

# Pretty Good Authentication

Vinnie Moscaritolo

---

---

---

---

---

---

---

## Overview

- Cryptography - more than just secrets.
  - How to use it to *securely* control and access internet services.
- A better way than passwords

---

---

---

---

---

---

---

## Who this talk is for

- (Future) Software developers
- System Administrators
- You??



---

---

---

---

---

---

---

## Background

### Who is Vinnie Moscaritolo?

- Used to be famous..
  - System software engineer, Apple Computer
  - Chief Consulting Engineer, PGP
  - Hosts the Mac-Crypto Workshop
- Not a Cryptographer
- Not a Lawyer
  
- Lots of "real world" security experience

---

---

---

---

---

---

---

---

## The model has changed

Secure Networks → Open Networks  
Insecure Comm. → Secure Comm.

- + Wireless is ubiquitous
- = New threat model

---

---

---

---

---

---

---

---

## Internet threat model

- The internet is an insecure channel
  
- Assume:
  - *anything* you say is overheard
  - *everything* you say is overheard

---

---

---

---

---

---

---

---

## Attacks to Network Services

- Packet Sniffing
- Automated Password Guessing
- Replay Attacks
- Session Stealing
- Infrastructure Penetration
- Device Penetration
- Social Engineering & Rubber Hose

---

---

---

---

---

---

---

---

## Packet Sniffing

- Packet sniffing SW is widely available.
- Clear-text passwords are common.
  - POP
  - FTP
- Wireless makes this much easier
  - AirSnort

---

---

---

---

---

---

---

---

## Automated Password Guessing

- Brute force vs dictionary attacks
- Online attacks
  - Easily detectable
- Offline attacks
  - Targets password databases
  - Accessed through other holes (cgi)
  - Many utilities available for cracking /etc/passwd

---

---

---

---

---

---

---

---

## Replay Attack

- Capture previous session
- Replay later.

---

---

---

---

---

---

---

---

## Session Stealing

- Wait for user to initiate login.
- Denial of service attack to client
  - Forge TCP reset, closes clients connection
- Hijack already authenticated session
  - (with victims authentication & privileges)

---

---

---

---

---

---

---

---

## Infrastructure Penetration

- Target name-servers or routers
  - Force reload with infected sw
- Initiate Man-in-the-middle attack
  - User notices no loss of service
  - Attacker monitors all traffic (even encrypted)

---

---

---

---

---

---

---

---

## Device Penetration

- Virus or Trojan Horse
- Keystroke capture
- Spoofed downloads
  - Sign your distributions!

---

---

---

---

---

---

---

---

## Social Engineering & Rubber Hose

- People are weakest link.
  - Easily fooled, coerced or intimidated.
  - Shoulder surfing
- Difficult to defend against
  - Requires management acknowledge the threat, and support threat awareness education for users.

---

---

---

---

---

---

---

---

## Cryptography

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



---

---

---

---

---

---

---

---

## 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

---

---

---

---

---

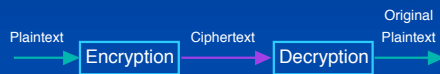
---

---

---

## How does Cryptography Work?

processes message with an **algorithm** or **cipher**



If no key is involved,  
then the **algorithm must be kept secret**.  
(Security through Obscurity)

---

---

---

---

---

---

---

---

## Encryption without a Key

- Problems with **secret algorithms**:
  - Keeping it a Secret
  - Requires distribution via secure channel
  - Easy to reverse engineer
  - Lack of peer review

---

---

---

---

---

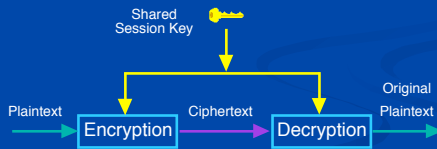
---

---

---

## 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.



---

---

---

---

---

---

---

---

## Key vs Passphrase

- Key
  - Must be random
  - All keys must be equally probable
- Passphrase
  - Must be easy to remember
  - Smaller keyspace
  - Words are vulnerable to Dictionary attack

