
- Limited deployment

SSH

- **Secure Shell**
- Provides encrypted terminal and file transfer
- Supports tunneling
- Widely available across platforms

3.3 Legal Restrictions on Cryptography

Cryptography is regulated due to:

- patents
- trade secrets
- export controls
- national security

Cryptography and the Patent System

- Software patents widely accepted
- Early cryptography patents date to 1960s–1970s
- Doctrine of equivalence applies
- Programs can infringe hardware patents

The Public Key Patents

Expired patents include:

- Hellman–Merkle (Expired 1997)
- Diffie–Hellman (Expired 1997)

- RSA (Expired 2000)

Other Patented Algorithms

- IDEA (U.S. patent 5,214,703)
- Patents reduce adoption
- DigiCash affected by patents

Cryptography and Trade Secret Law

- Secrecy does not increase security
- Public review improves algorithms
- Secret algorithms are often leaked
- Examples:
    - RC2
    - RC4
    - DVD CSS (cracked in 1999)

Trade secret protection exists but is unreliable for cryptography.

Regulation of Cryptography by International and National Law

Originally driven by:

- military intelligence
- law enforcement concerns

U.S. Regulatory Efforts and History

- ITAR export restrictions
- 40-bit compromise
- Clipper chip and **key escrow**
- LEAF mechanism (**Figure 4-2**)
- Public opposition (**Figure 4-3**)
- Software key escrow and key recovery failed

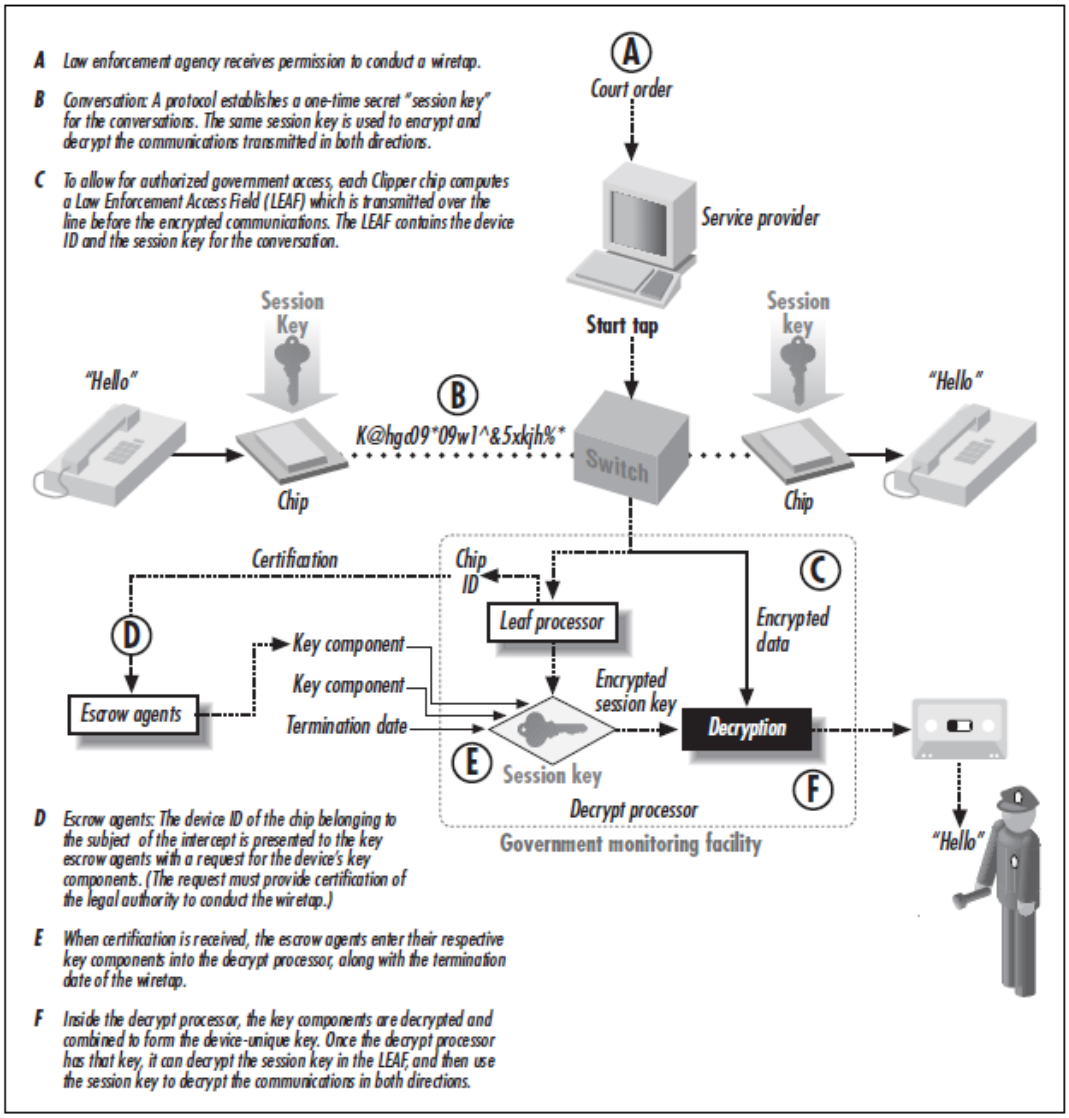

Figure 4-2. A copy of the key used to encrypt the message is included with the encrypted message in a block of data called the Law Enforcement Access Field (LEAF). Each LEAF is encrypted with a key that is unique to the particular Clipper chip. (Adapted with permission from a diagram prepared by SEARCH.)

Figure 4-3. The "big brother inside" campaign attacked the Clinton Administration's Clipper chip proposal with a spoof on the successful "intel inside" marketing campaign.

Export controls eased:

- 1998–2000 reforms
- Wassenaar Arrangement
- Source code export allowed

Outcome:

- Slowed spread
- Hurt U.S. cryptography leadership

The Digital Millennium Copyright Act

- Passed in 1998
- Criminalizes circumvention tools
- Impacts research and free speech
- Examples:
    - arrests
    - publication bans
    - chilled security research
- Under legal challenge (as of 2001)

International Agreements on Cryptography

- **COCOM**
- **Wassenaar Arrangement**
- **Council of Europe**
- **European Union regulations**
- Summarized in **Table 4-3**

Table 4-3. International agreements on cryptography

Agreement	Date	Impact
COCOM (Coordinating Committees for Multilateral Export Controls)	1991–1994	Eased restrictions on cryptography to allow export of mass-market and public-domain cryptographic software.
Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies	1996–present	Allows export of mass-market computer software and public-domain software. Other provisions allow export of all products that use encryption to protect intellectual property (such as copy protection systems).
Council of Europe	1995–present	Recommends that "measures should be considered to minimize the negative effects of the use of cryptography on investigations of criminal offenses, without affecting its legitimate use more than is strictly necessary."
European Union	2000–present	Export to other EU countries is largely unrestricted. Export to other countries may require a Community General Export Authorization (CGEA) or a General National License.

National Regulations of Cryptography Throughout the World

- Import, export, and use restrictions vary
- Summarized in **Table 4-4** (as of March 2001)

Table 4-4. National restrictions on cryptography[a]

Country	Wassenaar signatory?	Import/export restrictions	Domestic use restrictions
Argentina	Yes	None.	None.
Australia	Yes	Export regulated in accordance with Wassenaar. Exemptions for public domain software and personal-use. Approval is also required for software that does not contain cryptography but includes a plug-in interface for adding cryptography.	None.
Austria	Yes	Follows EU regulations and Wassenaar Arrangement.	Laws forbid encrypting international radio transmissions of corporations and organizations.
Bangladesh		None apparent.	None apparent.
Belarus		Import and export requires license.	A license is required for design, production, sale, repair, and operation of cryptography.
Belgium	Yes	Requires license for exporting outside of the Benelux.	None currently, although regulations are under consideration.
Burma		None currently, but export and import may be regulated by the Myanmar Computer Science Development Council.	Use of cryptography may require license.
Brazil		None.	None.
Canada	Yes	Follows pre-December 1998 Wassenaar regulations. Public domain and mass-market software can be freely exported.	None.

4. Digital Identification I: Passwords, Biometrics, and Digital Signatures

4.1. Physical Identification

Physical identification enables trust and accountability in everyday life. For example, a car rental agency can hand over an expensive vehicle based on a driver's license and a credit card. These physical credentials, supported by global computer networks, allow organizations to verify identity quickly and assess risk.

Identification reduces impersonation risk to an acceptable level rather than eliminating it completely. It also allows organizations to:

- Quantify residual risk
- Define policies
- Decide on insurance coverage
- Evaluate alternative identification systems

Historically, identification was based on **personal recognition within communities**, where individuals were known by face, voice, and behavior. As societies grew and interactions extended beyond local communities, **formal identification documents** became necessary.

Identification is closely linked with legal systems. Businesses rely on courts to enforce contracts, but this enforcement is only possible when the true identity of individuals is known. This is why producing false identification documents is a crime.

In online environments, identification becomes more difficult. Physical presence and location no longer provide assurance, making impersonation and fraud easier. Similarly, online businesses struggle to confirm that the person issuing commands is truly the authorized user.

Paper-Based Identification Techniques

Paper-based identification relies on documents issued by trusted authorities, such as:

- Passports
- Driver's licenses
- National identity cards

These documents are widely accepted because:

- Issuance is controlled
- Forgery is difficult
- Identity details are embedded

Verifying identity with physical documents

Verification involves:

1. Inspecting the document for authenticity (materials, seals, laminations)
2. Comparing the physical appearance of the holder to the photograph
3. Checking signatures or asking personal questions

The process is imperfect due to:

- Changes in appearance over time
- Subjective human judgment
- Limited available information

Figure reference:

Figure 6-1 illustrates examples of physical credentials such as a driver's license, passport, or gym membership card, showing how documents act as credentials for proving identity.

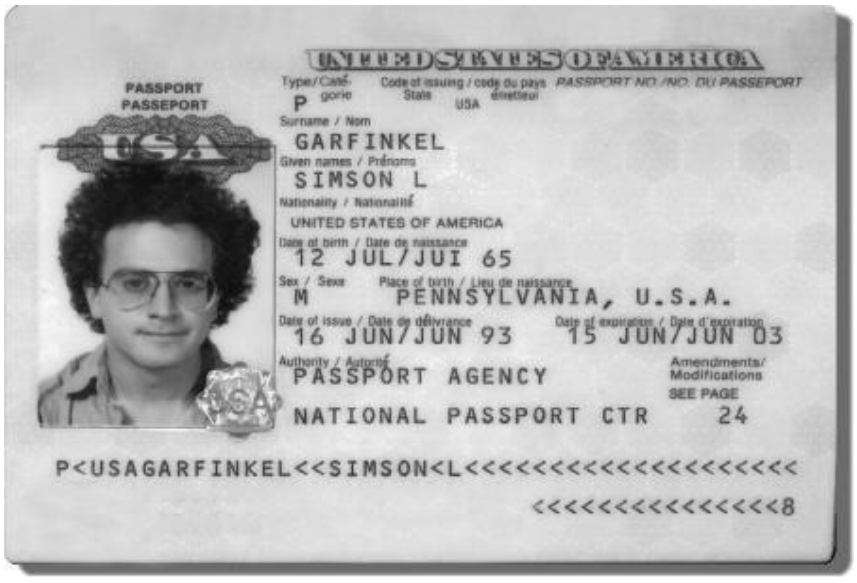

Figure 6-1. A driver's license, passport, or gym membership card is a credential that can be used to prove identification

Reputation of the issuing organization

The effectiveness of paper-based identification depends heavily on the issuing authority. Factors influencing trust include:

- Level of scrutiny during issuance

- Security of blank documents
- Internal accounting controls
- Resistance to bribery
- Likelihood of detecting fraud
- Legal penalties for misuse

A passport issued by a government is trusted far more than a gym membership card because the issuing standards are significantly higher.

An identification document without a reputable issuing authority is essentially worthless.

Tamper-proofing the document

Good identification documents are designed to be:

- **Tamper-resistant** (hard to alter)
- **Forgery-resistant** (hard to replicate)
- **Tamper-evident** (show evidence of alteration)

Methods include:

- Specialized paper and ink
- Laminations
- Ultraviolet (UV) patterns
- Molecular bonding laminates
- Security holograms

Polaroid's UV film and laminates are examples of commercial solutions used to prevent photo replacement and document alteration.

Security holograms, commonly seen on credit cards and software packaging, are difficult to mass-produce without expensive equipment.

No tamper-proof system is perfect. These methods increase the cost and difficulty of forgery rather than eliminating it entirely.

Computer-Based Identification Techniques

Computer-based identification systems have existed for decades and initially focused on:

- Accounting
- Resource usage
- Access control

Unlike paper-based systems, computer systems usually rely on **relative identification** rather than absolute identity. The system only needs to verify that the current user is the same authorized user as before, not their real-world identity.

Computer identification methods fall into four categories:

- Something that you know
- Something that you have
- Something that you are
- Someplace where you are

Password-based systems: something that you know

Password systems use:

- A username
- A secret password

If the entered password matches the stored value, the system assumes the user is legitimate.

Figure reference:

Figure 6-2 depicts a basic password authentication process where a user provides credentials that are compared against stored values.

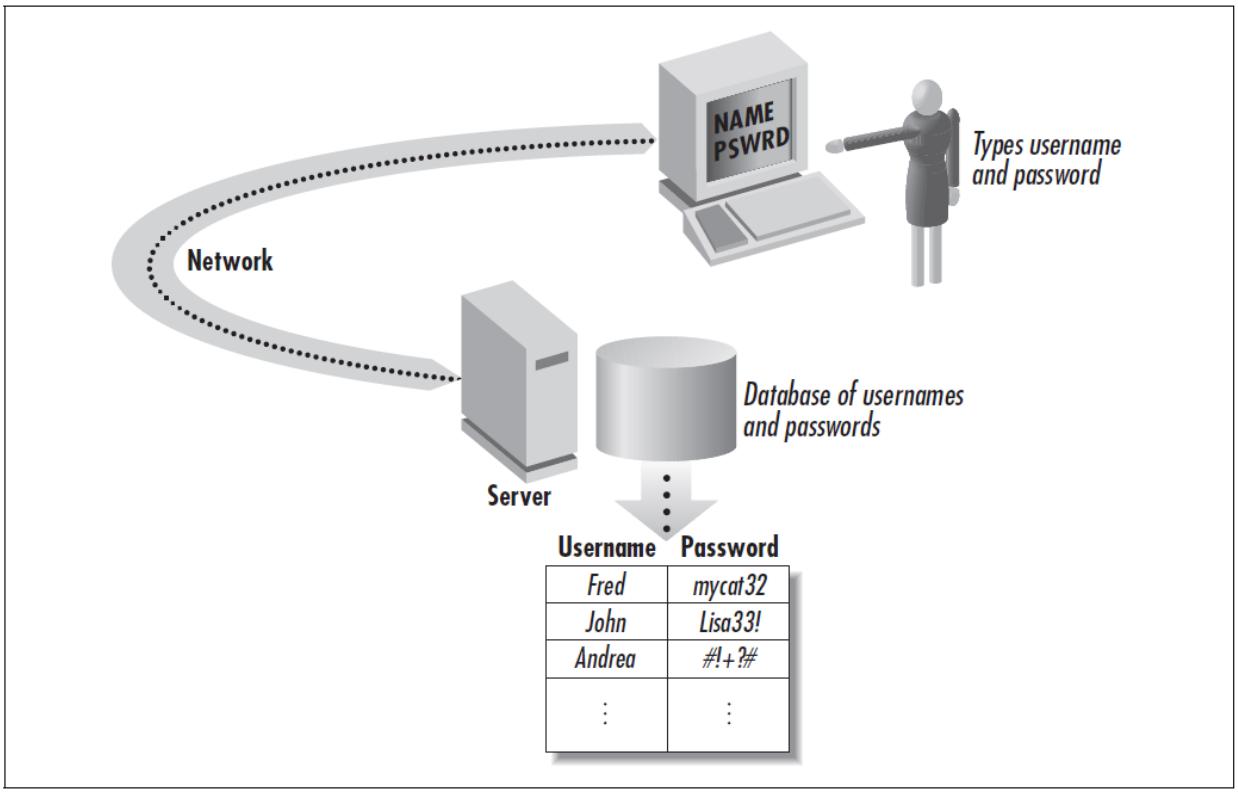

Figure 6-2. Using a username and a password to prove who you are

Advantages

- Simple
- No special hardware required
- Low cost
- Widely supported

Problems

- Easy to forget
- Users reuse or share passwords
- Vulnerable to guessing, interception, and phishing
- Managing many passwords is difficult

Despite these weaknesses, passwords remain the most common authentication mechanism.

Physical tokens: something that you have

Tokens are physical objects that grant access based on possession.

