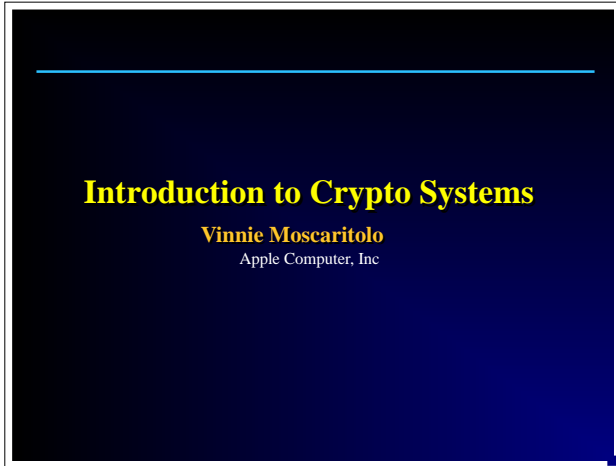


---

## Introduction to Crypto Systems



---

---

---

---

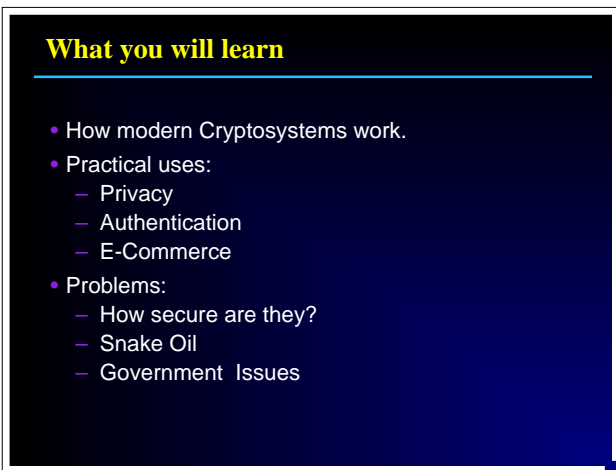
---

---

---

---

## What you will learn



---

---

---

---

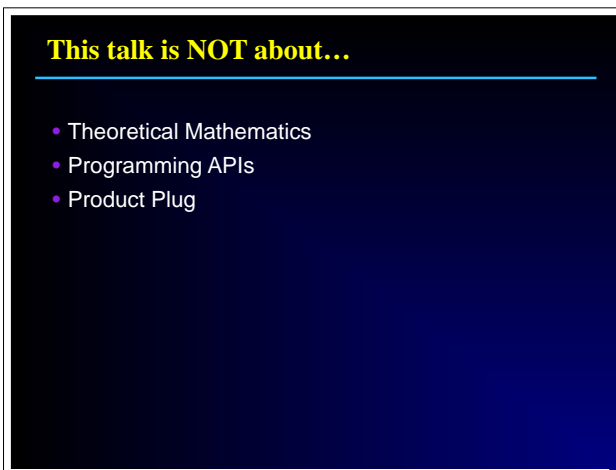
---

---

---

---

## This talk is NOT about...



---

---

---

---

---

---

---

---

### Who this talk is for

- Executives
- Marketing
- Engineers
- Anyone who needs to be Crypto competent
- You?



## Who this talk is for

---

---

---

---

---

---

---

---

### Background

- Who is Vinnie Moscaritolo ?
  - Apple Developer Services
  - (previously) Chief Consulting Engineer, PGP
  - Hosts the Mac-Crypto Workshop
  - Not a Cryptographer
  - Not even a Lawyer
  - Lots of "real world" security experience
  - Engineer > 15 years
  - <http://www.vmeng.com/vinnie/>

## Background

---

---

---

---

---

---

---

---

### What is Cryptography?

**crypt•tog•raphy** (krip-tog'-re-fe) *n.* The art and science of keeping messages private.



## What is Cryptography?

---

---

---

---

---

---

---

---

### Cryptography is not Security

Cryptography is to security  
what bricks are to buildings..

## Cryptography is not Security

---

---

---

---

---

---

---

---

### What is Encryption?

- Encryption uses mathematical algorithms:
  - to scramble data so that it is very difficult for anyone other than intended recipients to recover the original plaintext.
- Allows sensitive information to be:
  - stored on insecure computers
  - transmitted across insecure networks
- In order to recover the data:
  - recipient must have correct decryption key
- Confidentiality

## What is Encryption?

---

---

---

---

---

---

---

---

### Cryptography is not new

- Cyphered Pottery Glaze formula - 1500 B.C.
- Cypher like transformations in the Bible, Jeremiah 25:26, 51,41
- The Greeks described substitution cyphers
- The Kama-sutra lists secret writing as one of the 64 arts a woman should know and practice.
- Cryptography was widely used in Europe during the Renaissance
- "One if by land, two if by sea" - cryptography 1775
- See David Kahn's "the Code-Breakers" for more info.

## Cryptography is not new

---

---

---

---

---

---

---

---

### Whats Happening Today

- World Economy: Industrial -> Information Age
  - Information is bought & sold
  - Informational Assets
- National Borders & Economies are interlinked
  - via InterNet.
- Secure Networks --> Open Network  
Insecure Comm --> Secure Comm

## Whats Happening Today

---

---

---

---

---

---

---

---

### Why is Cryptography important?

- People are really starting to depend on the Internet.
  - but is not a secure channel
- Cryptography provides Privacy / Protection
  - from competitors, business rivals, news media
  - from "Governments" ( foreign & domestic )
  - from "the Bad Guys"
- Cryptography helps keep things secret!