KeySpace	4 Byte	8 Byte
Lowercase Letters (26)	460,000	$2.1 \cdot 10^{11}$
Lowercase Letters & digits (36)	1,700,000	$2.8 \cdot 10^{12}$
Alphanumeric characters (62)	$1.5 \cdot 10^7$	$2.2 \cdot 10^{14}$
Printable characters (95)	$8.1 \cdot 10^7$	$6.6 \cdot 10^{15}$
ASCII characters (128)	$2.7 \cdot 10^8$	$7.2 \cdot 10^{16}$
8 bit ASCII characters (256)	$4.3 \cdot 10^9$	$1.8 \cdot 10^{19}$

---

---

---

---

---

---

---

---

## 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



---

---

---

---

---

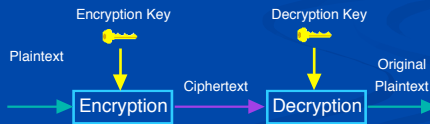
---

---

---

## 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



---

---

---

---

---

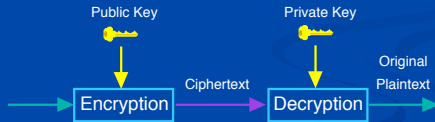
---

---

---

## Public 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**.



---

---

---

---

---

---

---

---

## Cryptography is not just about secrets

- Cryptography is also about trust and reputation.
- Use **digital signatures** for
  - Authentication
  - Integrity
  - Nonrepudiation

---

---

---

---

---

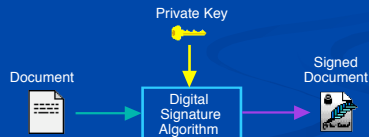
---

---

---

## 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



---

---

---

---

---

---

---

---

## 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



---

---

---

---

---

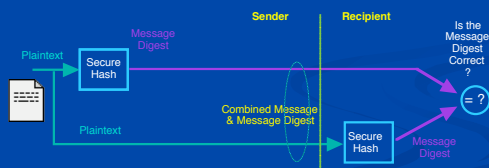
---

---

---

## 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.



---

---

---

---

---

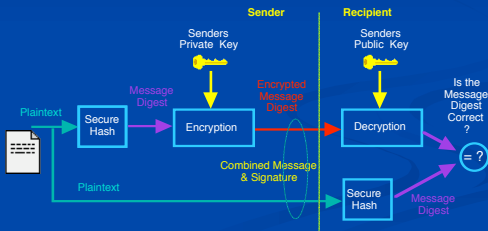
---

---

---

## Creating a Digital Signature w Encryption

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



---

---

---

---

---

---

---

---

## 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

---

---

---

---

---

---

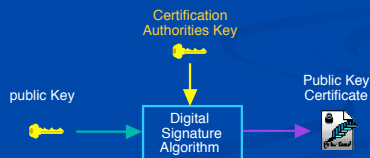
---

---

## 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**.



---

---

---

---

---

---

---

---

## What kinds of Certificate?

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

---

---

---

---

---

---

---

---

## Authentication

Who are you?

---

---

---

---

---

---

---

---

## User Authentication Methods

- Local Authentication
  - Authentication material never exits user's control
  - e.g. Mounting local a PGPDisk volume
- Remote Authentication
  - "A secret shared, isn't."
  - e.g. remote server password

---

---

---

---

---

---

---

---

## Authentication Methods

- Something one has.
  - Something one knows.
  - Something one is.
- Or a combination of the above

---

---

---

---

---

---

---

---

## Something one has

- Hardware token
  - Personal, mobile & convenient
  - Corp Badge, ATM card, Car Keys
- Passive = Key storage
- Active = On-board crypto, Key never leaves device
- Hardware tokens are subject to theft.
  - Combine with password or biometric.

---

---

---

---

---

---

---

---

## Something one is.

- Biometrics
  - Fingerprints, Retina scans, Voice recognition
    - Records measurement of human traits and later compares to a stored template.
  - Subject to Replay Attack
  - Fuzziness is unsuitable for key storage
  - Returns (True or False)
  - Combine with password or biometric.