Examples:

- Metal keys
- Magnetic stripe cards
- RFID access cards
- Smart cards
- ATM cards

Figure reference:

Figure 6-3 shows a hardware authentication token (such as the Robocard) used to generate or store authentication information.

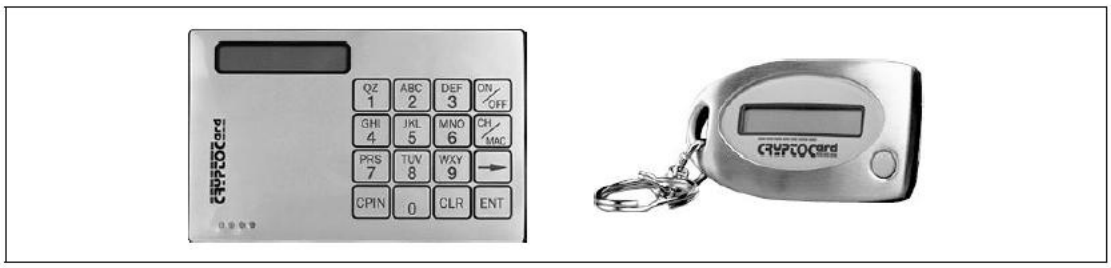

Figure 6-3. Using a token-based system to prove who you are (reprinted with permission)

Advantages

- Can be individually assigned
- Easy to deactivate if lost
- Supports time-based and area-based access control

Problems

- Tokens authenticate possession, not identity
- Can be stolen or copied
- Loss prevents legitimate access

Two-factor authentication

High-security systems combine:

- A token
- A password or PIN

Example: ATM systems require both a card and a PIN.

Biometrics: something that you are

Biometric systems identify users based on physical or behavioral traits.

Examples include:

- Fingerprints
- Iris or retina patterns
- Facial recognition
- Voice prints
- Hand geometry
- DNA
- Typing behavior

Figure reference:

Figure 6-4 illustrates an iris recognition system used for biometric identification.

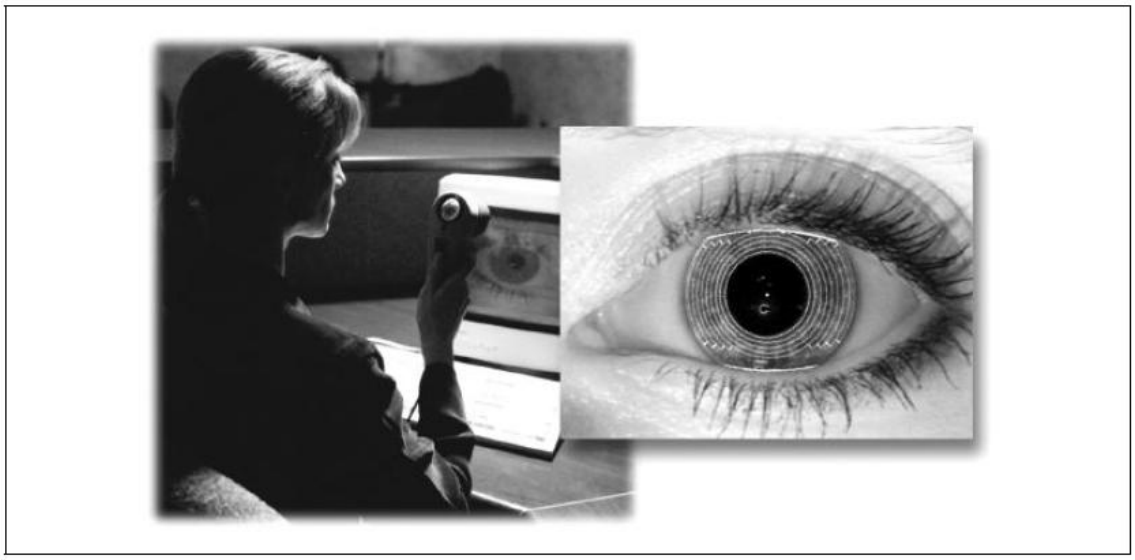

Figure 6-4. Using a biometric-based system to prove identity (reprinted with permission of Iridian)

Types of biometric identification

1. **Ongoing identification** – compares current biometric data to a stored profile
2. **Absolute identification** – compares data against a large biometric database

Advantages

- Cannot be forgotten
- Difficult to share or steal

Limitations

- False positives and false negatives
- Requires prior enrollment
- Database compromise invalidates security
- Sensors can be spoofed or sabotaged

To improve accuracy, biometrics are often combined with passwords or tokens.

Location: someplace where you are

Location-based authentication restricts access based on physical location.

Examples:

- Access limited to certain geographic regions
- Authorized terminals only

Technologies used:

- Mobile network location services
- Fixed-location systems

GPS has limited usefulness due to:

- Poor indoor performance
- Insecure transmission of location data

Location-based authentication is rarely used alone and is typically combined with other methods.

4.2. Using Public Keys for Identification

The identification and authentication techniques discussed earlier in this chapter share a common weakness: they generally require the physical presence of the person being identified.

When identification is performed remotely—such as over the telephone, by fax, or across the Internet—the risk of fraud and abuse increases significantly. This is primarily due to the possibility of **replay attacks**, where authentication data is captured and reused by an attacker. Traditional digital identification methods, including passwords, biometrics, tokens, and even position-based systems, are all vulnerable to replay attacks when authentication information is transmitted across a network.

Replay Attacks

A replay attack occurs when an attacker intercepts authentication data and later reuses that same data to impersonate a legitimate user. This type of attack does not require the attacker to understand or modify the data; simply replaying it is sufficient.

To illustrate this, consider a fingerprint-based authentication system. Under ideal circumstances, a user presses a finger onto a scanner and the system verifies the fingerprint locally. However, if the system is distributed—where one computer captures the fingerprint and another verifies it—then the fingerprint data must travel across a network.

Figure 6-5 depicts such a scenario, showing how fingerprint data is transmitted from one computer to another over a network. An attacker positioned on the network can intercept the digitized fingerprint data. Once captured, this data can be replayed to impersonate the original user.

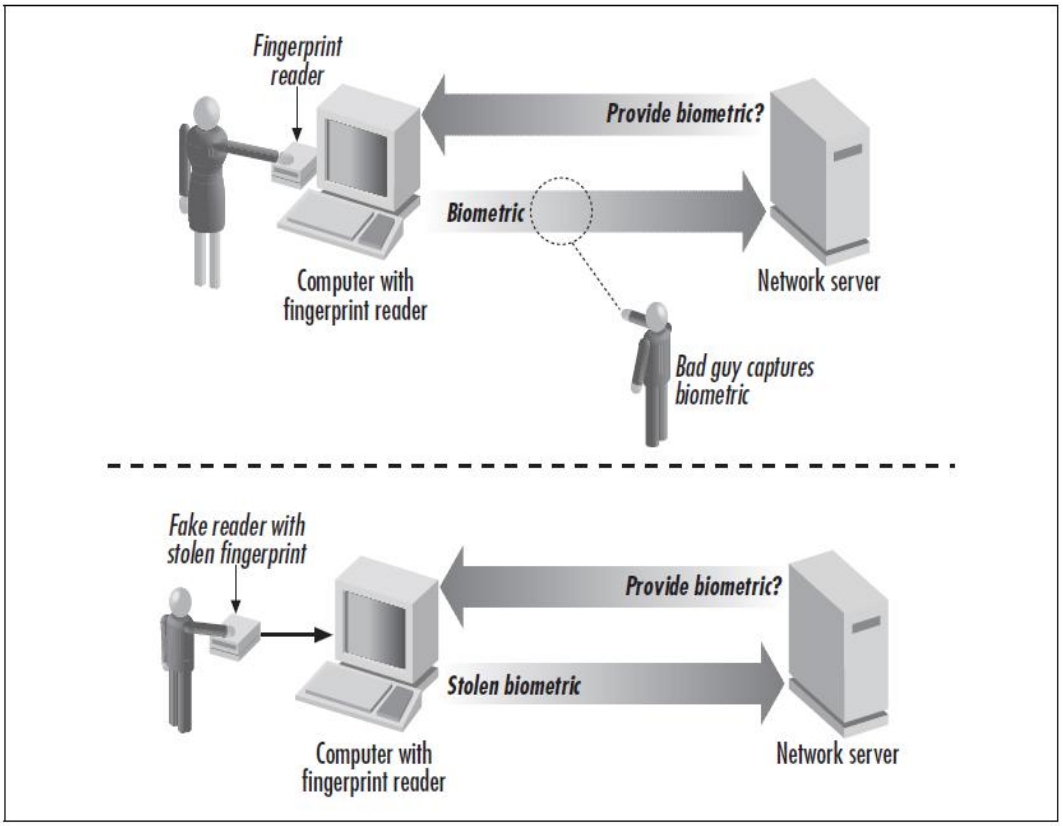

Figure 6-5. When a biometric verification is performed remotely over a computer network, the identification can be compromised by replay attacks (by tampering with the computer or software that measures the biometric).

Replay attacks are not limited to biometric systems. Passwords can be eavesdropped and reused, and even location-based authentication systems can be fooled by replaying previously captured location information.

Encryption and Its Limitations

Simple encryption helps reduce the risk of replay attacks by protecting authentication data while it is in transit. However, encryption alone does not solve the problem completely. If encrypted identification data is ever exposed—through theft, coercion, or system compromise—it becomes permanently compromised.

This limitation is evident in systems that rely on static personal information, such as Social Security numbers or a person's mother's maiden name. Such information cannot be both verifiable and secret at the same time. Once it is revealed, it loses all value as an authentication mechanism.

Stopping Replay Attacks with Public Key Cryptography

Public key cryptography, when properly implemented, can eliminate the risk of replay attacks. Public key systems use two cryptographic keys:

- A **public key**, which is widely distributed
- A **private key**, which is kept secret by its owner

In public key-based identification systems, the private key is used to create a **digital signature**, and the public key is used to verify that signature. Because the private key never leaves the possession of the user and is never transmitted over the network, there is no opportunity for an attacker to intercept and reuse it.

Public key cryptography supports both **offline authentication** and **online authentication**.

Offline authentication

In offline authentication, a digitally signed message is created and verified at a later time.

The process involves the following steps:

1. The user creates a message.
2. The user signs the message using the private key.
3. The message and the digital signature are sent to the remote server.
4. At a later time, the server verifies the signature using the user's public key.

Offline authentication is commonly used for document signing and email verification.

Online authentication

Online authentication involves real-time interaction between the user and a remote server and is generally more secure due to the use of a **challenge-response protocol**.

The steps are:

1. The user's computer connects to a remote server.
2. The server generates and sends a random challenge.
3. The user signs the challenge using the private key.
4. The signed challenge is sent back to the server.
5. The server verifies the signature using the public key.

Because each challenge is random and unique, replaying a previously signed challenge is ineffective.

PGP public keys

Pretty Good Privacy (PGP) is a widely used cryptographic system designed for public key encryption and digital signatures. Although originally developed for secure email, PGP is now

used for signing and encrypting many types of electronic documents. PGP also provides tools for generating, storing, and managing cryptographic keys.

A PGP public key can be thought of as a digital identity card. Early versions of PGP public keys included a person's name and the numerical values used by the RSA algorithm. Modern versions allow additional information, including photographs.

Figure 6-6 shows Simson Garfinkel's PGP public key in text form, while **Figure 6-7** displays the same key graphically using the Windows PGPkeys program. Anyone possessing the corresponding public key can verify digital signatures created with the matching private key.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 6.5.8