## Why is Cryptography important?

---

---

---

---

---

---

---

---

### Threats are emerging from

- Foreign Intel Agencies
- Network Hackers & E-Vandals
- Thieves
- Disgruntled Employees / Insiders
  - Eavesdropping / Data Browsing
  - Clandestine Alteration of Data
  - Spoofing
- Bad Design (Y2K)

## Threats are emerging from

---

---

---

---

---

---

---

---

### What kind of secrets?

- Personal and Business Communications
  - Telephone conversations, Fax , E-mail
- Financial
  - Electronic Funds Transfer
- Sensitive Biz Info
  - Trade secrets, source code, payoffs
- Critical Civilian Infrastructure Comm
  - Air Traffic Control, Power Grid, Telephone Network
- Personal Info
  - Health records, Personal files, etc

## What kind of secrets?

---

---

---

---

---

---

---

---

### Cryptography is not just about secrets

- Cryptography is also about trust and reputation.
  - Authentication
  - Integrity
  - Nonrepudiation

## Cryptography is not just about secrets

---

---

---

---

---

---

---

---

### Uses of Cryptographic Authentication

- Access Control
  - remote servers
- Web based software distribution
  - USB drivers
  - Shareware

## Uses of Cryptographic Authentication

---

---

---

---

---

---

---

---

### Crypto is becoming ubiquitous

- Crypto is not just for internet e-mail
- You will find it in:
  - Cellular phones
  - Cable/Sat TV broadcasts
  - radio modems
  - Smart cards, iButton
  - DVD
  - Garage door openers

## Crypto is becoming ubiquitous

---

---

---

---

---

---

---

---

### Why did it take so long to get here

- Lack of
  - Security Awareness (Denial)
  - Critical Mass
  - Supporting Infrastructure
  - Independent Certification
  - Standards / Interoperability
  - Performance
  - Usability
- Too Much
  - Cost
- FUD
  - Government Policy
  - Patent Issues

## Why did it take so long to get here

---

---

---

---

---

---

---

---

### Crypto = New Biz Opportunities

- Virtual Corp (Geographically Dispersed Units)
- Customer / Vendor EDI
- E-Commerce
  - World Wide Marketplace w/o Middleman
  - Small Biz / Large # of Customers
  - Cottage Industries!

## Crypto = New Biz Opportunities

---

---

---

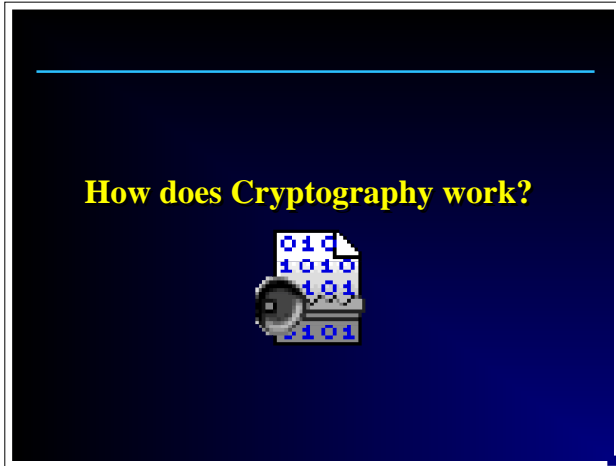
---

---

---

---

---



## How does Cryptography work?

---

---

---

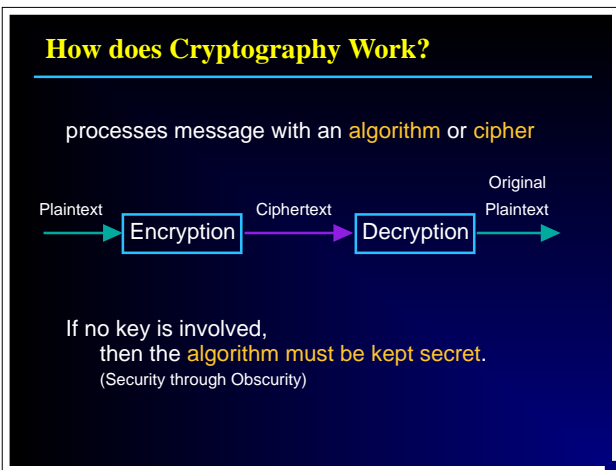
---

---

---

---

---



## How does Cryptography Work?

---

---

---

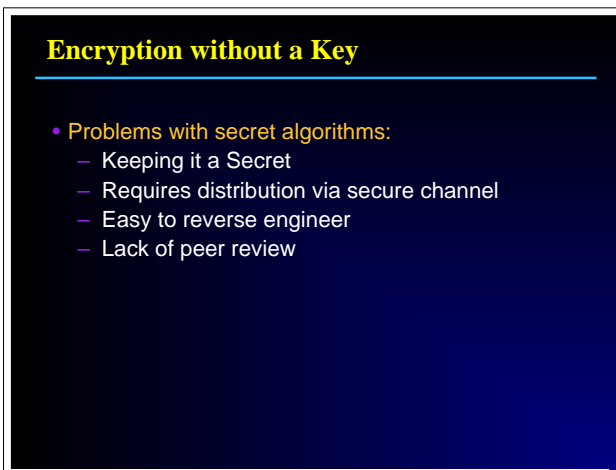
---

---

---

---

---



## Encryption without a Key

---

---

---

---

---

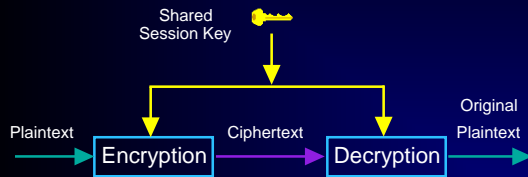
---

---

---

## Encryption with a Secret Key

- **Secret** (Conventional or Symmetric) **Key** Algorithm
- Same key is used to encrypt & decrypt
- Algorithm is published.
- **Session Key** must be kept secret.