---

---

---

---

---

---

---

---

## Something One Knows

- Secret Password, PIN,
- Oldest form of authentication
- Easiest method to breach

---

---

---

---

---

---

---

---

## What's wrong with Passwords ?

- Passwords in transit are subject to sniffing & replay attacks.
  - Never send passwords in clear-text (use APOP, SPEKE, etc)
- Simple passwords vulnerable to dictionary attack
- Complex passwords are difficult for user to manage.
  - Vulnerable to social engineering
- Remotely stored passwords are out of user's control.
  - Can be attacked at server.
  - "A secret shared, isn't."

---

---

---

---

---

---

---

---

## Too much to remember.

- Most IS policies require that passwords:
  - Complex variation of alpha, numeric or punctuation
  - Change periodically.
- Limitation of human memory
  - "Unrealistic to expect that users will reliably memorize more than one or two passwords"
  - Typically written down in convenient location
- Single sign-on to the rescue

---

---

---

---

---

---

---

---

## Single Sign-on Systems

- User authenticates to proxy
  - "Gives authority to negotiate all subsequent authentication to remote services autonomously"
- Password Caches & Keychains
- Remote Systems. (ldap)
- Kerberos

---

---

---

---

---

---

---

---

## Password Caches & Keychains

- Intercepts server logins & records passwords
- Issues:
  - Key database must be kept synced across multiple machines.
  - Database file must be strongly encrypted
  - No guarantee that file server login isn't in clear-text.
  - API needs to prevent rogue export of passwords
  - Integration with multi-factor systems can be awkward.

---

---

---

---

---

---

---

---

## Multi Factor Systems

- Combine password, biometric or token
- Most secure method
- Requires separate attacks on each method.
- E.g.: SecurID
  - Requires servers in physically secure locations
  - Open to other attacks
    - See <http://www.homeport.org/~adam/dimacs.html>
    - Major PITA to use daily..!

---

---

---

---

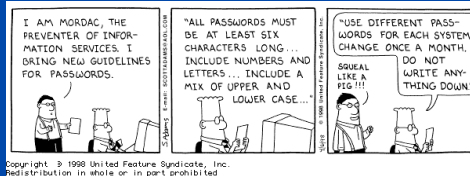
---

---

---

---

## Getting Beyond Passwords



---

---

---

---

---

---

---

---

## Beyond Passwords

- Provide Single Sign-On experience
- Strong user authentication
- No dependency on trusted servers
- A compromised server, doesn't effect others.
- Builds on existing infrastructure
- Scales to large user base.

---

---

---

---

---

---

---

---

## Authentication with Cryptographic Signatures

- Public Key Cryptography
  - Holder of private-key is only entity that can sign.
  - Holder of public-key can verify signature.
- "Public key functions as principles identity in cyberspace"

---

---

---

---

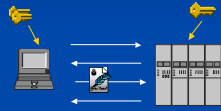
---

---

---

---

## Cryptographic Challenge - Response



1. Client **requests** access.
2. Server **generates** random challenge string.
3. Client **signs** challenge with private key
4. Server **verifies** signature with public key & **grants or denies** access.

---

---

---

---

---

---

---

---

## Why Crypto Authentication?

- Same key is also used to sign e-mail
  - User has only one passphrase to remember.
  - Existing key management infrastructure
- Strong user authentication.
  - Expensive Crypto operations are only occur once.
  - Random challenge prevents replay attack
- User maintains all secret material
  - Compromised server results in limited damage

---

---

---

---

---

---

---

---

## PGPuam

- AppleShare User Authentication Module
- Apple DTS Sample Code (CW3 C, C++)
- AppleShare client 3.8.1
- AppleShare IP 6.1
- PGPsdsk 1.5