mQGiBDPT4LYRBAD/sUgzCctn4KqHoK3ZoK/RKX4x5Lh88PaLdFUWNJiur/HWhNN/
F5yBppqgziFSB6DWZ/Wmrz+NcMjFroiXdtY96eEeRNVW/d4PiooJ+mx5EMoykbB+
YyEhNY7RmTPWDFSmfEZLjVCL17RzUsmXEqBx8LSYGyvS+UArzDsfamPHiQCg/xlE
NTw5r6+2hjpIokYoWFbc8ocEANtEoQOHGx9PG8XHXikpd1PNodkD68ubPz2DlWyO
RNqg6ZY1UtbsSLAhG+fidaJ+bm3+6JaN7F18nBBTnaLYqX8Vyc1NbWbFmr0Cx0Ed
ma4DDp8bxueqHuec1vdEoRqEbsA+2RXU3Qcr9CwhKHRTfg+IV/3Ml4ZOsFlBOZoc
SAFhA/43R2ziDS+sxrLmFY9jvRK1quLfT6kIPCZUKB+tA/VVLG3uHsruXuumRgUS
ZolbD05zvVOY5AP5/SZhT5GRiiNXpaWSDLBKPz/EJVsZ9Pg1QDq9KcrGzZX+ZDAh
ArMC8qIZniHE1mVwOjrTgszOx9khCBGvY0xO7CdEcdaidKPnpLQ1U21tc29uIEwu
IEdhcmZpbmt1bCA8c21tc29uZ0BhY20ub3JnPokATgQQEQIADgUCOodQ+wQLAwEC
AhkBAAoJEPKaG0LR8e7U+OsAoLgjooBAtrnYdVyjFIDED8vMvTPtAJ469yOR+kff
n/lSwV3Uu+xjaqha/rQwU2ltc29uIEwuIEdhcmZpbmt1bCA8c2xnQHdhbGRlbi5j
YW1icmlkZ2UubWEudXM+iQBLBBARAgALBQI6KP+sBAsDAQIACgkQ8pobQtHx7tQp
zQCfauoGugUM6vcnaMUUC5dcATFiDWkAniBbMC32NBWYPh+dBpZiVnjiv3W8uQQN
BDPT4LgQEAD5GKB+WgZhekOQ1dwFbIeG7GHszUUfDtjgo3nGydx6C6zkP+NGlLYw
S1PXfAIWSIC1FeUpmamfB3TT/+OhxZYgTph1uNgN7hBdq7YXHFHYUMoiV0MpvpXo
Vis4eFwL2/hMTdXjqkbM+84X6CqdFGHjhK1P0YOEqHm274+nQ0YIxswdd1ckOEri
xPDojhNn106SE2H22+s1Dhf99pj3yHx5sHIdOHX79sFzxIMRJitDYMPj6NYK/aEo
Jguuqa6zZQ+iAFMBoHzWq6MSHvoPKs4fdIRPyvMX86RA6dfSd7ZCLQI2wSbLaF6d
fJgJCo1+Le3kXXn11JJPmxiO/CqnS3wy9kJXtwh/CBdyorrWqULzBej5UxE5T7bx
brlLOCDaAadWoxTpj0BV89AHxstDqZSt90xkhkn4DIO9ZekX1KHTUPj1WV/cdlJP
PT2N286Z4VeSWc39uK50T8X8dryDxUcwYc58yWb/Ffm7/ZFexwGq01uejaClcjrU
GvC/RgBYK+X0iP1YTknbzSC0neSRBzZrM2w4DUUdD3yIsxx8Wy2O9vPJI8BD8KVb
GI2Ou1WMuF040zT9fBdXQ6MdGGzeMyEstSr/POGxKUAYEY18hKcKctaGxAMZyAcp
esqVDNmWn6vQC1CbAkbTCD1mpF1Bn5x8vY1LIhkmuquiXsNV6z3WFwACAhAAhjai
F3K0JVEIias6jAgLaVYmG4Omk61aI6cNdrgk/J6nqCwoGRJx0vpj6GOHkfHD+d64
b6Q5R6quhzHfRcs2OoCcSamGAK7kg9jtDDJ+zM/q+EH2N9/tLLX8nAG7qMuJN/IK
Jb7e438tnQjOjVaC4hW9Ju945vz1MWcJqeri9DffcnMIvLqC/aV7erJqy/A8aj5O
au29ud7Y9wQcF4XrEC3nRv5PTW4U2xmYRdqTajqjg8qtkTQCpp9SIGUGx4AVbnik
5qLM/awjiIKp+n0LN1VCp2IGsNJKAn0bFuheuQBNTRKfW7KwO6fRC76518rAalyv
/0HkFS/pBe6JcXTVRAGZ81RMqyrpvNjeBKHEsyUlecq5Xra9KIN7cEDjWZyaTU4C
EE1nfFOTQtSbDzydT4dxmgLUG+HMRFE/g+Ax2I71QCLUYEDB/saSXgAkFU180VK9
niUANwdjRL60sZZTTrVQia+QStUIjVo/Ds691Iy0cZ4Zvjt9SFmRAvtPsZ0WfgOx
Df5TNI77nqWwoZHOEhLDMn+Wp+it4CDVJTzw98p7iE2IDXpoJE1suAl4VHdnCBsE
nmR9k7j5FnODBMKOvpp535az1PJwyV0fXQuO32snyr1jb2nBV3dMkG2b4H85NT46
SUVZE/+UIwr6kKGe6rYTTrPUjQVkmq93TOoEmHKJAD8DBRgz0+C48pobQtHx7tQR
AvYTAJ9ATioJ1voy9+jLnQ8rrPDzxmAlnQCffTVnGNmzMxt8hO93MGXBwbf11bw=
=VW4i
-----END PGP PUBLIC KEY BLOCK-----
```

Figure 6-6. Simson Garfinkels's PGP public key, in text form

Figure 6-7. Simson Garfinkel's PGP public key, as shown by the Windows PGPkeys program

Limits of Digital Signatures as Proof of Identity

The ability to generate a valid digital signature does not prove a person's real-world identity. Instead, it proves possession of a specific private key. This distinction explains why public key repositories sometimes contain keys claiming to belong to famous or fictional individuals.

For digital signatures to function as true identity authentication, additional trust mechanisms are required.

Creating and Storing the Private Key

For digital signatures to authenticate identity, the following conditions must be met:

1. Each public/private key pair must be used by only one person.
2. The private key must be kept secure.
3. There must be a trust mechanism to associate a real name with a key.

Poor key generation or insecure storage can allow attackers to derive or steal private keys. In practice, evaluating the security of public key systems is difficult, and implementations vary widely.

Methods for Creating and Storing Private Keys

1. Smart cards (most secure)

The most secure method uses a **cryptographic coprocessor**, such as a smart card. A smart card contains:

- A microprocessor
- A hardware random number generator
- Secure memory for storing keys and certificates

Figure 6-8 shows a smart card used for storing private keys and certificates. The private key never leaves the card; data is sent into the card for signing, and the signed result is sent out.

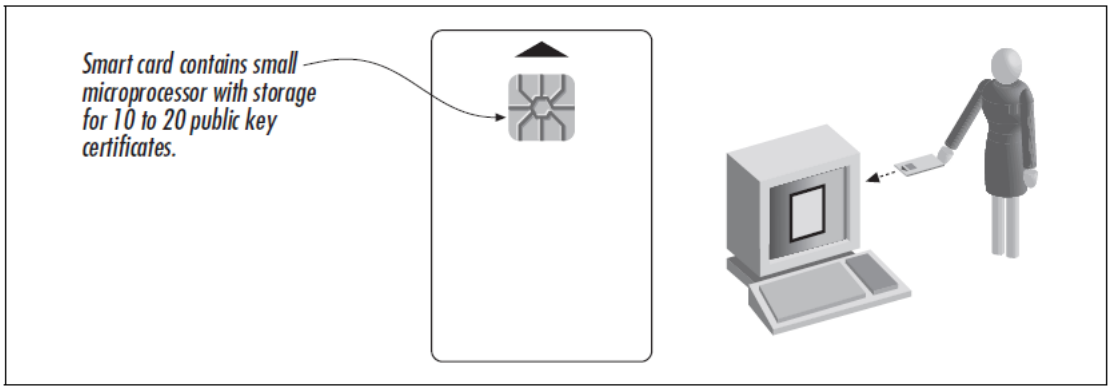

Figure 6-8. Using a smart card to store a private key/public key pair.

Smart cards can be enhanced with PINs or biometric fingerprint readers, ensuring that even possession of the card alone is not sufficient for misuse.

2. Removable media storage

In this method, keys are generated on a desktop computer and stored in encrypted form on removable media such as a floppy disk or flash drive. When needed, the key is loaded into memory, decrypted, and used.

This approach is less secure than smart cards because the private key temporarily exists in computer memory, making it vulnerable to malware and Trojan horses.

3. Local hard disk storage

This is the most common method used by applications such as PGP and web browsers. The key is generated on the computer, encrypted with a passphrase, and stored on the hard disk.

While convenient, this method is vulnerable if an attacker gains access to the computer and knows or guesses the passphrase. The private key is also exposed in memory during use.

4. Third-party key generation (least secure)

In this approach, a third party generates the public/private key pair and distributes it to the user. This method is inherently insecure because the private key is already compromised.