## Encryption with a Secret Key

---

---

---

---

---

---

---

---

## Key vs. Passphrase

- **Key**
  - Must be random
  - All keys must be equally probable
- **Passphrase**
  - Must be easy to remember
  - Actually has smaller keyspace
  - Dictionary attack

## Key vs. Passphrase

---

---

---

---

---

---

---

---

## Secret Key Encryption Algorithms

- **DES** - 56 bit key (easily broken with special hardware)
- **3 DES** (more secure)
- **IDEA** - 128 bit key
- **RC2/RC4** - key size variable
- **BlowFish** - key size variable
- **TwoFish** - (128, 192 or 256 bit key)
- **CAST** - Northern Telecom, 128 bit key
- **SkipJack** - 80 bit Key (used in Clipper)

## Secret Key Encryption Algorithms

---

---

---

---

---

---

---

---

### The problems with Secret Keys are..

- Selecting a Secret Key that can't be guessed
  - Is the algorithm predictable?
- Negotiating the shared secret key across an unsecured channel and still keeping it a secret.
  - Key Distribution
  - Key Security



## The problems with Secret Keys are..

---

---

---

---

---

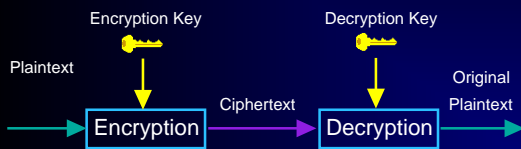
---

---

---

### Public (non-secret) Key Encryption

- Different key for encrypt and decrypt operations
- Credit for invention given to Whitfield Diffie and Martin Hellman in 1976
- James Ellis of British Gov Comm HQ in 1970 ( but was a military secret )
- IMHO Most important invention of 70s



## Public (non-secret) Key Encryption

---

---

---

---

---

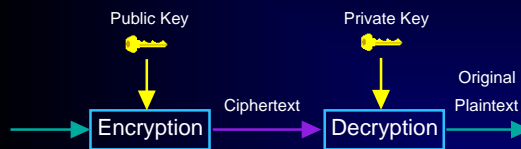
---

---

---

### Public Key Encryption

- Key Pairs
- Publish one key - keep other secret.
- Anyone who wants to send you a message **encrypts** it with your **public key**.
- To read the message you **decrypt** it with the **private key**.



## Public Key Encryption

---

---

---

---

---

---

---

---

## Public Key Encryption

- A good public key algorithm:
  - Infeasible to derive one key from other.
  - Keys are interchangeable
- Simplifies (but doesn't solve) key distribution problem
- Public Key is slower than Secret Key Algorithms
  - (RSA is about 1000-5000 times slower than DES)
  - Public Key Encryption is sometimes used to encrypt a Secret Key Algorithms Session key

## Public Key Encryption

---

---

---

---

---

---

---

---

## Public Key Encryption Algorithms

- RSA (Rivest, Shamir, Aldeman)
  - gets security from difficulty of **factoring** large (100 - 200 digit) prime numbers.
  - considered secure when longer (>768 bit) keys are used
  - ( Encumbered by patent )
- El Gamal (DH)
  - gets security from difficulty of calculating **discrete logarithms** in a finite field.
  - security similar to RSA for same key lengths
  - overcomes some weakness with RSA
  - No patent problems

## Public Key Encryption Algorithms

---

---

---

---

---

---

---

---

## Public Key Encryption Algorithms

- **Elliptic Curve**
  - Shows promise for future cryptosystems.
  - More resistant to brute force attack
  - Highest crypto strength per bit of key
    - 160 bit EC key  $\approx$  1024 bit RSA key  $\approx 10^{12}$  MIPS years
    - 320 bit EC key  $\approx$  5120 bit RSA key  $\approx 10^{36}$  MIPS years
  - Shorter key = Savings in storage, computation, bandwidth
- **Elliptic Curve Cryptography** is ideal for
  - limited computation power (Smartcards, wireless devices, etc)
  - intensive use of signing or encryption (web based TP)
  - high speed/ bandwidth devices
- **Legal & Market issues**
  - hinders acceptance

## Public Key Encryption Algorithms

---

---

---

---

---

---

---

---

## Algorithms Patents

- RSA - US only, expires Sep-20-2000
- DH - patent by Cylink, expired Sept-6-1997 (GATT)
- IDEA - patent by Ascrom Systec AG, Switzerland
- DES - patent by IBM, patent is expired
- RC2, RC4 were trade secrets
  - implementation were published on net
  - names are protected by trademark
- Blowfish - not patented
- ElGamal - not patented
- Elliptic Curve - patents on implementations, not algorithm  
Certicom, NeXT/Apple, etc

I Am Not A Lawyer!!

