



# PGPUAM

## Public Key Authentication for AppleShare

Vinnie Moscaritolo  
Apple Computer, Inc

---

---

---

---

---

---

---

---

### Overview

- Existing User Authentication Methods
  - ◆ Common attacks & weaknesses
- Getting beyond passwords
  - ◆ Cryptographic signatures
- PGPUAM
  - ◆ Enhancing AppleShare authentication

---

---

---

---

---

---

---

---

### Who this talk is for

- System Administrators
- Security conscious users
- Mac OS developers



---

---

---

---

---

---

---

---

## See also

- AppleShare Authentication Architecture  
(Weds)
- PGPTicket - A Secure Authorization Protocol  
(Thurs)

---

---

---

---

---

---

---

## Background

### Who is Vinnie Moscaritolo?

- ◆ Apple Developer Services
- ◆ (formerly Chief Consulting Engineer, PGP)
- ◆ Hosts the Mac-Crypto Workshop
  
- ◆ Not a Cryptographer
- ◆ Not a Lawyer
  
- ◆ Lots of "real world" security experience
- ◆ <<http://www.vmeng.com/vinnie>>

---

---

---

---

---

---

---

## What has changed?

Secure Networks    →    Open Networks  
Insecure Comm    →    Secure Comm

= New threat model

---

---

---

---

---

---

---

## 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.
- Cleartext passwords are common.
  - ◆ POP
  - ◆ FTP
  - ◆ PPC Toolbox

---

---

---

---

---

---

---

---

## 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 & privs)

---

---

---

---

---

---

---

## 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.

---

---

---

---

---

---

---

## 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 recog
    - ◆ Records measurement of human traits and later compares to a stored template.
  - ◆ Subject to Replay Attack
  - ◆ Fuzzyness 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 corp. 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. (Idap, Jade)
- Kerberos

---

---

---

---

---

---

---

## Password Caches & Keychains

- Intercepts server logins & records passwords
- Introduced in PowerTalk (sys 7.1.1)
- 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 rouge export of passwords
  - ◆ Integration with multi-factor systems can be awkward.

---

---

---

---

---

---

---

## Kerberos

- Popular among higher Ed
- Based on Secret Key encryption
- Depends on trusted servers
  - ◆ Requires physically secure location
  - ◆ Synchronized clocks
- “Inappropriate for small biz or large scale Internet deployment”.

---

---

---

---

---

---

---

## Multi Factor Systems

- Combine password, biometric or token
- Most secure method
- Requires separate attacks on each method.
- Eg: SecurID
  - ◆ Requires servers in physically secure locations
  - ◆ Open to other attacks
  - ◆ See <ftp.secnet.com/pub/papers/securid.ps>

---

---

---

---

---

---

---

---

## 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

---

---

---

---

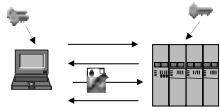
---

---

---

---

## Cryptographic Challenge - Response



- Client requests access.
- Server generates random challenge string.
- Client signs challenge with private key
- 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 OK
  - ◆ Random challenge prevents replay attack
- User maintains all secret material
  - ◆ Compromised server results in limited damage

---

---

---

---

---

---

---

---

## PGPUAM

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



---

---

---

---

---

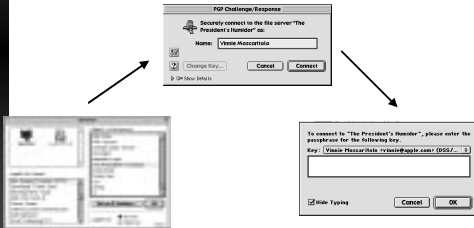
---

---

---

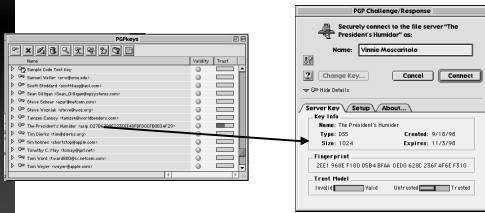
# PGPUAM

Enables users to securely connect to AppleShare IP servers



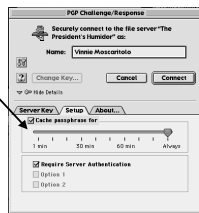
# PGPUAM

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



# PGPUAM

- Single Sign-On



## Design Decisions

### ■ PGPsdk

- ◆ Leverage existing key infrastructure
- ◆ Needs to work at deferred task time
- ◆ Trust model not critical

### ■ Random challenge / counter challenge

- ◆ Prevent sign-this attack

### ■ Sign Only

- ◆ Export Control issues

---

---

---

---

---

---

---

---

## Protocol Details (login)

FPLogin	1 byte	
<AFP Version>	p-string	
"PGPUAM 1.0"	p-string	→
<username>	p-string	
<challenge>	p-string (16-20 bytes)	

	AuthContinue	1 byte
	<total challenge len>	p-string
←	<offset to orig challenge>	1 byte
	<counter challenge>	64 bytes
	<sig>	20 bytes
	<user key fingerprint>	p-string

---

---

---

---

---

---

---

---

## Protocol Details (login cont)

FPLoginCont	1 byte	
0000	1 byte	→
<ID number>	2 bytes	
<signature>	p-string	

←	<status>
---	----------

---

---

---

---

---

---

---

---

# DEMO

---

---

---

---

---

---

---

---

## Learning Experience

- Getting keys to server (bootstrapping)
- PGPsdk improvements
  - ◆ Working with raw key material
  - ◆ Key database

---

---

---

---

---

---

---

---

## What's next?

- Server Manager
  - ◆ User Interface
- Security
  - ◆ Prevent Session Stealing, HMAC
  - ◆ Encrypt sessions
  - ◆ Macsbug attack of passphrase cache
- PGPticket
  - ◆ Authorization Certificates

---

---

---

---

---

---

---

---

## Summary

- Cryptography is more than secret messages

---

---

---

---

---

---

---

## For More Info

- PGP/PGP  
◆ <http://www.vimeng.com/vinnie/pubs.html>
- PGP/PGP  
◆ <http://www.pgp.com/sdk/>
- Appleshare IP  
◆ <http://www.apple.com/appleshare/>

---

---

---

---

---

---

---

## Q & A

---

---

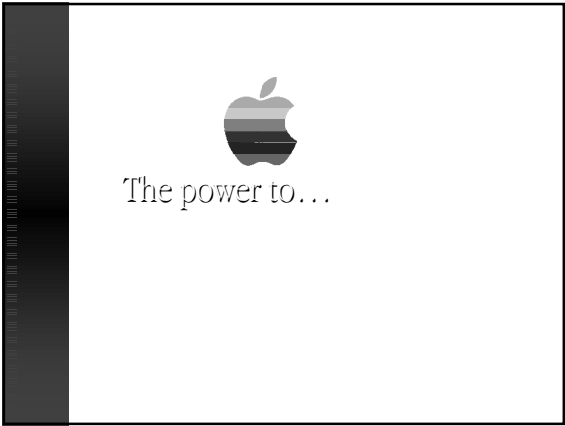
---

---

---

---

---



---

---

---

---

---

---

---