Despite this weakness, some organizations and governments require third-party key generation to allow key escrow and message decryption.

Creating a public key/private key pair with PGP

PGP provides a straightforward process for key creation. Using the Windows version, the user selects the “New Key” option from the PGPkeys application.

Figure 6-9 shows the “New Key” menu option.

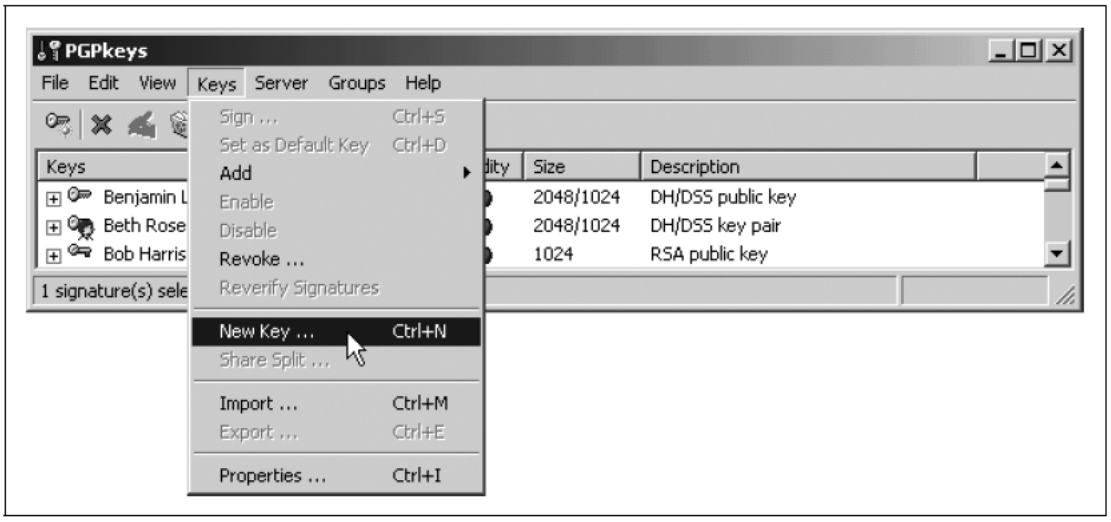

Figure 6-9. To create a new PGP key, select the "New Key..." option from the "Keys" menu of the PGPkeys application program

Figure 6-10 displays the PGP Generation Wizard, which prompts the user for a name and email address.

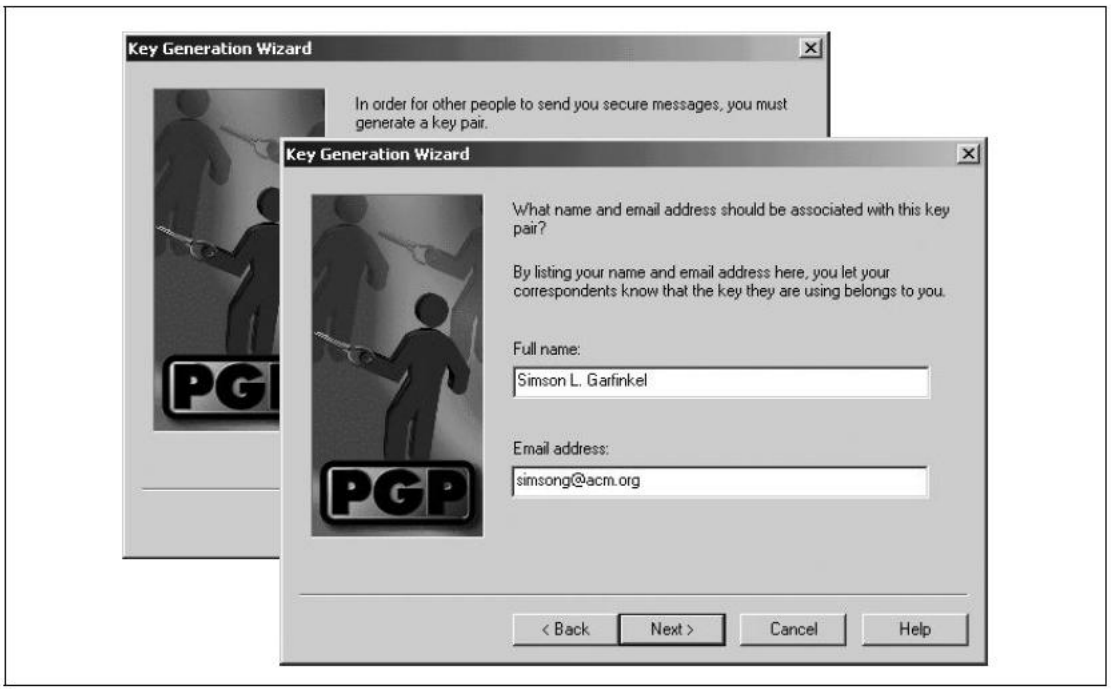

Figure 6-10. When you run the PGP Key Generation Wizard, you will be prompted to enter your full name and email address. This information is recorded on the key. You can change the full name or email address at a later time, but if you do, you will need to have your key reassigned.

PGP supports both RSA and DSA algorithms. The user selects the algorithm and key size, as shown in **Figure 6-11**. Larger key sizes provide greater security but reduce performance.

The user may also choose a key expiration date. Next, PGP prompts the user to enter a passphrase to encrypt the private key. This passphrase is critical for key security.

Figure 6-12 shows the passphrase quality meter, which rates the strength of the passphrase as it is entered. Strong passphrases are long and include letters, numbers, and spaces.

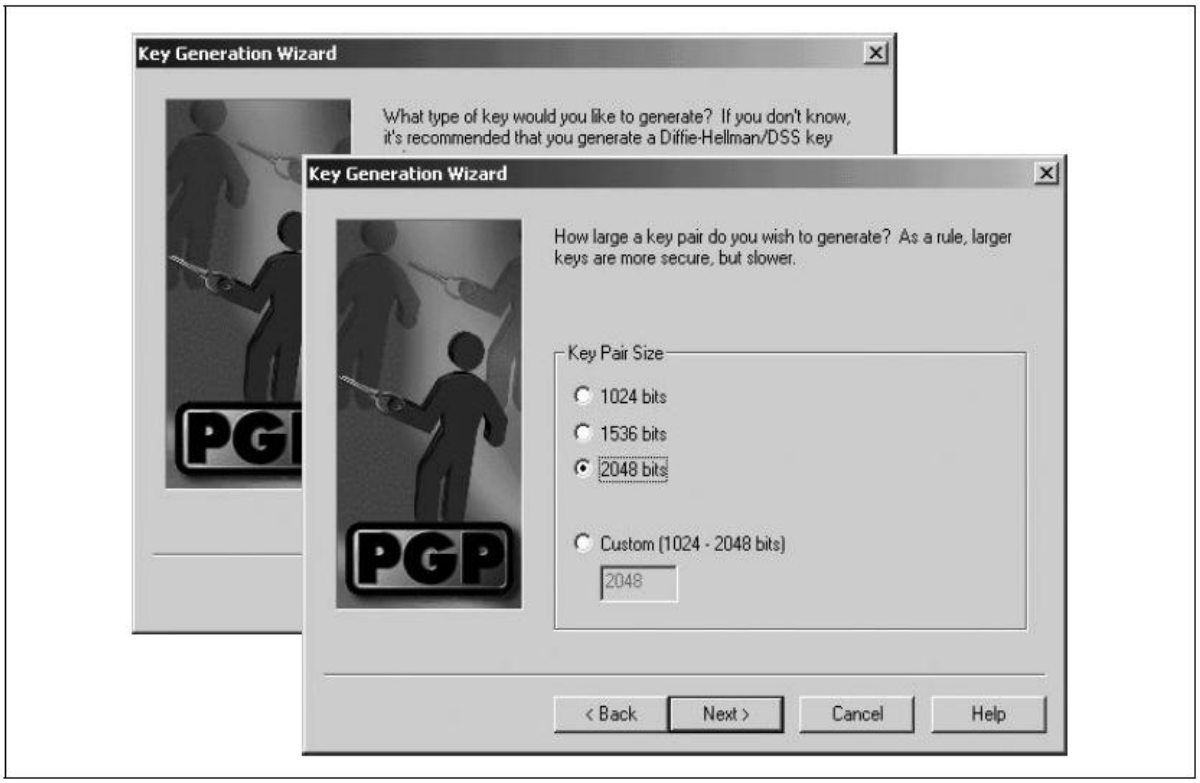

Figure 6-11. After you have given the PGP Key Generation Wizard your name, you will be asked to choose whether you are creating a Diffie-Helman/DSS key or an RSA key. Although PGP recommends that you use a Diffie-Helman key, such keys are not compatible with older versions of PGP. After you choose which algorithm the key will use, you can choose the key's size.

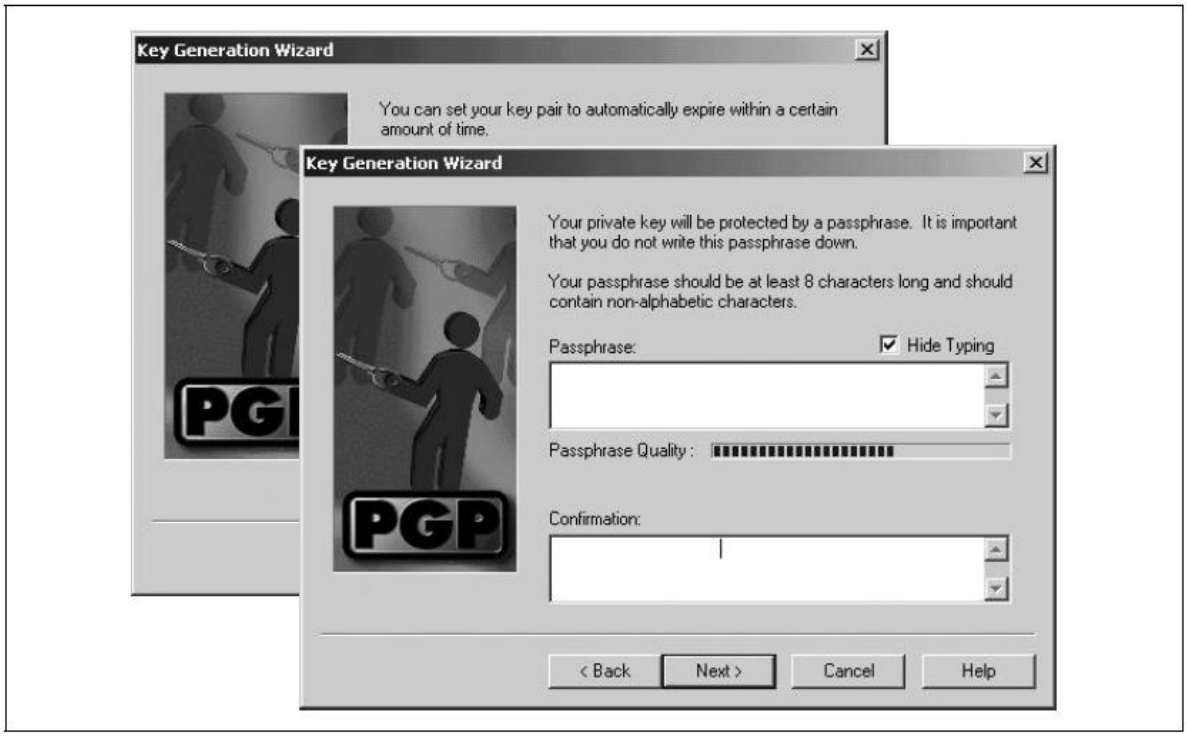

Figure 6-12. The PGP Key Generation Wizard allows you to specify when your key automatically expires. Although this is useful in high-security applications, most users won't use this feature. After you choose an expiration time, you are prompted for a passphrase. PGP shows a passphrase rating.

Figure 6-13. Once all of the parameters of the key have been entered, the PGP Key Generation Wizard creates a key pair.

Finally, PGP generates the key, as shown in **Figure 6-13**. Modern versions of PGP gather randomness automatically and no longer require users to move the mouse or type randomly.

Smart cards

Smart cards provide strong protection for private keys. Removing the card ensures that no one else can access the key. Cards may require a PIN, passphrase, or biometric input and can erase keys after repeated incorrect attempts.

However, smart cards have limitations. If a card is lost, stolen, or damaged, the keys may be permanently lost. For long-term encryption, key backup or escrow may be necessary. For digital signatures, lost keys can simply be replaced.

Smart cards are not completely tamper-proof. Vulnerabilities may exist in their operating systems, and physical attacks are possible. Research by Anderson and Kuhn in 1996 demonstrated successful attacks on commercial smart cards.

More advanced attacks include:

- **Timing attacks**, which analyze operation timing differences
- **Differential Power Analysis (DPA)**, which examines power consumption during cryptographic operations

These attacks demonstrate that even hardware-based key storage is not immune to compromise.

4.3 Real-World Public Key Examples

This section explains how **public key cryptography** is used in real-world systems to authenticate identity. Two major examples are discussed:

- **PGP**, which is an **offline authentication system** used to prove authorship of electronic documents.
- **SSH**, which is an **online authentication system** used for authenticating interactive users and remote systems.

Document Author Identification Using PGP

Email systems do not inherently provide mechanisms to verify the **authorship** or **integrity** of messages. Users can freely modify the "From:" field, and messages may pass through many intermediate systems, making it difficult to confirm whether the message has been altered.

The lack of built-in authorship verification is humorously illustrated by **Peter Steiner's 1993 New Yorker cartoon**, where a dog says, *"On the Internet, nobody knows you're a dog."* This highlights anonymity and impersonation risks.

Despite this, identity and authorship can be authenticated using **Pretty Good Privacy (PGP)**. Although PGP was originally designed for **confidentiality**, it is now widely used for **digitally signing documents**, such as security advisories and source code.

CERT/CC's PGP signatures

The **CERT/CC (Computer Emergency Response Team Coordination Center)** uses PGP to digitally sign its security advisories. These advisories often recommend urgent actions, such as applying patches or reconfiguring services.

Because attackers could send forged advisories, it is essential that recipients verify both:

- **Authenticity** (who sent the message)
- **Integrity** (that the message was not altered)

CERT/CC advisories begin with the line:

----BEGIN PGP SIGNED MESSAGE---