## Algorithms Patents

---

---

---

---

---

---

---

---

## Digital Signatures



## Digital Signatures

---

---

---

---

---

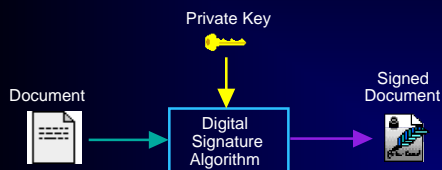
---

---

---

## What are Digital Signatures?

- Works with your **private key** to:
  - Authenticate that a message came from you.
  - ensures data has not been changed (**Integrity**)
  - makes it hard for you to **repudiate** your knowledge of data



## What are Digital Signatures?

---

---

---

---

---

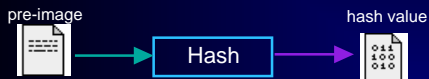
---

---

---

## OneWay or Secure Hash

- Takes a variable length input string and creates a shorter fixed length (128-256 bits) summary string or hash-value.
- hard to find input for a given hash (non reversible)
- difficult to find two pre-images with same hash
- hash-value aka
  - message digest
  - fingerprint
  - cryptographic checksum



## OneWay or Secure Hash

---

---

---

---

---

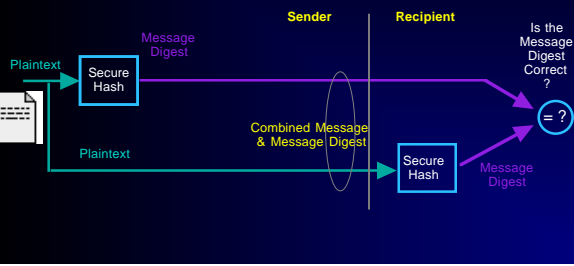
---

---

---

## Checking a message's validity

By comparing the Message Digest of a candidate pre-image to that of a given pre-image you can ensure that the message most likely hasn't been tampered with.



## Checking a message's validity

---

---

---

---

---

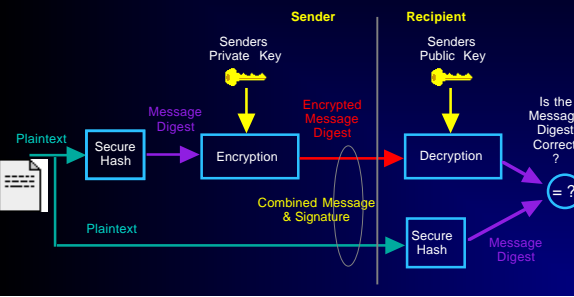
---

---

---

## Creating a Digital Signature w Encryption

Encrypting the message digest with the sender's public key ensures that the message originated from the sender.



## Creating a Digital Signature w Encryption

---

---

---

---

---

---

---

---

## Digital Signature Example

• -----BEGIN PGP SIGNED MESSAGE-----

By comparing the Message Digest of a candidate pre-image to that of a given pre-image you can ensure that the message most likely hasn't been tampered with.

-----BEGIN PGP SIGNATURE-----  
Version: 5.0 beta  
Charset: noconv

iQCVAwUBM49V3vMF2+rAU+UGAQFW7wP+PHxIH0geLUaWy1yoWJG/NShyzByM3rb  
m/NgLOQ+wuro+NcF21jK4WTYoDEoF4fr4he4mnoxBgksCEYyJhoJYMPtgn0ltT99  
PnEdni/EAdmj56DckVTnV8SB6LxouE3TV7o+ehOULZiCP6WanfeLZVCCi2iQ11LK  
7dal+VncGHM=  
=VKa9  
-----END PGP SIGNATURE-----

## Digital Signature Example

---

---

---

---

---

---

---

---

---

---

## Digital Signatures Algorithms

- Message Digests (Secure Hash)
  - MD5 (RSA)
    - Yields 128 bit hash
    - recently found to have weakness.
  - SHA, SHA-1
    - Yields 160 bit hash
- Signature Only Algorithms
  - DSA (DSS)
    - not intended for encryption use
    - developed by NSA

## Digital Signatures Algorithms

---

---

---

---

---

---

---

---

---

---

## Digital Signature Uses

- Signatures can be used on
  - Text
  - Code (virus detection)
  - Public Keys
  - Other signatures
  - Random Challenge Strings (authentication)
- Signatures can be used with
  - Timestamps
  - Name & Directory Servers
  - Software Distributions

## Digital Signature Uses

---

---

---

---

---

---

---

---

---

---

### Other Digital Signature Modes

- Standard Digital Signature requires:
  - signer to know contents of message
  - anyone with public key can verify correctness of signature without consent of signer
  - self-authenticating
- Blind Signatures
  - sign without being able to read contents
- Group Signatures

## Other Digital Signature Modes

---

---

---

---

---

---

---

---

### Keys and Key Management



## Keys and Key Management

---

---

---

---

---

---

---

---

### Key Mangement

- Key management is hardest part of cryptography
- Answers the questions
  - How do I get a key pair?
  - How do I get someone else's public key?
    - and how do I know its really theirs?
  - What should I do if I lose my key?
    - or it's stolen?
  - How long is my key good for?

## Key Mangement

---

---

---

---

---

---

---

---

## Lifetime of a Key

- **Creation & Registration**
  - centralized (Kerberos) / distributed (RSA, PGP)
- **Certification**
  - centralized (X.509) / distributed (PGP)
- **Distribution**
  - introduction (PGP)
  - storage / backup (escrow)
- **Usage**
  - validity checking
- **Termination & Revocation**
  - deliberate destruction or expired
  - archival

## Lifetime of a Key

---

---

---

---

---

---

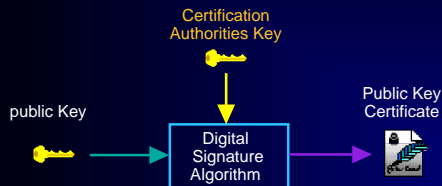
---

---

## Certification & Trust Management

Q: How does Alice know that a public key is Bob's and not someone pretending to be Bob ?

A: Bob's public key is signed by a **trusted entity**.