---

---

---

---

---

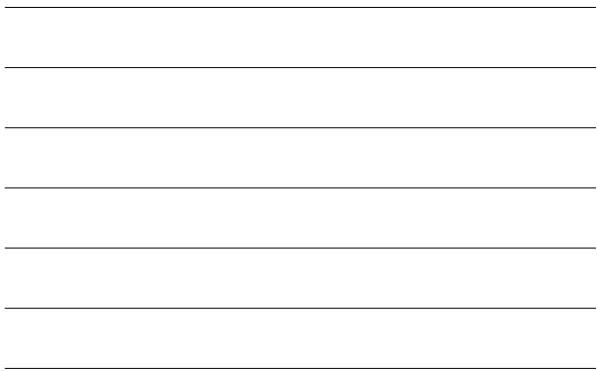
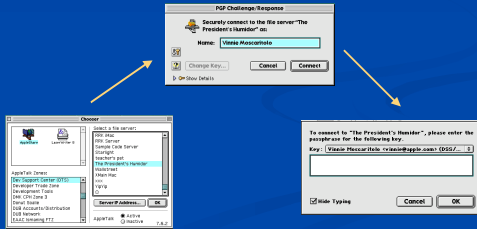
---

---

---

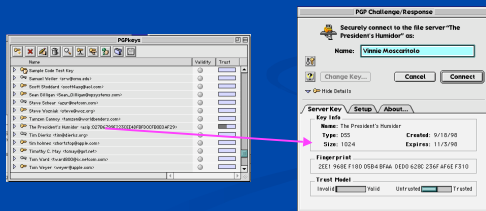
# PGPum

Enables users to securely connect to AppleShare IP servers



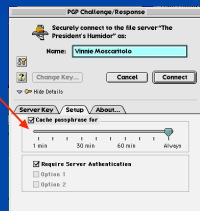
# PGPum

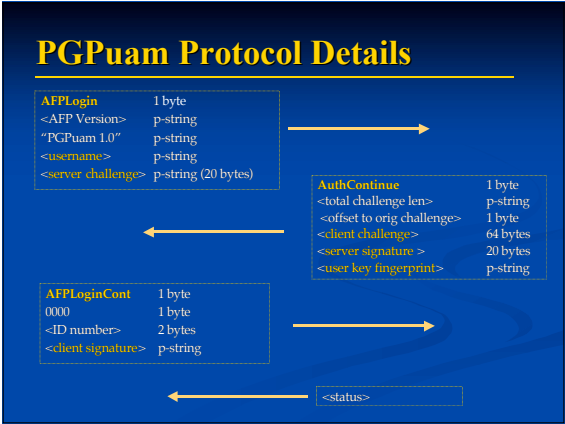
- Two way authentication
  - Client has copy of server public key



# PGPum

- Single Sign-On






---

---

---

---

---

---

---

---

---

---

- ## PGPuam design decisions
- PGPsdK
    - Leverage existing key infrastructure
    - Trust model not critical
  - Random challenge / counter challenge
    - Prevent *sign-this* attack
  - Sign Only
    - Export Control issues

---

---

---

---

---

---

---

---

---

---

- ## What about Authorization?
- PGPuam just authenticates user.
  - Requires a permission database.
  - Bootstrapping the keys to server.

---

---

---

---

---

---

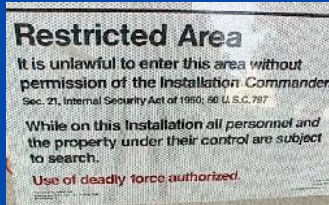
---

---

---

---

## Authorization



---

---

---

---

---

---

---

---

## Centralized Authorization

- LDAP, Kerberos
- Auth server must be physically secure
- Must be in communications with servers
- Weakest point in system
  - Ideal target
  - Open to denial of service attack

---

---

---

---

---

---

---

---

## Decentralized Authorization

- Capabilities based system
  - Simple Public Key Infrastructure (SPKI)
  - Policymaker
- Authorization Certificate  
(permission slip)

---

---

---

---

---

---

---

---

## SPKI Authorization Certificate

- **Issuer**
  - Generates and signs certificate
- **Subject**
  - Principle(s) which cert grants authorizations.
- **Validity**
  - Dates or tests specifying conditions or validity period
- **Authorization**
  - Principle's access permissions
- **Delegation**
  - Can a principle pass on these rights to others?