A **PGP digital signature** is appended at the bottom of the message. Unlike handwritten signatures, PGP signatures **cannot be visually verified** and must be validated using cryptographic software.

Obtaining CERT/CC's PGP key

To verify a PGP signature, the verifier must possess the **public key** of the signer.

If the CERT/CC public key is not present, PGP will report that the signature cannot be checked.

There are **two methods** to obtain CERT/CC's public key:

1. Downloading from CERT/CC's web server

- The key can be downloaded directly from CERT/CC's official website.
- **Figure 6-14** shows Internet Explorer prompting the user to open or save the downloaded key file.
- **Figure 6-15** shows PGP displaying key details and allowing the user to import the key into the local key ring.

This method provides higher confidence because the key is obtained directly from the organization.

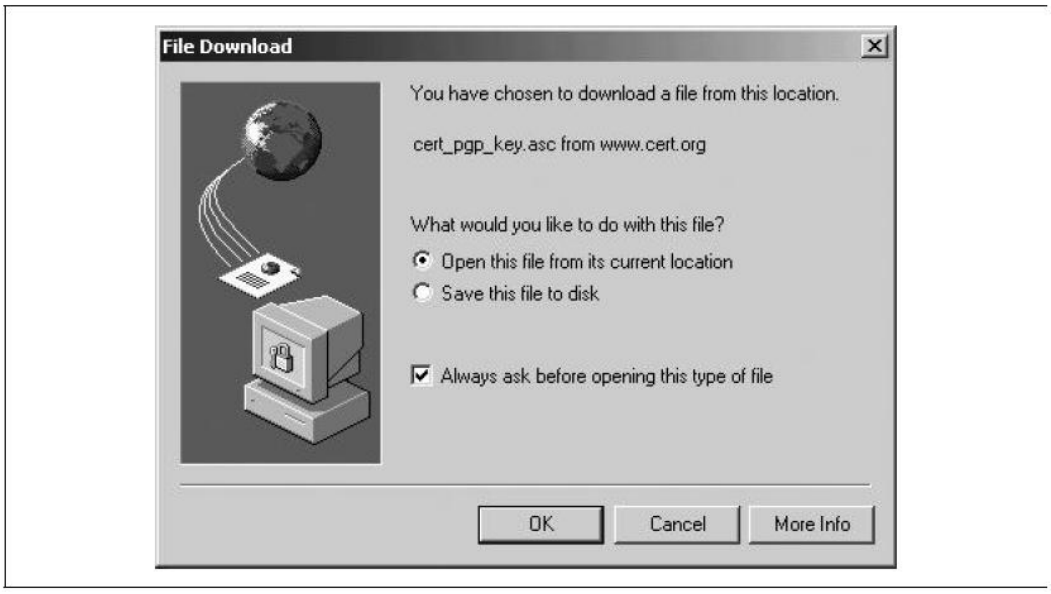

Figure 6-14. When you download the file, Internet Explorer allows you to open the file directly or save it to your disk. Choose "Open" to view the file with the PGPkeys application.

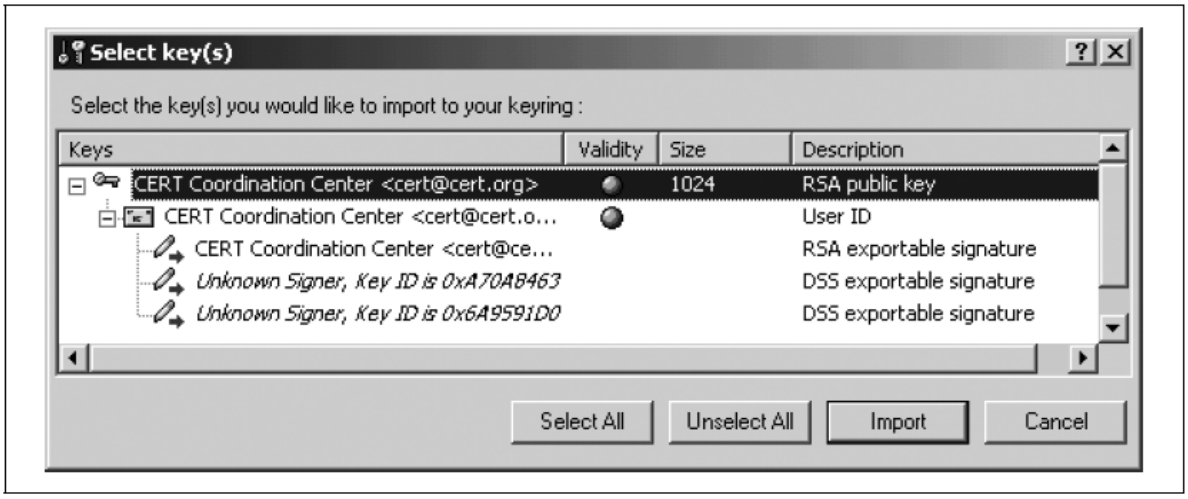

Figure 6-15. The PGPkeys application allows you to view the contents of any key on your hard disk. Once you view the key, you can decide whether or not to "import" the key—that is, to add it to your key chain.

2. Downloading from a PGP key server

- Using the "Search" option from the **Server menu** in PGP.
- **Figure 6-16** shows the key server search option.
- **Figure 6-17** shows searching for the specific **KeyID** that signed the message.
- **Figure 6-18** shows importing the selected key into the local key ring.

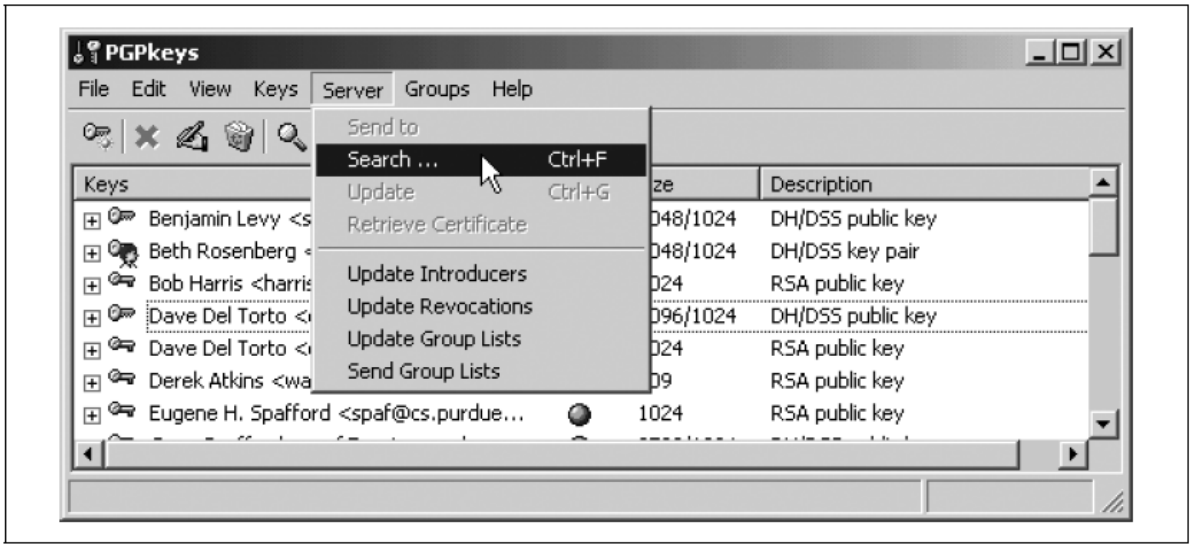

Figure 6-16. To search for a key on the PGP public key server, choose "Search..." from the "Server" menu

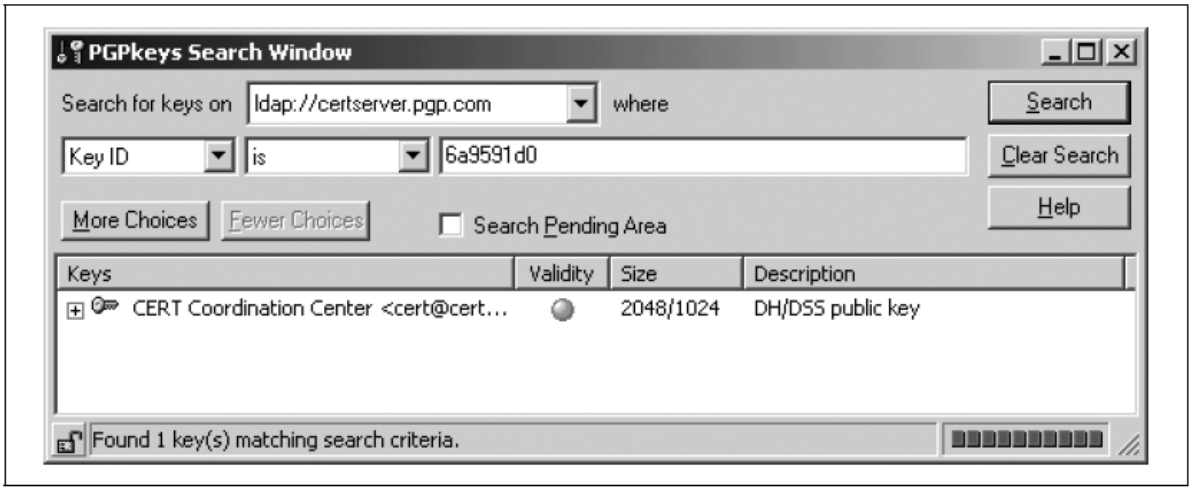

Figure 6-17. Searching for Key0x6A9591D0 on the PGP public key server finds the PGP key for the CERT Coordination Center

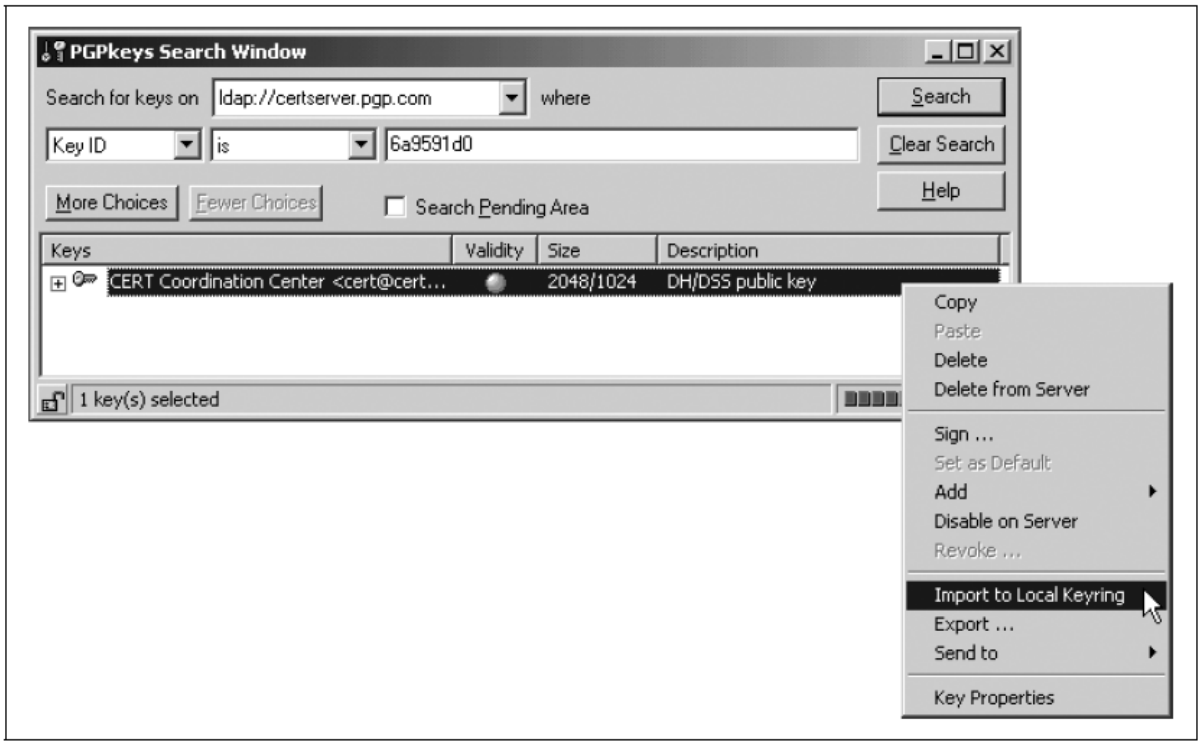

Figure 6-18. After you find a key on the public key server, you can import it by right-clicking on the key and choosing "Import to Local Keyring."

Verifying the PGP-signed file

Once the public key is available, the digital signature can be verified.

- The **command-line version of PGP** displays a message indicating a "Good signature" along with the signer's identity.
- Graphical versions simplify the process:
  - **Figure 6-19** shows right-clicking a file in Windows to verify it.
  - **Figure 6-20** shows PGP verifying the signature.
  - **Figure 6-21** shows the final verification result window indicating whether the signature is valid.

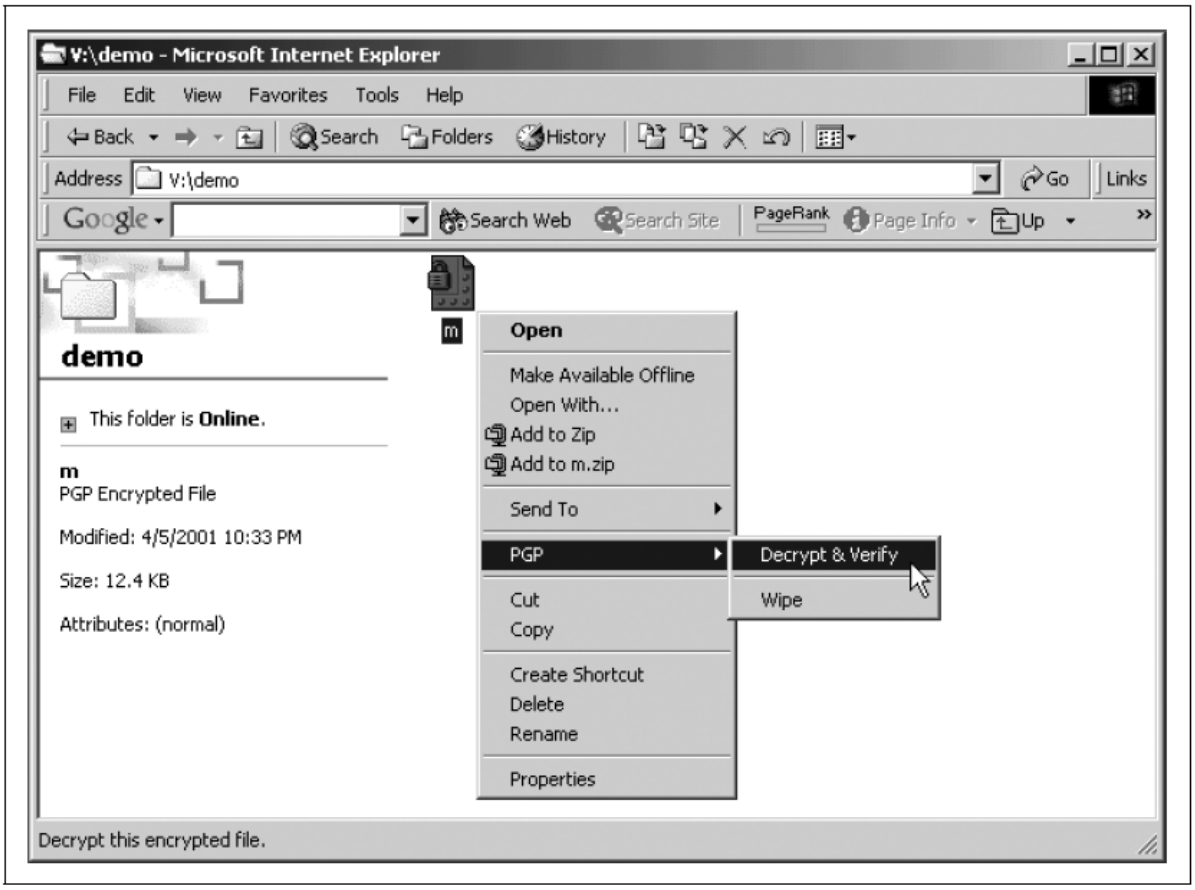

Figure 6-19. You can decrypt a PGP-encrypted file directly from Explorer by right-clicking on the file and choosing "Decrypt & Verify" from the PGP menu.

Figure 6-20. PGP displays the "Decrypting File(s)..." pop-up while it is decrypting a file. You can cancel this operation by clicking on the Cancel button.

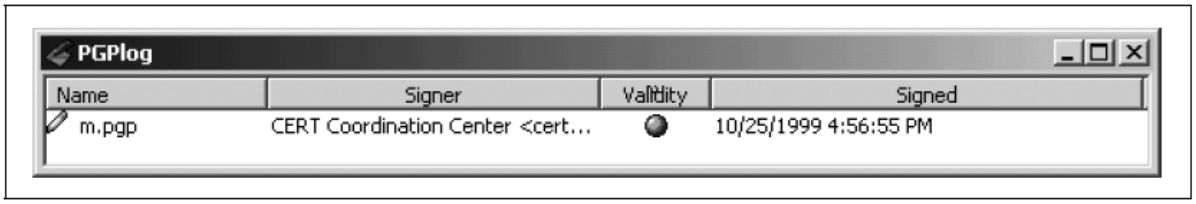

Figure 6-21. The PGPlog window will tell you whether a signature on a file is valid. In this case, the dot under "Validity" is green, and the signature is valid.

This confirms that:

- The message was not altered.
- The signer possessed the corresponding private key.

PGP certification

While PGP signatures verify possession of a private key, they **do not prove real-world identity**. Anyone can create a key with any name.

CERT/CC mitigates this problem by publishing its public key on its official website. Other individuals, such as Gene Spafford, use the same approach.

This limitation motivates the need for **key certification mechanisms**.

Public Key Authentication Using SSH

Secure Shell (SSH) is a secure remote login protocol that uses encryption to protect communications.

Most users authenticate using **password-based SSH**, where the password is encrypted during transmission.

SSH also supports **public key authentication using RSA**, which eliminates the need to transmit passwords.

RSA authentication in SSH

To use RSA authentication:

1. A user generates an RSA key pair using **ssh-keygen**.
2. Two files are created:
    - identity.pub – public key
    - identity – private key (optionally encrypted)

The public key is copied to the remote system's .ssh/authorized_keys file.

Once registered, the user can log in **without entering a password**.

SSH authentication process

The verbose (-v) SSH output shows the **cryptographic steps** of authentication.

Key steps include:

- The server sends a **random challenge (nonce)**.
- The client signs the challenge using its **private key**.
- The server verifies the signature using the **stored public key**.

Figure 6-22 schematically illustrates this **public key challenge-response protocol**.

This method prevents replay attacks because:

- The private key is never transmitted.
- The challenge is randomly generated.

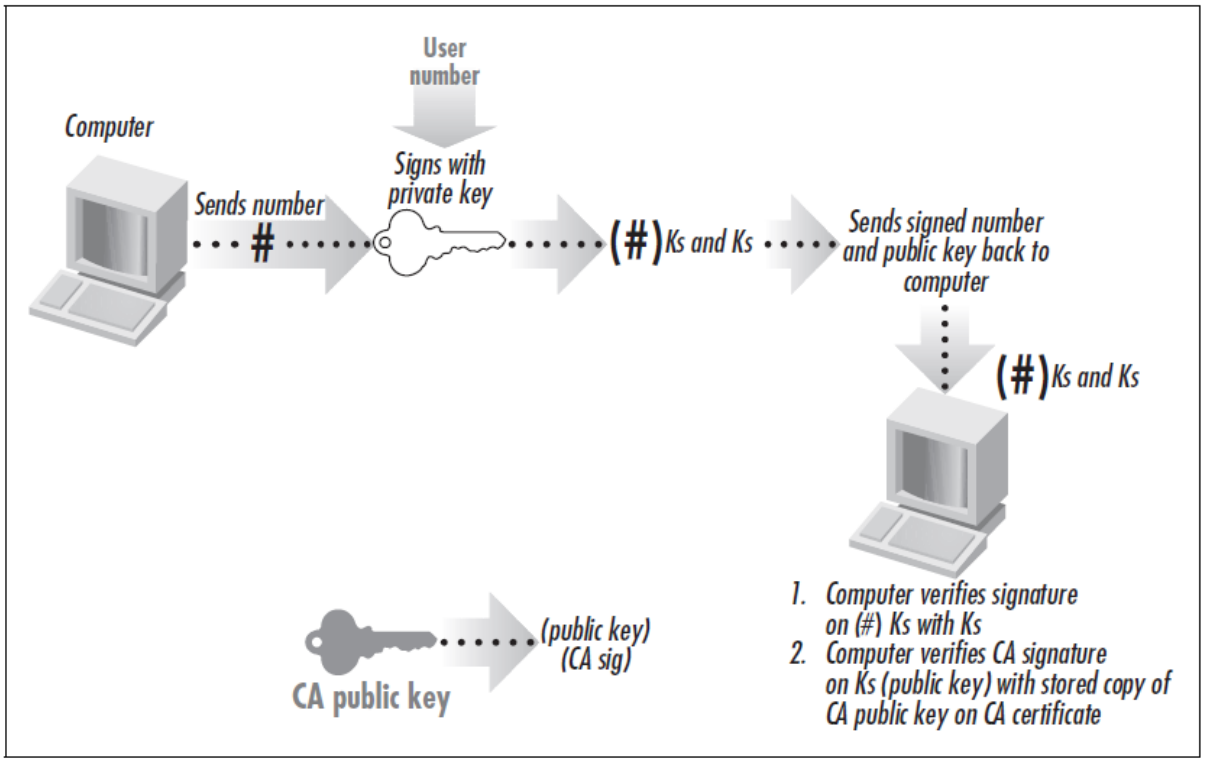

Figure 6-22. In a public key challenge-response process, a computer can provide a person with a number (the challenge) which must be signed. If the person can sign the number and return it to the computer, and if the digital signature can then be verified by the computer using the public key on file for the individual, the person must possess the private key that matches the given public key.