## Certification & Trust Management

---

---

---

---

---

---

---

---

## What kinds of Certificate?

- **Identity Certificate**
  - Binds the name of an identity to a public key.
  - X.509, PGP
  - ID Card
- **Meta Certificate**
  - Delegates an attribute or authority to a public key.
  - SPKI
  - Permission Slip

## What kinds of Certificate?

---

---

---

---

---

---

---

---

## Identity Certificates

- X.509
  - Originates from X.500 database design
  - Names are organized into a hierarchy
    - Corporate roles, Notarized Documents.
  - Requires Certificate Authority (CA)
- OpenPGP
  - Signer of key might not be known or trusted
  - Allows independent signatures to vote a binding into validity
  - "Web of Trust"

## Identity Certificates

---

---

---

---

---

---

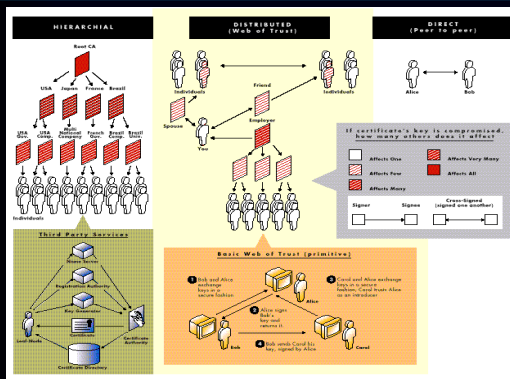
---

---

---

---

## Trust Models



## Trust Models

---

---

---

---

---

---

---

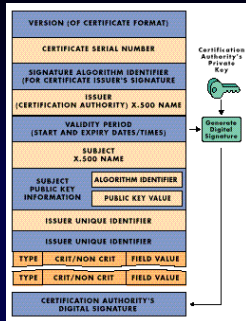
---

---

---

## X.509 Certificate

- Each user has distinct name
- Signed by hierarchical CA
  - tree structure
- Protocol to verify validity
- Keys can be
  - individuals
  - groups / organizations



## X.509 Certificate

---

---

---

---

---

---

---

---

---

---

### X.509 Certificate

- Keys need to be signed into validity by CA
- Everyone must trust the same CA
  - Interoperability issue
  - If CA is compromised, entire structure falls
- Hierarchical model only.
  - works best in corp environment
  - not the best model for personal use
- Some keys have privacy issues
  - (name, address, phone number of owner)

## X.509 Certificate

---

---

---

---

---

---

---

---

### OpenPGP Certificate

- No CA that everyone needs to trust
  - Every copy of PGP is a CA
  - No implicit guarantee of validity
  - Flex trust policy
  - Key can only be trusted if there is path of signatures between verifier and sig in question
- Network Model
  - can simulate hierarchical model (corp key)
  - Web of Trust
- Keys can be
  - individuals, groups, organizations, network services
  - pseudonyms, anonymous addresses

## OpenPGP Certificate

---

---

---

---

---

---

---

---

### OpenPGP Certificate

- Primary Signing key
- Encryption Key(s)
- Revocation Key(s)
- User ID(s)
  - Email address
  - Signed by other keys
  - (Optional) ARR
- Notations
  - mechanism for adding new stuff

<b>Primary Signing Key (DSS)</b> <ul style="list-style-type: none"><li>- Algorithm, parameters</li><li>- Validity Period</li><li>- Key material</li></ul>
<b>Encryption Subkey (s)</b> <ul style="list-style-type: none"><li>- Algorithm, parameters</li><li>- Validity Period</li><li>- Key material</li><li>[ signed by signing key ]</li></ul>
<b>Revocation Key</b> [ signed by signing key ]
<b>User IDs</b> <ul style="list-style-type: none"><li>- Namestring</li><li>- Addition Recipient Request</li><li>[ signed by signing key ]</li><li>[ signed by other keys ]</li></ul>
<b>Notations</b> <ul style="list-style-type: none"><li>-extension mechanism</li></ul>

## OpenPGP Certificate

---

---

---

---

---

---

---

---

## OpenPGP Certificate (example)

```
Vinnie Moscaritolo <vinnie@vsmeng.com>
Preferred Algorithms: CAST IDEA IDEA
Key ID: 026838F042 sub Key Date: 18284 06/27/1997 - never
Fingerprint: 3F90 3472 C3AF 622D 5D91 8D98 D881 0009 083E F042
-----
Key ID: 026838F042 sub Key Date: 40964 06/27/1997 - never
Fingerprint: 2C58 8158 40C5 0808 07F2 1848 5541 F31A C1C0 D2D5
-----
--S- Vinnie Moscaritolo <vinnie@vsmeng.com>
--S- Vinnie Moscaritolo <vinnie@apple.com>
--S- Vinnie Moscaritolo <vinnie@vsmeng.com>
--S- Leland A Wallace <raldwall@apple.com>
--S- Martin Minow <minow@apple.com>
--S- Lucky Green <vsmeng@vsmeng.com>
--S- Jon Callas <jon@ppp.com>
--S- Tim Holmes <shotstop@apple.com>
--S- Robert A. Hettling <rah@shipwright.com>
--S- Rodney Thayer, Sable Technology Corporation <rodney@sabletech.com>
--S- Ignatius Piazza, Front Right Firearms Training Institute <ignatius@frontsight.com>
--S- Cynthia Hertling <cynthia.hertling@sun.com>
--S- Dave Del Torto <ddt@ppp.com>
--S- Dave Polaschek <dave@metoo.com>
--S- Black Bullcom <vsmeng@blackbull.com>
--S- Dave Heller <deller@ppp.com>
--S- Steve Muzniak <stevem@oz.com>
--S- Amanda Walker <amanda.walker@usncad.com>
--S- Peter N Lewis <peter@stairways.com.au>
--S- CME sig <http://www.clark.net/pub/cme/html/pgps-email.html>
--S- Adam C. Ruppe <ac@cs.tulane.edu>
--S- Lloyd Lammert Chambers <lloyd@llc4.com>
--S- Win Treese <treese@openmarket.com>
```