---

---

---

---

---

---

---

---

## SPKI Example:

- **Issuer**
  - 0AAF 9A74 113F 7FDD 97E6 04BD 1D7A 2D86 6F17 2A0A (Vinnie Moscaritolo)
- **Subject**
  - 7E83 D31E 9E25 FF26 9167 51B4 615B 681F 2C54 C8FA (Leland Wallace)
- **Validity**
  - Tues Oct 6 1998, 15:05:58
  - Fri Oct 9 1998, 16:05:58
- **Authorization**
  - (tag (pkpfs://afp.vmeng.com/pub/vinnie/file4 (read, write))
- **Delegation**
  - False



SPKI is now XML based!

---

---

---

---

---

---

---

---

## PGPticket

- OpenPGP v4 standalone signature packet

- Issuer: Key Id
- Subject(s): (Key Id, Algorithm Id, Key FP)
- Validity: Creation/Expiration time
- Auth: p-string (XML)
- Delegation: Boolean

---

---

---

---

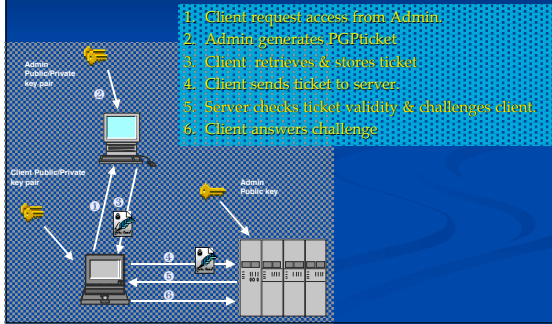
---

---

---

---

## PGPticket in action



---

---

---

---

---

---

---

---

---

---

## Issuing PGPtickets

- Out of band:
  - Admin must verify client's key through standard practice.
  - Ticket can be returned via:
    - Private clear-text mail
    - Posted on website or newsgroup
    - Placing on website and sending URL to client.
- In Band request:
  - Automatic??

---

---

---

---

---

---

---

---

---

---

## Storing PGPtickets

- Client needs a ticket/service database.
- No need to encrypt ticket
- Could be posted on public server
  - Mobile access
  - Cached by server

---

---

---

---

---

---

---

---

---

---

## Using PGPtickets

- Server challenges client to prove authenticity.
  - Doesn't need to possess client key ahead of time, since client fingerprint is signed into ticket.
- Two way challenge. (same as PGPuam)
- Tickets can be identified by hash (ticket id)
- Ticket Revocation vs. Key Revocation
- Subjects can be groups

---

---

---

---

---

---

---

---

## Advantages

- Server only needs copy of root public-key.
- No certificate authority (CA) needed
- Auto expiring access
- No dependency on trusted servers
  - Remote access

---

---

---

---

---

---

---

---

## Applications:

- File & Web Server
  - Paying Anon access.. XXX
- Firewalls & VPN
- Wireless Access Points
- Screen sharing
- Automatic expiring access.
- Combine with Digital Cash!

---

---

---

---

---

---

---

---

## Implementations

- IETF Internet-draft
- Code:
  - PGPTicketLib
    - Built on PGP SDK
    - Source code available (ask me)
  - PGPTicketMgr
    - Handles protocols.
    - Unfinished.

---

---

---

---

---

---

---

---

## Futures:

- Ticket Management Code
- Auto Ticket Renewal
- Ticket Revocation
- Anon Users

---

---

---

---

---

---

---

---

## Summary



We are almost at the end

---

---

---

---

---

---

---

---

## What have we learned

- Internet service are vulnerable to attack
  - Passwords are no longer safe
- Cryptography is more than just secrets
  - Can be used for both authentication & authorization.
  - Can make thing easier

---

---

---

---

---

---

---

---

## For More Info

- PGPticket, PGPuam
  - <http://www.vmeng.com/vjinnie/pubs.html>
- PGP sdk
  - <http://www.pgp.com/>

---

---

---

---

---

---

---

---

## Q & A

---

---

---

---

---

---

---

---