Security considerations in SSH

SSH authentication can be compromised if:

- An attacker gains access to the machine storing the private key.
- The private key is not encrypted with a passphrase.

Another theoretical risk exists if a nonce is reused, which is why timestamps are often included.

5. Digital Identification II: Digital Certificates, CAs, and PKI

This section introduces **digital certificates** and their role in large-scale identity systems.

5.1. Understanding Digital Certificates with PGP

A **digital certificate** is a signed block of data that binds a **public key** to identity information.

Figure 7-1 shows a conceptual digital certificate containing:

- Public key
- Identity attributes
- Signature of a certifying authority

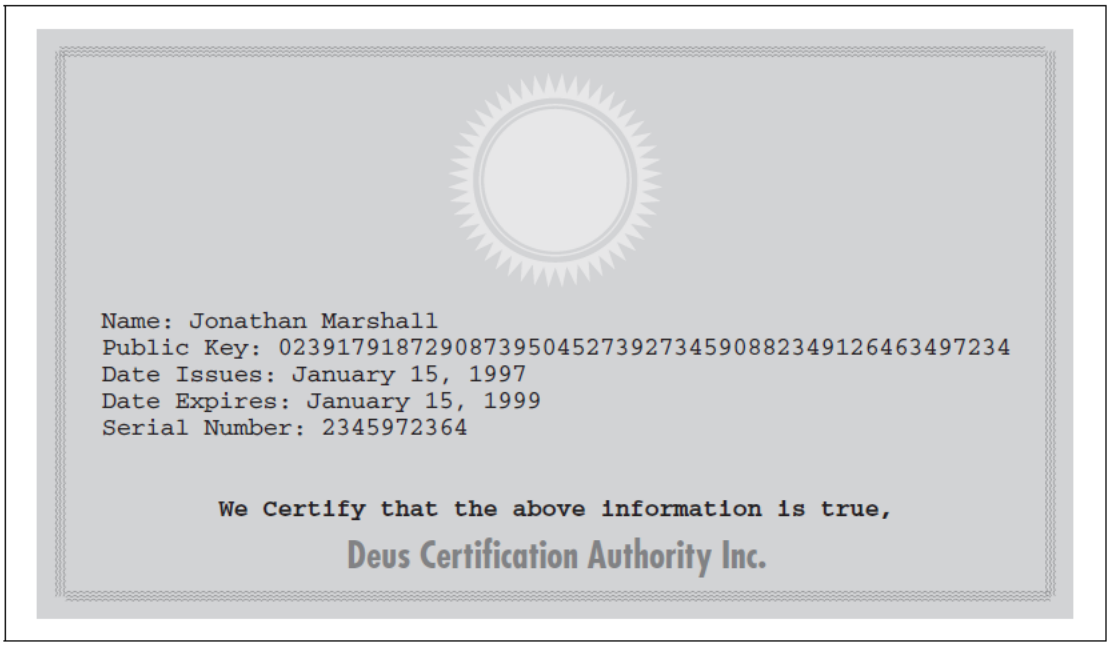

Figure 7-1. A digital certificate consists of a public key, additional information such as a person's name or affiliation, and a digital signature from a certification authority (CA).

PGP public keys are actually **digital certificates** that include:

- Name
- Email address
- Self-signatures
- Signatures from others

Certifying Your Own Key

PGP allows users to:

- Create their own keys
- Certify their own identity information

This freedom led to widespread adoption but also resulted in many **fraudulent keys**.

Figure 7-2 shows fake keys claiming to belong to famous individuals.

Figure 7-3 shows CERT/CC warning about a fraudulent CERT key.

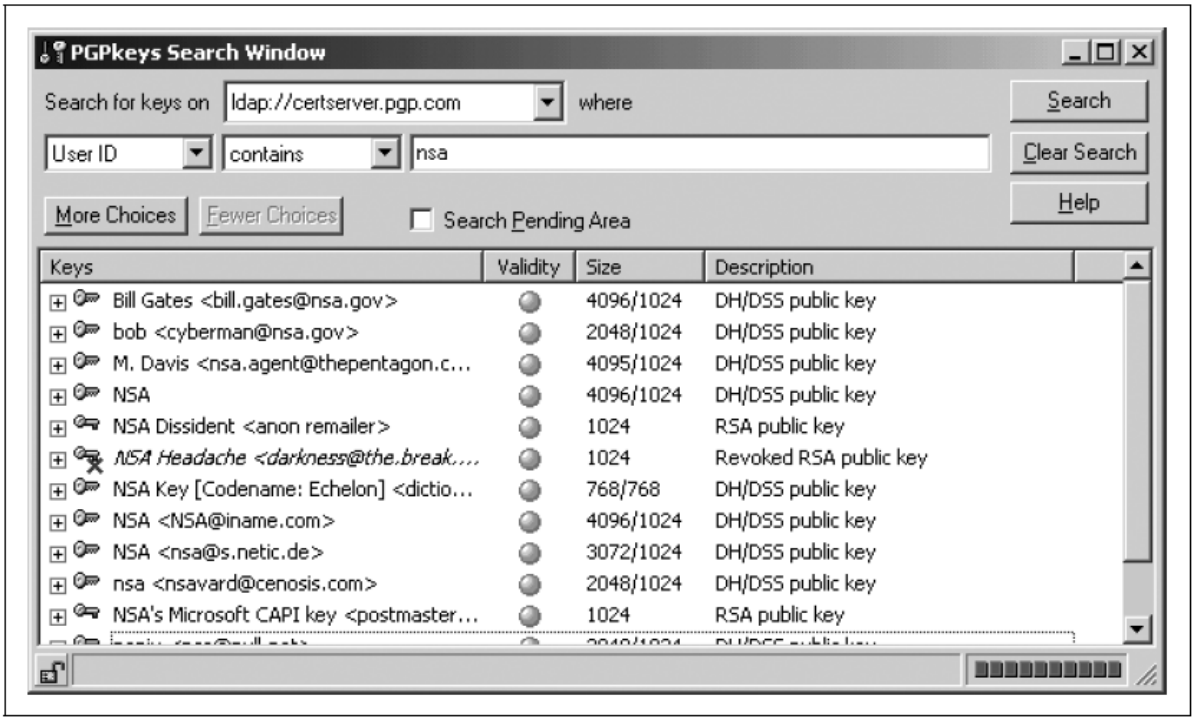

Figure 7-2. Many keys put on the PGP key server don't really belong to the person whose name is listed on them

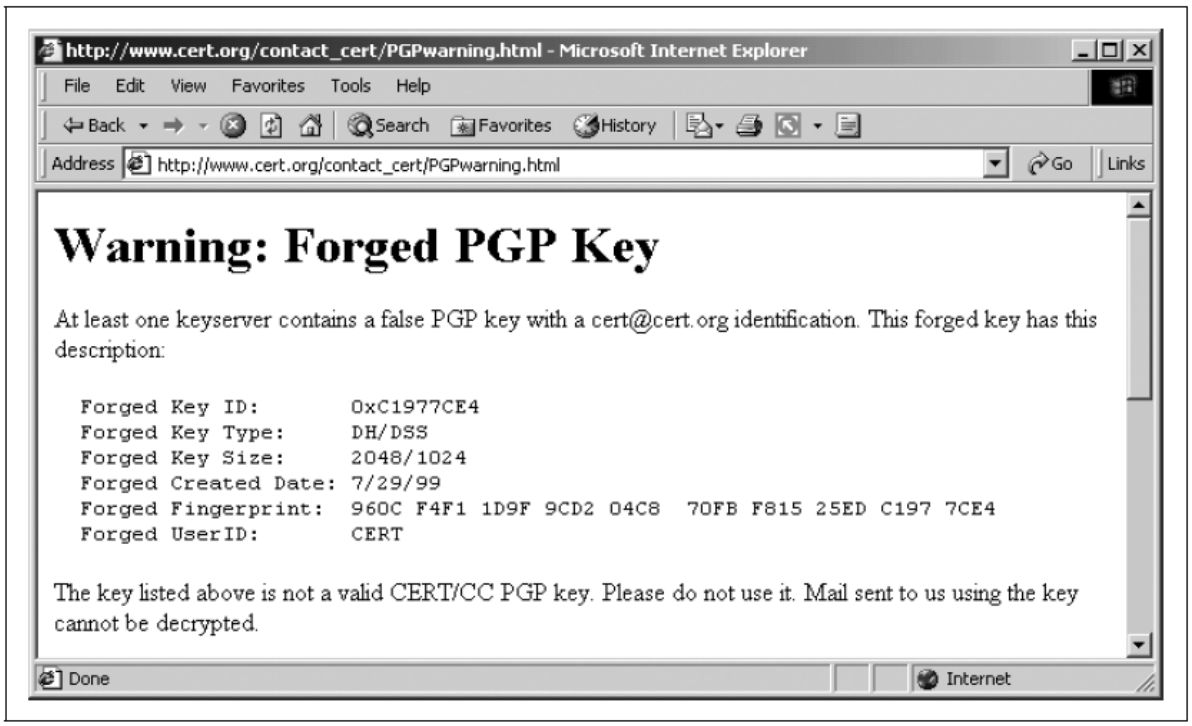

Figure 7-3. CERT/CC issued a warning about a fraudulent PGP key with CERT/CC's name that was put on the PGP key server.

Certifying Other People's Keys: PGP's "Web of Trust"

PGP models trust socially, similar to how people trust others through acquaintances.

Users can **sign other users' keys**, asserting that the key belongs to that person.

Trust and validity

PGP distinguishes between:

- **Validity**: Whether the key truly belongs to the claimed identity.
- **Trust**: How much confidence you have in the key holder to certify others.

Figure 7-4 shows a PGP key ring displaying varying levels of trust and validity.

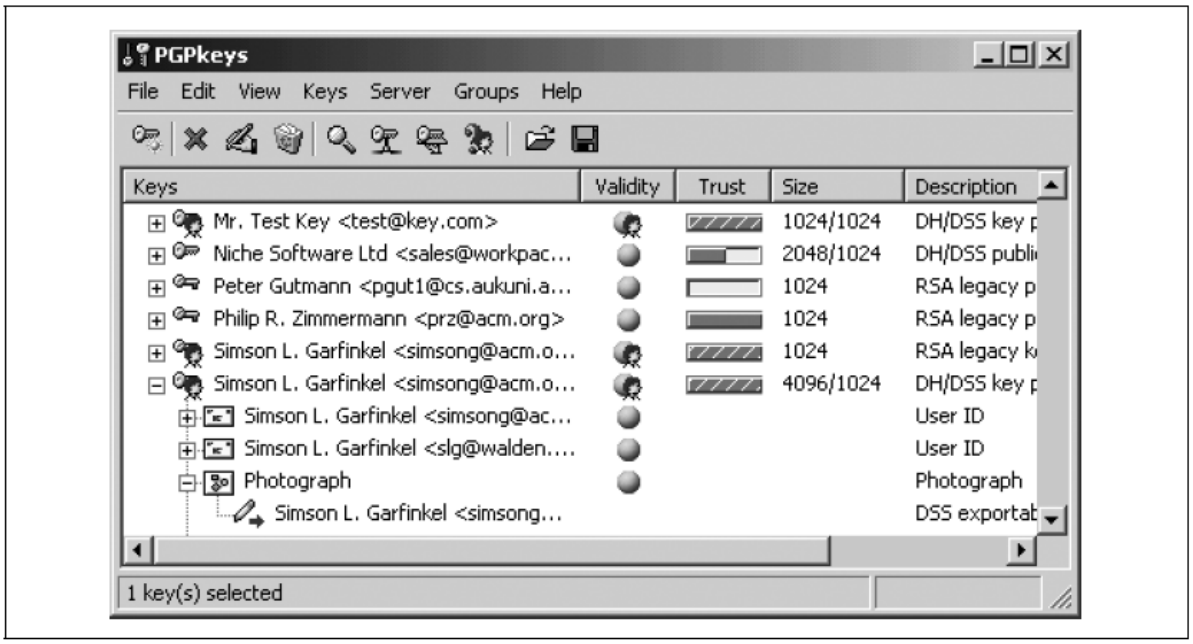

Figure 7-4. The PGPkeys application

The Web of Trust and the key servers

Phil Zimmermann envisioned a decentralized **Web of Trust**, shown in **Figure 7-5**.

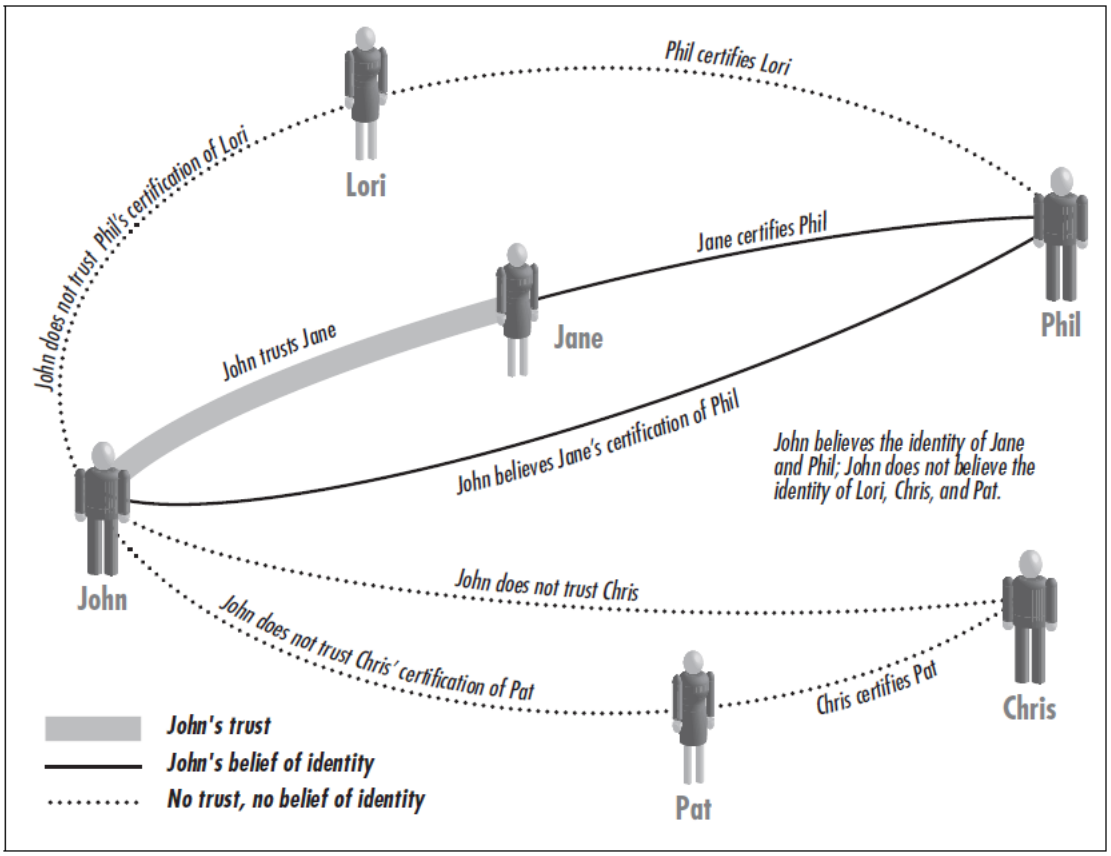

Figure 7-5. The PGP Web of Trust

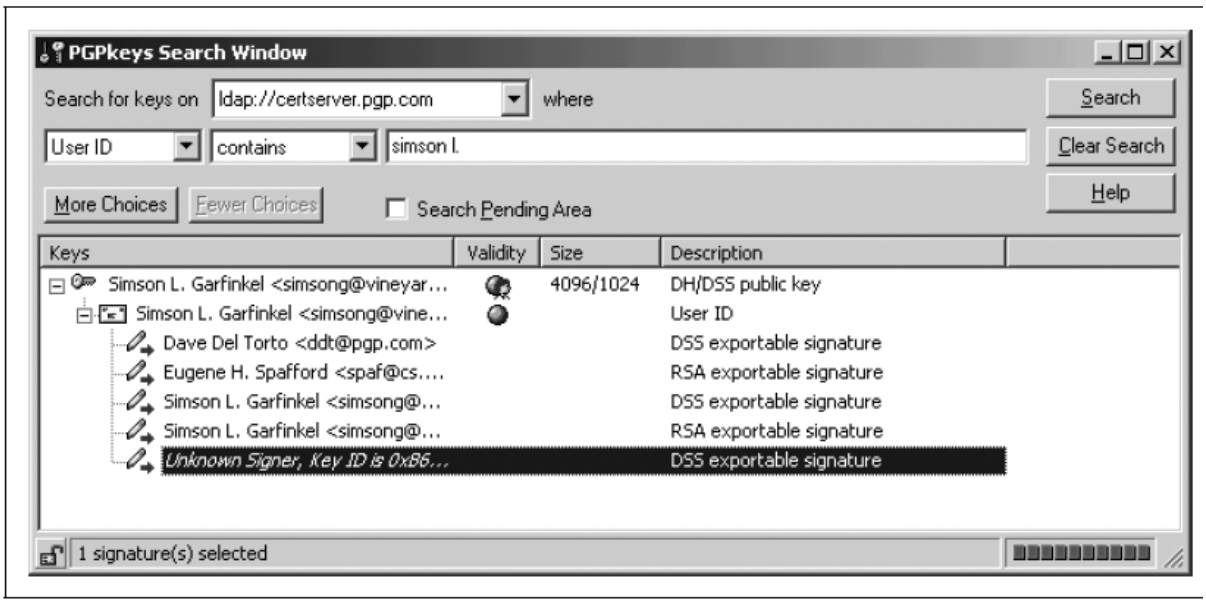

Figure 7-6. Simson's key on the PGP key server has five signatures on it.

Figure 7-6 shows multiple signatures on a public key retrieved from a PGP key server. Trusted signatures help distinguish real keys from fraudulent ones.

Key signing parties

Key signing parties allow users to:

- Exchange keys
- Verify identity using official documents
- Sign each other's keys

While socially effective, they are impractical for large-scale systems and raise privacy concerns.

5.2. Certification Authorities: Third-Party Registrars

A **Certification Authority (CA)** issues digital certificates by signing public keys.

Examples include:

- Internal organizational CAs
- Outsourced CAs
- Trusted third-party CAs (e.g., VeriSign)

Figure 7-7 illustrates a CA certificate structure.

Figure 7-7. A schematic certification authority certificate.

Certification Practices Statement (CPS)

A **CPS** describes:

- How certificates are issued
- How identity is verified
- Liability policies

It answers: *"What does it mean when this organization signs a key?"*

The X.509 v3 Certificate

Most CAs issue **X.509 v3 certificates**.

Each certificate contains:

- Version
- Serial number
- Subject identity
- Public key
- CA signature

Figure 7-8 shows the structure of an X.509 v3 certificate.

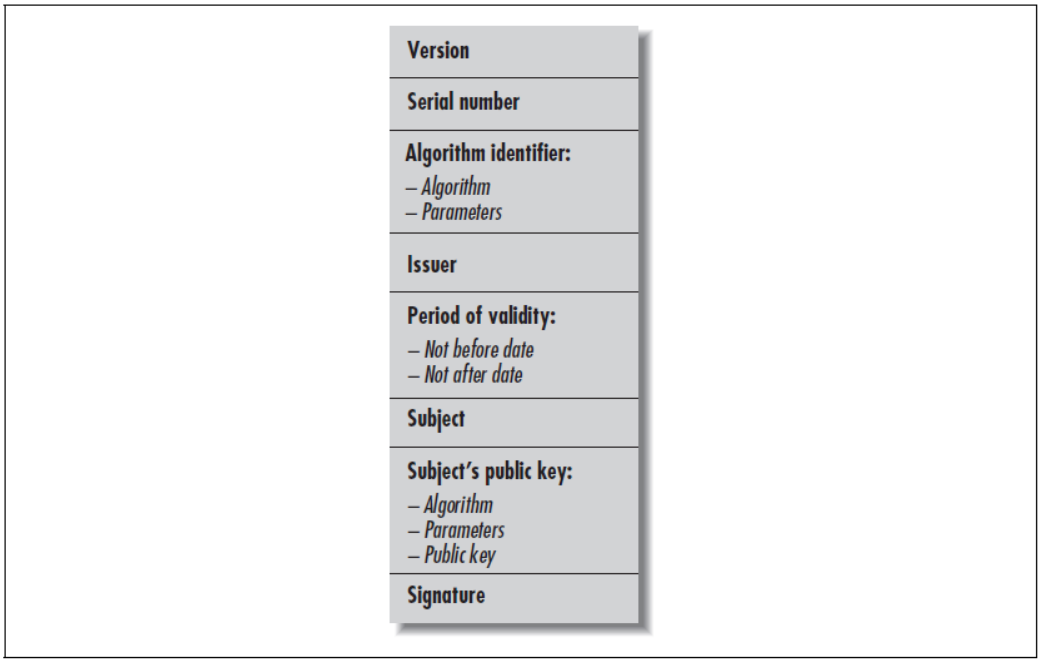

Figure 7-8. The schematic structure of a typical X.509 certificate

Exploring the X.509 v3 certificate

Internet Explorer can display certificate details.

- **Figure 7-9** shows the General properties view.
- **Figure 7-10** shows detailed fields such as Subject, Thumbprint, and validity period.
- **Figure 7-11** shows the **Certificate Path**, illustrating a **certificate chain**.

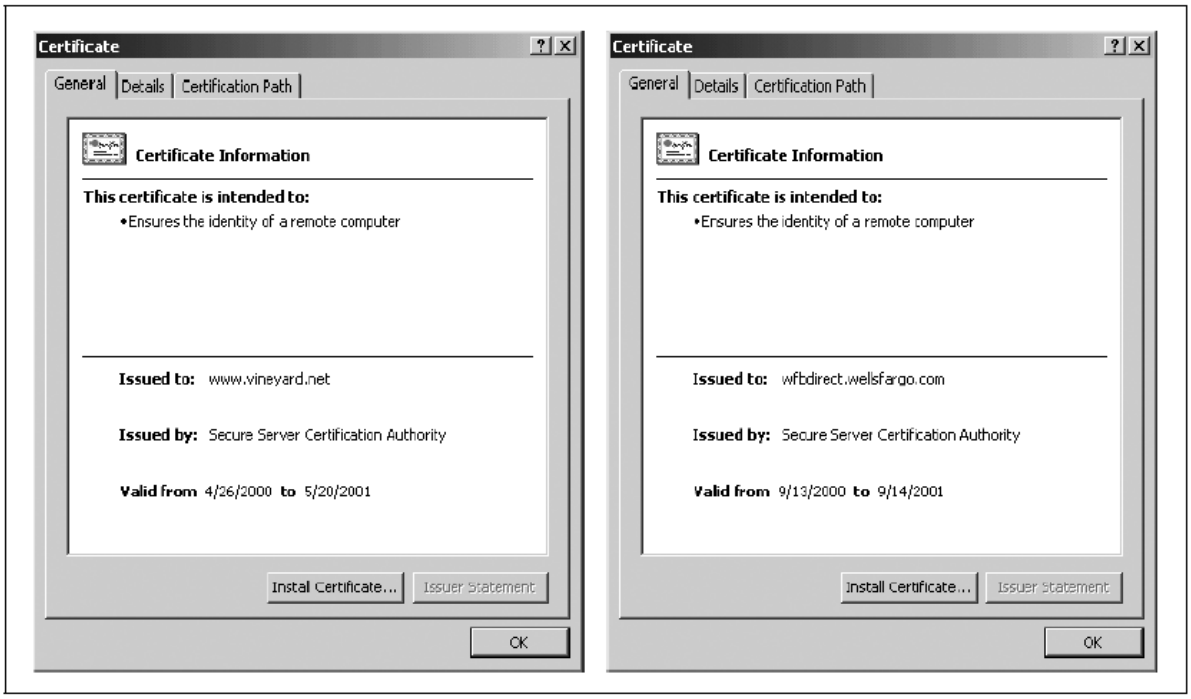

Figure 7-9. The "General" certificate properties, as viewed by Internet Explorer, for certificates downloaded from Vineyard.NET and Wells Fargo.

Field Code	Meaning
CN	Common Name (for SSL certificates, the Common Name should be the DNS address of the server)
OU	Organizational Unit
O	Organization
L	Location
S	State
C	Country

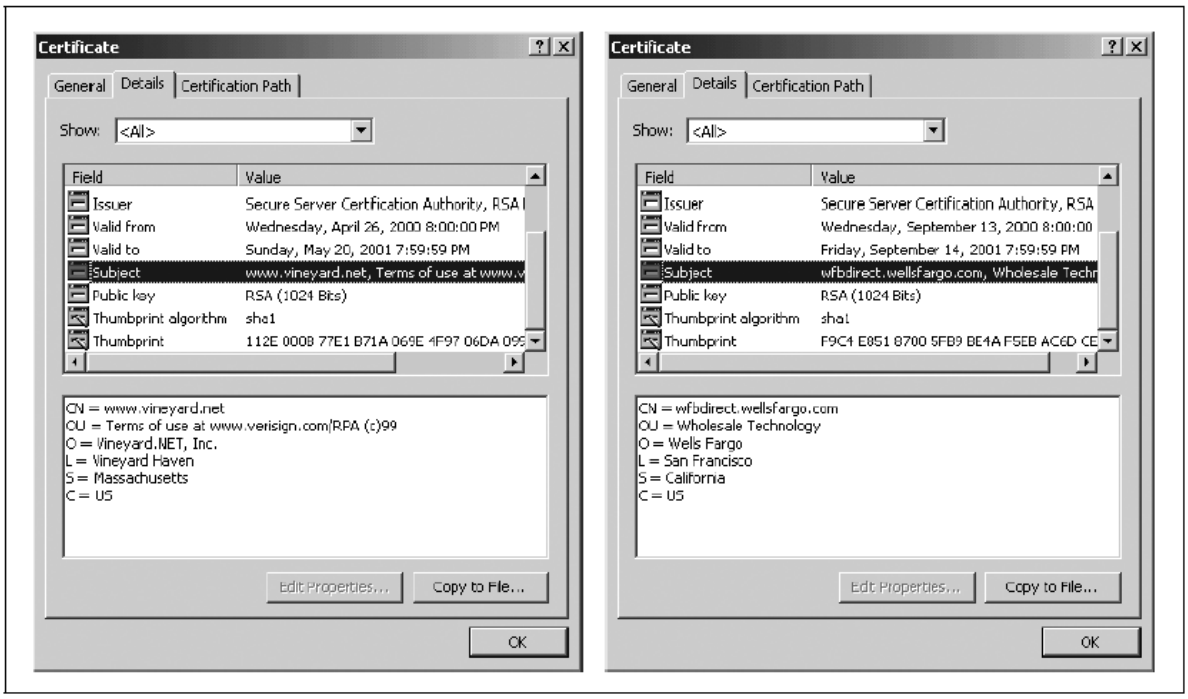

Figure 7-10. Some of the additional fields in the X.509 v3 certificates belonging to Vineyard.NET and Wells Fargo, as displayed by Microsoft Internet Explorer.

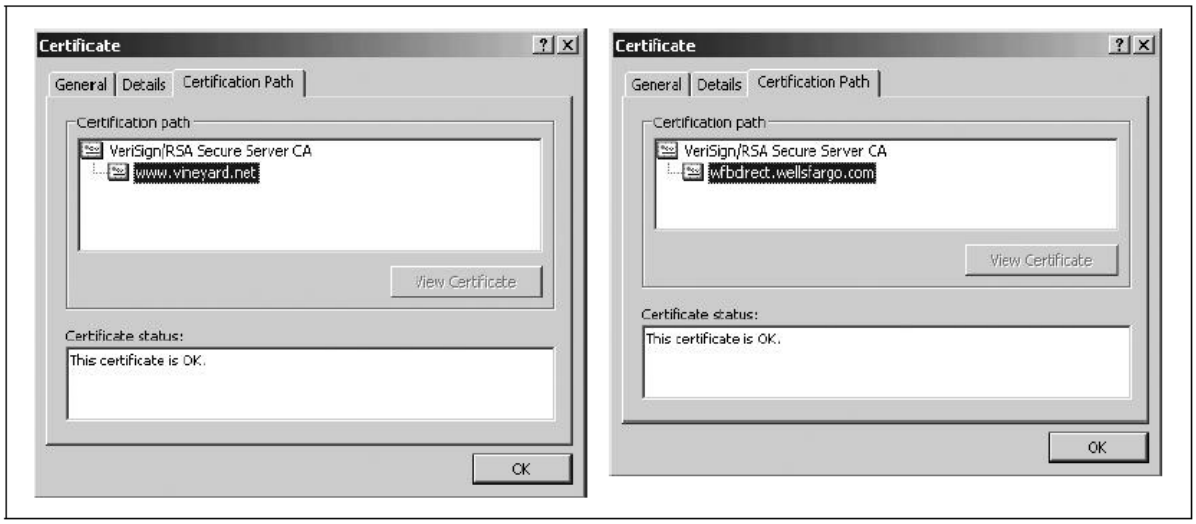

Figure 7-11. The Certificate Path panel for the certificates belonging to Vineyard.NET and Wells Fargo, as displayed by Microsoft Internet Explorer.

Self-signed certificates

Figure 7-12 shows an early self-signed CA certificate.