## OpenPGP Certificate (example)

---

---

---

---

---

---

---

---

---

---

## KeyServers

- Where do I get someone else's public key..
  - Ask them for it. (local keychain)
  - Get it from a Keyserver
    - [http <http://pgpkeys.mit.edu:11371>](http://pgpkeys.mit.edu:11371)
    - [ldap <ldap://certserver.pgp.com>](ldap://certserver.pgp.com)
- Keyserver Integrated into PGPkeys

## KeyServers

---

---

---

---

---

---

---

---

---

---

## Meta Certificates ??

- Decentralized Trust Management
  - see <ftp://ftp.research.att.com/dist/mab/policymaker.ps>
- Digitally signed, structured message which delegates an attribute (trust or authority) of some form to a public key
- Simple Public Key Infrastructure SPKI
  - see <http://www.clark.net/pub/cme/spki-reqts.html>

## Meta Certificates ??

---

---

---

---

---

---

---

---

---

---

### Meta Certificate (Example)

- PGPTicket
  - Signed permission slip
  - see [ietf draft-moscariolo-mione-pgpticket-00.txt](#)
- Replaces Users & Groups Database
  - Server mgr signs an authorization
    - \* Allows server login & permissions
    - \* could include expiration date
  - Server only needs to know sys mgrs Public key
  - Scales well
  - server keeps revocation list.

## Meta Certificate (Example)

---

---

---

---

---

---

---

---

### Crypto on the Internet



## Crypto on the Internet

---

---

---

---

---

---

---

---

### Internet threat model

- The internet is an insecure channel
- Assume:
  - *anything* you say is overheard
  - *everything* you say is overheard
- Never send any secure info in the clear
  - "passphrases over telnet"

## Internet threat model

---

---

---

---

---

---

---

---

### Which layer do you encrypt?



## Which layer do you encrypt?

---

---

---

---

---

---

---

---

### Which layer do you encrypt?

- How much granularity of control you want?
- Documents -> **Applications Layer**
  - S/MIME, OpenPGP
- Process or Socket -> **Transport Layer**
  - SSL, TLS
  - e.g. Web Browser to Server
- Host -> **Network Layer**
  - IPSEC
  - e.g. Node to Node

## Which layer do you encrypt?

---

---

---

---

---

---

---

---

### Network Layer Encryption

- Encrypted pipe between:
  - Host to host
  - Roving host (portable) to Firewall
  - aka Virtual Private Network
  - see <<http://www.cygnus.com/~gnu/swan.html>>
- Internet Protocol Security (IPSEC)
  - Encrypted pipe between hosts or firewalls.
  - IPv6 requirement, IPv4 optional
  - see RFC 1825
- Not a new idea
  - used on ARPAnet (early 1970s)
  - see: <<http://www.cygnus.com/~gnu/netcrypt.html>>

## Network Layer Encryption

---

---

---

---

---

---

---

---

## Transport Layer Encryption

- Encrypted pipe between processes
  - Layered on top of reliable transport
  - Most network applications must be modified for support
  - Doesn't handle datagrams (UDP)
- Secure Socket Layer (SSL)
  - Netscape
- Private Communications Technology
  - Microsoft
  - based on SSLv2
- Transport Layer Security (TLS)
  - IETF working group
  - Based on SSLv3
- SSL FAQ:
  - <<http://www.consensus.com/security/ssl-talk-faq.html>>

## Transport Layer Encryption

---

---

---

---

---

---

---

---

## SSL (Misc Notes)

- RSA 40 bit vs 128 bit
  - college student broke 40 bit in 3 hours, with spare cycles on university computers
- SSL is compute intensive
  - Webservers that process large amount of transactions might require cryptographic acceleration hardware.

## SSL (Misc Notes)

---

---

---

---

---

---

---

---

## Applications Layer Encryption

- Secure at document level
- Encrypted or Authenticated files
  - E-mail, text files, www pages

```
-----BEGIN PGP SIGNED MESSAGE-----  
this is a signed message by vinnie  
-----BEGIN PGP SIGNATURE-----  
Version: 5.0 beta  
Charset: noconv  
iQVWwIBAAQAAQF2+rM+HSAQVWP/BEEN3gm9Lz77j8w7d0cae5tqK011  
7at11frubjEw0m0p0d0b5yWskp0m2yEjM1 3lv1H1 57f6dTh1M1 1p8c  
rEYme0y0z2m0p0g10x22mg0j0b0c0c0c0p0y0m0 01k0p01a0w0c0  
3h0z7VEk0g  
=BwE  
-----END PGP SIGNATURE-----
```

## Applications Layer Encryption

---

---

---

---

---

---

---

---

## Crypto in Email

- Two competing IETF WGs
- S/MIME
  - Integrated with Netscape / IE
  - requires X.509 Key
  - some interoperability problems
- OpenPGP
  - Integrated with most email programs
  - No CA required!
  - Highly interoperable

## Crypto in Email

---

---

---

---

---

---

---

---

## Anonymous Remailers

- Provides mail w/o receiver knowing your name or e-mail address
- Allows user to speak w/o fear of reprisal
  - alt.child.abuse.recovery, etc
- Original remailer
  - A script that allowed anonymous posts into <alt.sex.bondage> !
- Cypherpunk remailer
  - attacked by monitoring length of packets
- MixMaster
  - Uses fixed length packets
  - see <<http://www.stack.nl/~galactus/remailers/>>