```
Data:
        Version: 0 (0x0)
        Serial Number:
            02:41:00:00:01
        Signature Algorithm: MD2 digest with RSA Encryption
        Issuer: C=US, O=RSA Data Security, Inc.,
                OU=Secure Server Certification Authority
        Validity:
            Not Before: Wed Nov  9 15:54:17 1994
            Not After: Fri Dec 31 15:54:17 1999
        Subject: C=US, O=RSA Data Security, Inc.,
                 OU=Secure Server Certification Authority
        Subject Public Key Info:
            Public Key Algorithm: RSA Encryption
            Public Key:
                Modulus:
                    00:92:ce:7a:c1:ae:83:3e:5a:aa:89:83:57:ac:25:
                    01:76:0c:ad:ae:8e:2c:37:ce:eb:35:78:64:54:03:
                    e5:84:40:51:c9:bf:8f:08:e2:8a:82:08:d2:16:86:
                    37:55:e9:b1:21:02:ad:76:68:81:9a:05:a2:4b:c9:
                    4b:25:66:22:56:6c:88:07:8f:f7:81:59:6d:84:07:
                    65:70:13:71:76:3e:9b:77:4c:e3:50:89:56:98:48:
                    b9:1d:a7:29:1a:13:2e:4a:11:59:9c:1e:15:d5:49:
                    54:2c:73:3a:69:82:b1:97:39:9c:6d:70:67:48:e5:
                    dd:2d:d6:c8:1e:7b
                Exponent: 65537 (0x10001)
    Signature Algorithm: MD2 digest with RSA Encryption
    Signature:
        88:d1:d1:79:21:ce:e2:8b:e8:f8:c1:7d:34:53:3f:61:83:d9:
        b6:0b:38:17:b6:e8:be:21:8d:8f:00:b8:8b:53:7e:44:67:1e:
        22:bd:97:27:e0:9c:85:cc:4a:f6:85:3b:b2:e2:be:92:d3:e5:
        0d:e9:af:5c:0e:0c:46:95:ff:a1:1c:5e:3e:e8:36:58:7a:73:
        a6:0a:f8:22:11:6b:c3:09:38:7e:26:bb:73:ef:00:bd:02:a4:
        f3:14:0d:30:3f:61:70:7b:20:fe:32:a3:9f:b3:f4:67:52:dc:
        b4:ee:84:8c:96:36:20:de:81:08:83:71:21:8a:0f:9e:a9
```

Figure 7-12. The original RSA Secure Server Certification Authority certificate

Trust arises from:

- Software vendors bundling CA keys
- Social trust in the software ecosystem

Types of Certificates

Four major types are used:

1. Certification authority certificates
2. Server certificates
3. Personal certificates
4. Software publisher certificates

Each supports different authentication needs.

1. Certification Authority (CA) Certificates

- These certificates contain the **public keys of CAs**.
- They include either:
    - the **name of the CA**, or
    - the **name of the particular service being certified**.
- These certificates are typically **self-signed**, that is, signed with the **CA's own private key**.
- CAs can also **cross-certify**, or **sign each other's master keys**.
- The meaning of such cross-certification is an **open question**.
- **Microsoft Windows, Microsoft Internet Explorer, and Netscape Navigator** are shipped with **more than a dozen different CA certificates**.

2. Server Certificates

- These certificates contain:
    - the **public key of an SSL server**
    - the **name of the organization** that runs the server

    - the **DNS name of the server**
- Every **cryptographically-enabled web server** must be equipped with a **server certificate** for the **SSL encryption protocol** to function properly.
- The originally stated purpose:
    - allow consumers to determine the **identity of web servers**
    - prevent **man-in-the-middle attacks**
- In practice, **server certificates are not used for this purpose**.

3. Personal Certificates

- These certificates contain:
    - an **individual's name**
    - the **individual's public key**
- They may include:
    - email address
    - postal address
    - birth date
    - other identifying information
- Some **banks and investment houses** issue digital certificates to depositors.
- Certificates are typically kept on the **depositor's home computer**.
- They provide an **extra level of assurance** when accessing accounts.
- Many **corporations issue digital certificates to employees**.
- Web servers can grant access to anyone with a **valid certificate**.
- This eliminates the need for:
    - entire employee rosters
    - multiple usernames and passwords
- Personal certificates are required for **S/MIME email encryption**.
- Personal certificates are a **substantially more secure** method of identification than usernames and passwords.

4. Software Publisher Certificates

- These certificates are used to **verify the signatures on software**.
- Applied to:
    - ActiveX components
    - downloadable executables
- Every copy of **recent Windows operating systems** is distributed with **software publisher certificates**.
- These certificates are used to **validate signatures on Windows applications** as shown in Figure 7-13.

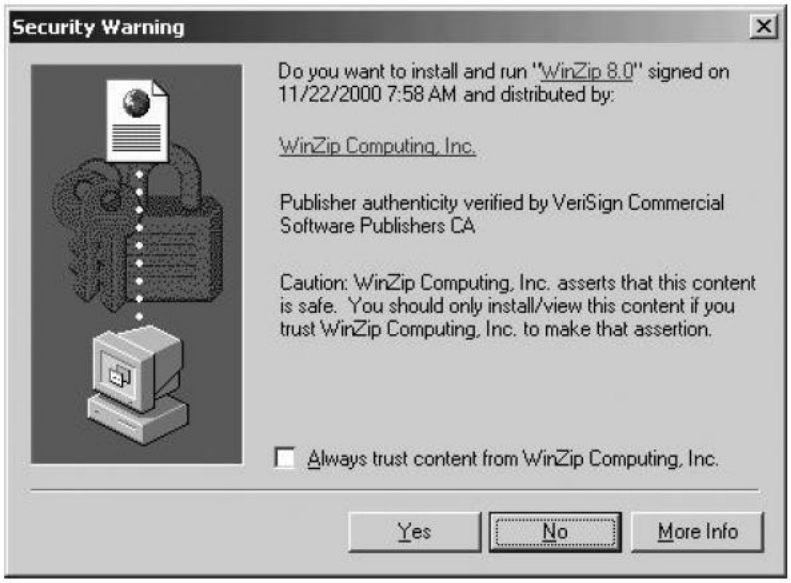

Figure 7-13. Digital signatures and software publisher certificates are used to verify the integrity and authorship of software that is downloaded over the Internet.

How May I Certify Thee?

The Windows operating system allows you to specify for what purposes a certificate can be used. Allowable uses include:

- Server Authentication
- Client Authentication
- Code Signing
- Secure Email
- Time Stamping
- Microsoft Trust List Signing
- Microsoft Time Stamping
- IP security end system
- IP security tunnel termination
- IP security user
- Encrypting File System
- Windows Hardware Driver Verification
- Windows System Component Verification
- OEM Windows System Component Verification
- Embedded Windows System Component Verification
- Key Pack Licenses
- License Server Verification
- Smart Card Logon
- Digital Rights
- File Recovery

Additional purposes can be added on a certificate-by-certificate basis using the "Edit Properties..." button in the Certificate/Details panel (see Figure 7-10).

Netscape Navigator 6.0 also allows you to specify the so-called *trust settings* of what a certificate can be used for (see Figure 7-14). Perhaps because Navigator is not integrated with the operating system, Netscape allows only three uses for each certificate:

- "This certificate can identify web sites."
- "This certificate can identify mail users."
- "This certificate can identify software makers."

Figure 7-14. Netscape Navigator 6.0's Security Manager allows you to specify for what purpose a certificate will be used.

Minimal disclosure certificates

Digital certificates can threaten privacy by revealing excessive information.

Minimal disclosure certificates allow selective proof of facts without revealing identity.

They were invented by **Stefan Brands** and licensed to **Zero Knowledge Systems**.

Revocation

Certificates must be revoked when:

- Keys are compromised
- Certificates are issued incorrectly
- Authorization changes

The **VeriSign–Microsoft incident (2001)** illustrates the importance of revocation.

Certificate revocation lists

A **CRL** lists revoked certificates.

Problems include:

- Large size
- Delays
- Poor implementation

Real-time certificate validation

Alternatives include:

- XML Key Management Specification
- SAML

These avoid CRLs but introduce scalability and DoS risks.

Short-lived certificates

Certificates with very short lifetimes reduce revocation complexity by requiring frequent re-issuance.

5.3. Public Key Infrastructure

Public Key Infrastructure (PKI) is the **collection of digital certificates, certification authorities (CAs), software tools, systems, and hardware** used to deploy and manage public key cryptography. PKI enables secure communication, authentication, data integrity, and nonrepudiation in distributed systems.

The term **"public"** in PKI has historically been ambiguous. Early visions imagined a **government-operated public PKI**, where each citizen would receive a state-certified digital certificate. These certificates were expected to function as **electronic equivalents of driver's licenses**, enabling activities such as digitally signing tax returns and conducting official online transactions.

However, this **public PKI vision did not materialize**. Instead, private companies such as **VeriSign** issued millions of certificates verifying the identities of individuals and organizations. These certificates are widely used today, especially for **web server authentication**, even though the trust hierarchies are run by **private businesses**, not governments. Thus, the word *public* in PKI refers to **public keys**, not public ownership.

Certification Authorities: Some History

When **Netscape Communications** entered the market in 1995, the World Wide Web was rapidly expanding. Browsers such as **Mosaic** were popular, and multiple web servers existed, but none dominated the market.

Netscape's strategy focused on enabling **Internet commerce**, particularly credit-card transactions. However, banks and security experts raised two major objections:

1. **Lack of protection for credit card numbers** during transmission, allowing eavesdropping.
2. **No reliable way to verify the identity of online merchants**, leading to possible fraud.

To address these issues, Netscape developed the **Secure Sockets Layer (SSL) protocol**, which provided:

- Encrypted communication channels
- Server authentication using **digital certificates**

The SSL protocol required that a web server present a **certificate signed by a trusted certification authority**.

Figure reference:

The SSL figure illustrates a secure channel between browser and server, with certificate verification preventing site spoofing.

Netscape's SSL Trust Model

Netscape introduced a **broken key / whole key icon** in its browser:

- **Whole key** → SSL enabled and secure
- **Broken key** → No SSL, insecure

Consumers were advised **not to enter credit card information** on sites without SSL.

Netscape's model achieved:

1. Revenue generation by requiring SSL-enabled servers (Netscape Commerce Server)
2. Mandatory purchase of **CA-signed certificates**

Instead of operating its own CA, Netscape partnered with **RSA Data Security**, which already ran **RSA Certification Services**.

Rise of Competition and VeriSign

Microsoft broke Netscape's dominance by releasing **Internet Information Server (IIS)**. Open-source implementations like **SSLeay** further expanded SSL availability.

Despite this, **competition among CAs failed to flourish**, largely due to RSA's aggressive enforcement of its patents. In 1995, RSA spun off its CA services into **VeriSign**, which quickly became dominant.

Browser Support for Multiple Certification Authorities

Netscape Navigator Versions

- **Version 1.0**: One CA (RSA Secure Server CA)
- **Version 2.0**: User-added CAs allowed
- **Version 3.0**: 16 CAs at 11 organizations *(AT&T, BBN, Canada Post, VeriSign, Thawte, etc.)*

Figure reference:

The certificate manager figure shows viewing, deleting, and adding CA certificates.

Internet Explorer 3.0

Shipped with a subset of Navigator's CA certificates.

Despite technical readiness for competition, **VeriSign became the dominant CA**, absorbing competitors such as **Thawte**.

Internet Explorer Preinstalled Certificates

Internet Explorer ships with **preinstalled certificates**, which users can view via:

1. Internet Options
2. Content tab
3. Certificates button

Figure 7-15 reference:

The Certificates panel figure displays certificate categories.

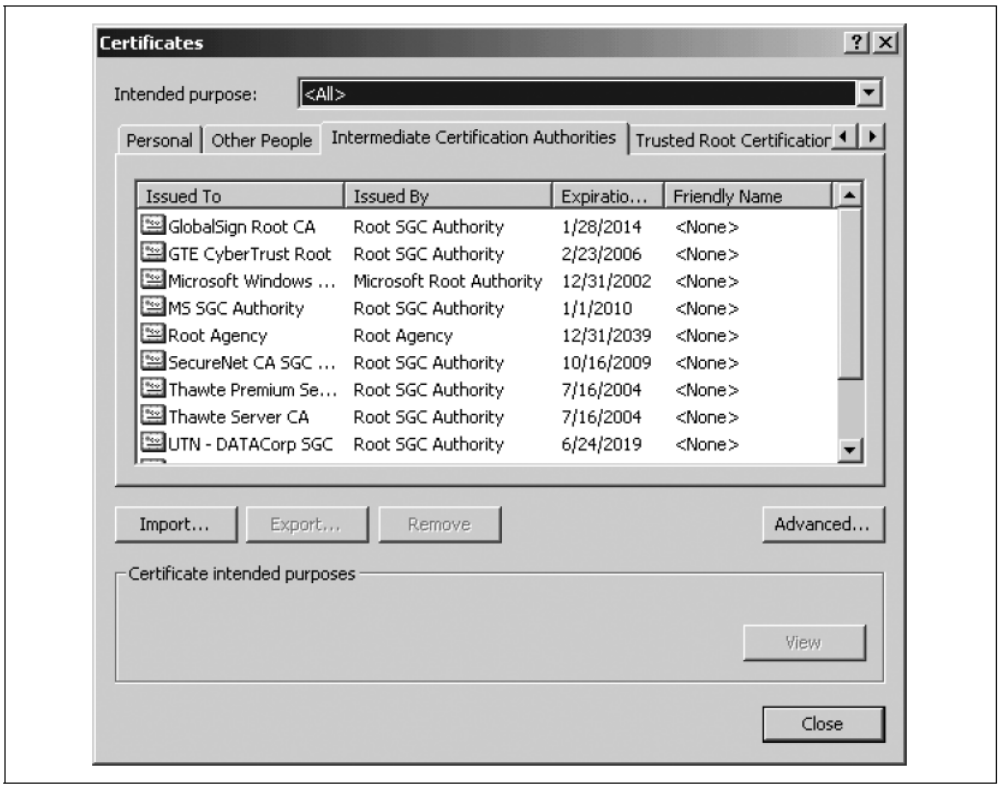

Figure 7-15. Internet Explorer comes with a set of built-in CA certificates.

Certificate Types

Personal

Certificates issued to the user for identification and email encryption. (No default certificates)

Other People

Certificates for verifying others' email signatures. (No default certificates)

Intermediate Certification Authorities

CA certificates bundled but not fully trusted.

Trusted Root Certification Authorities

Self-signed root certificates forming the **roots of trust hierarchies**.

Table 7-1 reference:

The table lists trusted root CAs bundled with Internet Explorer.

IE distributes **107 certificates**, all given equal trust—raising concerns about **unequal real-world reliability**.

Table 7-1. Certification authority keys bundled with Internet Explorer 5.0

Certification authority	Country	# of certificates
ABA.ECOM, Inc.	U.S.	1
Autoridad Certificadora de la Asociacion Nacional del Notariado Mexicano	Mexico	1
Autoridad Certificadora del Colegio Nacional de Corre-duria Publica Mexicana	Mexico	1
Baltimore EZ (Digital Signature Trust)	U.S.	1

Table 7-1. Certification authority keys bundled with Internet Explorer 5.0 (continued)

Certification authority	Country	# of certificates
Belgacom E-Trust	Belgium	1
C&W HKT SecureNet	Hong Kong	4
CA 1 (ViaCode)	Great Britain	1
Certiposte	France	2
Certisign Certificadora Digital Ltda.	Brazil	4
Certplus	France	4
Deutsche Telekom	Germany	2
Digital Signature Trust	U.S.	6
Entrust.net	U.S.	1
Equifax Secure Certification Authority	U.S.	4
EUnet International	N/A	1
FESTE	Spain	2
First Data Digital Certificates	U.S.	1
FNMT	Spain	1
GlobalSign	Belgium	1
GTE CyberTrust	U.S.	3
IPS Seguridad	Spain	1
Microsoft	U.S.	3
National Retail Federation (Digital Signature Trust)	U.S.	1
NetLock Tanusitvanykiado	Hungary	3
PTT Post	Netherlands	1
Saunalahden Serveri Oy	Finland	2
Secure Server Certification Authority, RSA Data Security	U.S.	1
SecureNet	Australia	4
SecureSign, Japan Certification Services, Inc.	Japan	3
Servicios de Certificacion, Servicios Electronicos, Administracion Nacional de Correos	Uruguay	1
SIA S.p.A.	Italy	2
Swisskey AG	Switzerland	1
TC TrustCenter for Security in Data Networks GmbH	Germany	5
Thawte Consulting	South Africa	6
United Parcel Service (Digital Secure Trust)	U.S.	1
UserTrust	U.S.	5
ValiCert Validation Authority	U.S.	3

Table 7-1. Certification authority keys bundled with Internet Explorer 5.0 (continued)

Certification authority	Country	# of certificates
VeriSign	U.S.	21
Xcert EZ (Digital Secure Trust)	U.S.	1

Netscape Navigator Preinstalled Certificates

Netscape Navigator also includes many CA certificates.

Figure 7-14 reference:

The Netscape Personal Security Manager figure shows certificate management.

Figure 7-14. Netscape Navigator 6.0's Security Manager allows you to specify for what purpose a certificate will be used.

Table 7-2 reference:

The table lists CA keys bundled with Netscape Navigator 6.0.

Certification authority	Country	# of certificates
ABA.ECOM, Inc	U.S.	1
AddTrust	Sweden	4
American Express	U.S.	2
Baltimore CyberTrust	U.S.	3
BankEngine	Canada	1
BelSign Object Publishing CA (Since renamed GlobalSign)	Brussels,	4
beTRUSTed	"WW"[a]	1
CertEngine	Canada	1
Deutsche Telekom	Germany	1
Digital Signature Trust	U.S.	4
E-Certify	Canada	2
Entrust.net	U.S.	3
Equifax Secure Certification Authority	U.S.	5
FortEngine	Canada	1
GlobalSign	Belgium	5
GTE CyberTrust	U.S.	5
MailEngine	Canada	1
TC TrustCenter	Germany	5
Thawte Consulting	South Africa	6
TraderEngine	Canada	1
United States Postal Service	U.S.	1
UserTrust	U.S.	5
ValiCert Validation Authority	U.S.	4
VeriSign (and RSA)	U.S.	18
Visa International	U.S.	5
Xcert (Digital Secure Trust)	U.S.	5

[a] This is what the key says; it does not correspond to any particular country code.

Multiple Certificates for a Single CA

Some CAs issue **multiple certificates** to indicate different trust levels. **VeriSign** has over **21 certificates**.

Table 7-3 and Table 7-4 references:

These tables show the evolution of VeriSign certificates (1996 vs 2001).

Table 7-3. VeriSign certificates in 1996

Certificate name	Certificate type	Certification practice	Cost	Liability protection
Class 1	Client[a]	VeriSign assures that the user can receive email at the given address and that no other certificate for the email address has been issued.	Free (nominally $9.95/year)	$100
Class 2	Client	VeriSign assures the identity of a digital ID holder through online identity verification against a consumer database.	$19.95/year	$5,000
Class 3	Client	VeriSign validates the entity applying for the certificate using background checks and investigative services.	$290/first year; $75/renewal	$100K
Secure Server	Server	VeriSign validates the entity applying for the certificate using background checks and investigative services.	$290/first year; $75/renewal	$100K

Table 7-4. VeriSign certificates in 2001

Certificate name	Certificate type	Strength[a]	Certification practice	Cost	NetSure protection
Class 1 Digital ID	Client[b]	N/A	VeriSign assures that the user can receive email at the given address.	$14.95 per year	$1000
Secure Site	Server	40-bit	VeriSign validates the entity applying for the certificate by verifying the organization's address using Dunn & Bradstreet.	$349 per year	$100K
Secure Site Pro	Server	128-bit	VeriSign validates the entity applying for the certificate by verifying the organization's address using Dunn & Bradstreet.	$895	$250K
Commerce Site	Server	40-bit	VeriSign validates the entity applying for the certificate by verifying the organization's address using Dunn & Bradstreet. Price includes a performance audit of the web site from two cities.	$995	$100K
Commerce Site Pro	Server	128-bit	VeriSign validates the entity applying for the certificate by verifying the organization's address using Dunn & Bradstreet. Price includes a performance audit of the web site from ten cities.	$1495	$250K
OnSite for ServerIDs	Intermediate CA	40-bit or 128-bit	After validating an organization and negotiating a fee, VeriSign issues a certificate that allows the organization to issue its own certificates for SSL servers throughout its own enterprise.	Negotiated	

VeriSign introduced **NetSure Protection**, an extended warranty program with **per-certificate liability limits**, not per-transaction limits.

Shortcomings of Today's CAs

Lack of permanence for Certificate Policies field

Certificates contain URLs pointing to **Certification Practice Statements (CPS)**.

Figure 7-16 and Figure 7-17 references:

Figures show CPS URLs that are no longer accessible, even though certificates remain valid.

CAs must maintain CPS URLs **for the lifetime of certificates**, possibly **20+ years**, but many fail to do so.

Figure 7-16. The General panel of Internet Explorer's Certificate window shows general information abut a certificate

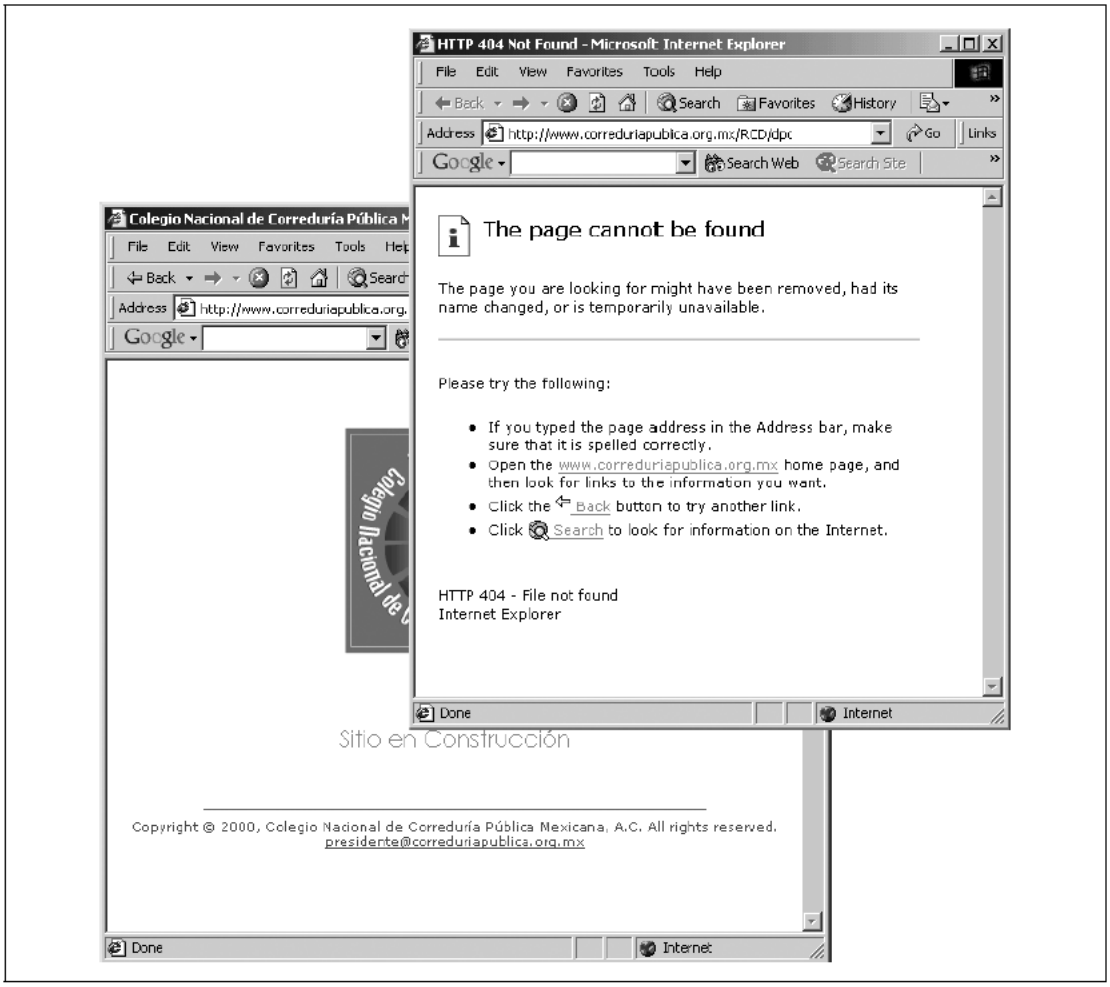

Figure 7-17. All CAs are not created equal. The home page of the web server for the Colegio Nacional de Correduria Publica Mexicana, A.C. reveals that the site is "en construcción"—and has been, apparently, for more than a year. The URL for the CA's certification practies statement does not exist. Yet this CA's key is fully trusted by Internet Explorer.

Inconsistencies for "Subject" and "Issuer" fields

Different CAs use **inconsistent Distinguished Name (DN) formats**, making automated verification difficult.

Examples include:

- ValiCert
- VeriSign
- PTT Post
- SecureNet

Consistency is critical for **programmatic certificate validation**.

Unrealistic expiration dates

Early certificates expired too soon (1999). Later certificates swung too far in the opposite direction, with expiration dates extending to **2028**.

Figure 7-18 reference:

The figure shows long-lived certificates with potentially weak cryptographic strength.

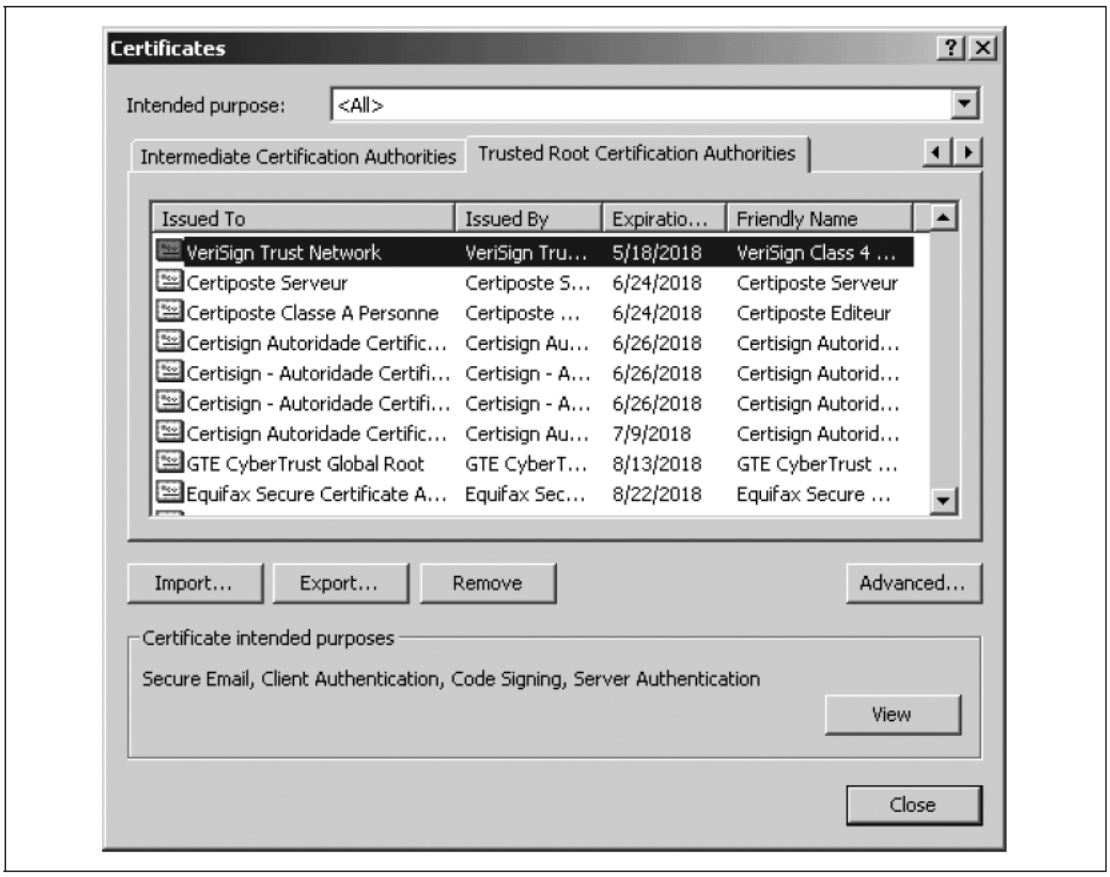

Figure 7-18. VeriSign distributes many keys with Internet Explorer 5.5 that have unrealistically late expiration dates

5.4. Open Policy Issues

Despite growing fraud and digital signature legislation, **widespread PKI adoption remains elusive**. Managing millions of certificates across many CAs remains largely untested at scale.

Private Keys Are Not People

Digital signatures prove **access to a private key**, not identity.

End-user systems are often insecure:

- Malware
- Viruses
- Trojan horses
- Weak random number generators

Even encrypted private keys must be decrypted for use, making them vulnerable.

Smart cards may help, but **cannot guarantee absolute security**.

Distinguished Names Are Not People

Possession of a certificate does not guarantee that the **Distinguished Name** is accurate.

Key issues:

- Trustworthiness of CAs
- Audits and accreditation
- Policy enforcement
- Accidental or fraudulent issuance

There Are Too Many Robert Smiths

Names alone are insufficient identifiers in large populations.

Certificates must include **additional unique identifying information**.

Today's Digital Certificates Don't Tell Enough

Modern certificates lack:

- Age
- Gender
- Photograph
- Biometrics

Adding such data raises **serious privacy concerns**, highlighting the tension between **identity and anonymity**.

X.509 v3 Does Not Allow Selective Disclosure

X.509 certificates do not support selective disclosure.

Alternative approaches:

- Multiple certificates
- SPKI project

Selective disclosure allows proving **specific attributes** without revealing full identity.

Digital Certificates Allow for Easy Data Aggregation

Digital certificates may become **powerful aggregation tools**, enabling large-scale profiling more effectively than Social Security numbers.

How Many CAs Does Society Need?

Key questions:

- One CA vs many CAs
- Centralized power vs fragmentation
- Risk of exclusion from cyberspace

Carl Ellison questions whether identity certification always requires CAs.

How Do You Loan a Key?

Key-sharing scenarios raise unresolved questions:

- Delegation
- Role-based keys
- Legal responsibility
- Revocation

Why Do These Questions Matter?

Digital signatures are **brittle**:

- Minor changes invalidate signatures
- Fail to show *where* changes occurred

Paper documents still offer advantages in **tamper visibility**.

Brad Biddle on Digital Signatures and E-SIGN

This section outlines the **legal evolution** of electronic signatures:

- Utah Digital Signature Act
- Criticism of PKI-centric laws
- Shift to technology-neutral approaches

E-SIGN and UETA

E-SIGN and UETA:

- Recognize electronic signatures
- Are technology-neutral
- Replace Utah-style PKI mandates

Electronic Contracting

Electronic contracts follow traditional contract principles:

- Offer
- Acceptance
- Consideration

Digital signatures improve **proof**, not validity.

"Signed Writing" Requirements

Most contracts do not require signed writings.

E-SIGN and UETA simplify compliance for electronic records.

Proof

Proof of:

- Contract formation
- Contract terms
- Party identity

Digital signatures reduce disputes but are not always cost-effective.