## Anonymous Remailers

---

---

---

---

---

---

---

---

## Electronic Commerce

"Digital Commerce is Financial Cryptography"  
- Robert Hettinga, 1994



## Electronic Commerce

---

---

---

---

---

---

---

---

### E-Commerce is Financial Cryptography

- Electronic commerce depends on cryptography
  - to make binding commitments
- IDC estimates e-commerce market will be worth > \$200B by 2000
  - But only if systems are very secure

\$\$\$\$\$

## E-Commerce is Financial Cryptography

---

---

---

---

---

---

---

---

### Kinds of E-Commerce

- Book Entries
  - uses encrypted channel to pass debit & credit info
  - requires double entry book keeping via intermediary
- Electronic Cash
  - Cryptographic objects that have monetary value assigned to it.
  - Bearer Certificates
  - Peer to Peer transaction.
  - requires mint or underwriter to issue coins

## Kinds of E-Commerce

---

---

---

---

---

---

---

---

### Book Entry Systems

- Cybercash
  - Encrypted credit cards ... BFD
- First Virtual
  - Requires account, not encrypted, wants to be a bank...
- Mondex
  - Bank puts book entries on smartcard, good for small amounts
- SET
  - Credit card numbers exchanged
- Electronic Checks
  - Financial Service Technology Consortium

## Book Entry Systems

---

---

---

---

---

---

---

---

### Bearer Certificate Systems

- Milicent
  - Site issues values that can only be spent there.
- Digicash
  - Anonymous Digital Cash!!
- Bearer Bonds
  - Application of Digicash Patent

## Bearer Certificate Systems

---

---

---

---

---

---

---

---

### How Secure are Crypto Systems?



## How Secure are Crypto Systems?

---

---

---

---

---

---

---

---

### Bad Design Decisions

- Unauthorized Access to protected info via:
  - Maintenance & Monitor ports
  - Master Keys
  - Mech for Key Escrow & Backup
  - Encryption is not default behavior (people forget to enable it)

## Bad Design Decisions

---

---

---

---

---

---

---

---

---

## Methods for Accessing Info

### Methods for Accessing Info

- Interception in the Ether (radio)
- Wire Tapping
- Traffic Analysis
- Reverse Engineering
- Poor Design Choices
- TEMPEST attack
- Device Penetration ( virus, recording keystrokes)
- Infrastructure Penetration ( routers )

---

---

---

---

---

---

---

---

### Security of Algorithms

- Cryptanalysis
  - The art and science recovering the plaintext of a encrypted message without access to the key.
- Strong vs Weak Crypto
  - An algorithm is **computationally secure** or **strong** if it cannot be broken with **available resources** (both present and future)?
- Rule of Thumb
  - The value of data must remain less than the cost of breaking the security protecting it.

## Security of Algorithms

---

---

---

---

---

---

---

---

### Who can break a crypto system?

- Government Intel agency
  - NSA
- Large Corporations
- Organized Crime
- College students
  - with spare time from a bunch of computers
- distributed.net

## Who can break a crypto system?

---

---

---

---

---

---

---

---



## Internet Algorithm Cracking effort

- DESCHALL
  - Broke 56 Bit DES via brute force keysearch
  - Aprox 14,000 machines
  - Spare CPU time (screen saver)
  - Searched 25% of keyspace in 5 Months (32 days if full time)
  - Peak speed: 7 B keys / sec, 500 Mips years
  - Cost < 10K
  - see <<http://www.frii.com/~rcv/deschall.htm>>
- distributed.net
  - Attacking RC5 (and others)
  - see <<http://www.distributed.net>>

## Internet Algorithm Cracking effort

---

---

---

---

---

---

---

---

---

---

## Cryptographic Attacks

- Passive attacks
  - Cyphertext only attacks
  - Known plaintext attacks
  - Traffic Analysis
- Active attacks
  - Chosen plaintext
  - Man in the middle
  - Rubber hose
- Bypass the Cryptosystem...
  - plant a bug in the room
  - Tempest attack

## Cryptographic Attacks

---

---

---

---

---

---

---

---

---

---

## Snake Oil Cryptography

- Designing secure software is very very hard..
  - most security products can be broken by unfunded attackers
- The sure signs of Snake Oil Crypto
  - "Secret" algorithms - reverse engineering an algorithm is easy
  - "Secret protocols" - see above
  - Misuse of cryptographic terms like "one time pad"
  - Use of term "Unbreakable"
- To avoid Snake Oil Crypto:
  - Use well examined algorithms which don't have known flaws.
  - Publish or make available your protocols and source code for rigorous internal and external review.

## Snake Oil Cryptography

---

---

---

---

---

---

---

---

---

---

## Security holes in MacOS

- **Virtual Memory Backing Store**
  - sensitive data gets swapped out to unsecure disk
- **Temporary Files - Persistent Data Storage**
  - Can still be recovered after being written over up to 9 times.
  - Public guidelines for sanitizing data are no longer valid
    - predate modern recording methods
    - deliberately understated to allow intel orgs to recover data!
  - WRITE TO DISK ENCRYPTED!
- **Wipe Up after yourself**
  - Memory buffers must be properly erased when done
  - DRAM retains traces of data.
- **Using Passphrases**
  - Bullets reveal length info (\*\*\*\*\*)

## Security holes in MacOS

---

---

---

---

---

---

---

---

## Security holes in the MacOS

- **Disk Block Shrinkage**
  - SetEOF doesn't erase data behind itself.
- **Virus or Trojan Horse File Tampering**
  - release digitally signed executables
- **Sending secure info over the net as cleartext**
  - POP, Telnet, NetInfo.
  - Using PGP via Telnet
  - PPC toolbox - cleartext login

## Security holes in the MacOS

---

---

---

---

---

---

---

---

## Crypto and the Government



## Crypto and the Government

---

---

---

---

---

---

---

---

### We are in a Crisis (of Policy)

- Crisis of Policy vs Tech / Industry/ Law Enforcement
  - LEO fear that widespread crypto will impede data collection.
  - Civil Libertarians fear intrusion on private lives of citizens.
- Secrecy is double Edge Sword
  - Has legit basis for National Security
  - Has been used to stifle public debate & conceal bad policies

"Protecting commo against government surveillance is time honored way to defend against tryanny"

## We are in a Crisis (of Policy)

---

---

---

---

---

---

---

---

### History Lesson

- In 1993, NSA figured out that they couldn't stop progress
  - Strong crypto would soon be ubiquitous
- FBI feared that phone taps would no longer work
  - Louis Freeh made his rep by bugging mobsters
- FBI propaganda
  - "Uncrackable encryption will allow drug lords, terrorists, and even gangs to communicate with impunity, ... and will devastate our ability to fight crime and prevent terrorism." -- Louis Freeh

## History Lesson

---

---

---

---

---

---

---

---

### Government response

- Export restrictions on strong Cryptography
- Government want access to private keys

## Government response

---

---

---

---

---

---

---

---

## Export Restrictions on Strong Crypto

Strong crypto is viewed as threat to national security

Unlawful to export (import is OK, for now)

Classified as munition!

International Traffic in Arm Restrictions (ITAR)

Reality:

strong crypto is very available worldwide

Results:

Weakens US corp defenses against foreign intel

Forces US products to be less competitive

Lost jobs for US (est. \$60B by year 2000)

## Export Restrictions on Strong Crypto

---

---

---

---

---

---

---

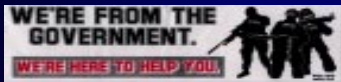
---

---

---

## Government want access to private keys

- Agencies seeking to conduct covert surveillance
  - claim the wide availability of strong crypto technology is a serious threat to law enforcement, public safety and national security.
  - fear that crypto will be used by organized crime, terrorists money launders and child pornographers.
  - want to guaranteed access to encrypted information without the knowledge or consent of owner.
  - insist that it won't be abused or compromised.
  - claim that it can be done safely.



## Government want access to private keys

---

---

---

---

---

---

---

---

---

---

## GAK by any other name

- Government Access to Keys (GAK)
  - aka Key Escrow, Key Recovery, Data Recovery
- For GAK to work, you need:
  - That non-escrowed cryptosystems uninvent themselves.
  - A method external to the cryptosystem to decode encrypted data.
  - A way to transport and store a highly sensitive secret key for an extended period of time.
  - A large scale key management system (no experience)
  - LEA requires high availability (24-7), access to keys within 2 hours, under current regulations.

## GAK by any other name

---

---

---

---

---

---

---

---

---

---

### What about...

- Risks
  - Improper disclosure of keys
  - Theft of keys
- Complexity
  - Strong crypto is hard to design
  - Scale (are you going to store every session key?)
  - Operational complexity
- Costs
  - Operational
  - Product
  - Government Oversight
  - User Costs
- see <[http://www.crypto.com/key\\_study](http://www.crypto.com/key_study)>

### What about...

---

---

---

---

---

---

---

---

### Message recovery vs Key recovery

- Message recovery
  - encryption of session key to multiple recipients
  - corp have legitimate need for this tech
- Key recovery
  - Improperly accessed key can be used to forge signature
  - Requires large scale key management infrastructure
  - Cell phones, FAX, secure web session will require a copy of session key be securely retained until sent to escrow agent.
  - No commercial incentive to develop this technology

### Message recovery vs Key recovery

---

---

---

---

---

---

---

---

### It's really about money (no surprise)

- Encryption on internet & e-commerce.
- Internet based biz move to offshore tax havens
  - Anguilla, FC97
- If flow of money among citizens becomes invisible
  - currency regulations become unenforceable
  - taxes become uncollectable.

### It's really about money (no surprise)

---

---

---

---

---

---

---

---

### Emotional Issue

- Legislators don't use the net
- Knee-Jerk Reaction
  - "Something must be done"
- Freedom is never given away, it is taken
  - Politicians believe that Cops Vote - Techies don't

## Emotional Issue

---

---

---

---

---

---

---

---

### Oh and that pesky document..

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated..

AMENDMENT IV, The Bill of Rights.  
Passed by Congress September 25, 1789

## Oh and that pesky document..

---

---

---

---

---

---

---

---

### Summary



We are almost at the end..

## Summary

---

---

---

---

---

---

---

---

### What have we learned

- Cryptography is more than secrets
- It's also be about **Trust** and **Reputation**.
- Required for E-Commerce
  
- Good Cryptography is hard to do
  - Avoid Snake Oil Crypto
- But shouldn't be hard to use.
  
- **Get out there and Vote!**
  - "Crypto Law is way too important to leave up to legislators"

## What have we learned

---

---

---

---

---

---

---

---

### For more info.

**Applied Cryptography**, Second Edition, Bruce Schneier  
Great review of current protocols and algorithms

**The Code-Breakers**, David Kahn  
History of cryptography

Internet Crypto resources  
<<http://www.vmeng.com/vinnie/crypto.html>>

## For more info.

---

---

---

---

---

---

---

---

# Q&A

## Q&A

---

---

---

---

---

---

---

---

