

# *Network Security and IPsec*

## *Part 2*

*Angelos D. Keromytis*  
*University of Pennsylvania*

*John Ioannidis*  
*AT&T Labs - Research*

# Key Management - The Need

- Manual/Static Keying
  - Tedious
  - Prone to Misconfiguration
  - Requires Significant Human Intervention
  - Typically Weak Keys
  - Does not Scale
  - Boring
- Key Establishment Needs Automation

# Key Management - Requirements

- Negotiate SA Parameters
  - Capabilities vs. Desired Configuration
- Establish Strong Keys
- Dynamic Rekeying
  - Increased Security
  - Crash Recovery
- Require Minimal Configuration
  - No Human Intervention

# Key Management - Requirements

- Algorithm Independence
- Identity Protection
- Forward Secrecy If Needed
- Fast
- Scalable
- Cure Cancer
  
- Simplicity Not A Requirement!

# Key Management - Situation

- ISAKMP
- Oakley
- Internet Key Exchange (IKE)
- Photuris
- SKIP
- KINK
  
- Will Focus on IKE

# ISAKMP

- Framework for Writing Security Protocols
  - Standardized Payloads
  - Exchange Types
  - Payload Processing Rules
  - Flexibility
- Domain Of Interpretation Concept

# Internet Key Exchange (IKE)

- Combination of ISAKMP and Oakley
- Uses UDP (port 500)
- Two Phase Protocol
  - Establish Secure Channel
  - Authenticate Peers
  - Negotiate Application Parameters
- Various Authentication Mechanisms
- Various Key Agreement Mechanisms
  - Diffie-Hellman
  - Kerberos (W2K)

# Diffie-Hellman Algorithm

- Small Integer  $g$ , Generator for  $p$  (512+ bits)
- Alice Creates Random  $x$ 
  - Computes  $y = g^x \text{ mod } p$
- Bob Creates Random  $x'$ 
  - Computes  $y' = g^{x'} \text{ mod } p$
- Exchange of Computed Values ( $y, y'$ )
- Alice Computes (Similar for Bob)
  - $z = y'^x \text{ mod } p \Rightarrow$
  - $z = (g^{x'} \text{ mod } p)^x \text{ mod } p \Rightarrow$
  - $z = g^{(x * x')} \text{ mod } p$

# Diffie-Hellman Observations

- Used Over Insecure Link
- Derived Shared Secret Used As Key
- Eavesdropper Cannot Find  $z$
- Vulnerable to Active Attacker
  - Man In The Middle Attack
- Authentication Needed
  - Closely Tied to Keying!

# IKE Negotiations

- Initiator-Driven
  - Propose Set Of Parameters
  - Responder Picks One Or More
    - Cannot Counter-Propose
- Capability Discovery And Policy Combined
  - Heavy Dependence On External Policy
- Potential Move Towards Profiles
- Work In Auto-Configuration

# IKE Phase 1 - Negotiation Details

- Lifetime of Phase 1 SA
- Key Agreement Mechanism
- Encryption/Hash Algorithms
- Authentication Algorithm
  - Preshared Secret
  - Public Key Signature
  - Public Key Encryption

# IKE Authentication

- What Is Being Authenticated ?
  - Directly
    - Hosts, "Users"
    - Liveness
  - By Implication
    - Capabilities
- Possession Of Secret
  - Passphrases, RSA key ...
- Heavy Policy Interdependencies

# IKE Phase 1 - Authentication (1)

- Preshared Secret
- Passphrase-based
- Non-sniffable
- Dictionary Attack Possible
- Easy for Testing
- Scalability Problems
  - Requires Agreement

# IKE Phase 1 - Authentication (2)

- Public Key Signatures
- Various PK Algorithms (RSA/DSA)
- Certificate-based Verification
  - X.509/PKIX Most Common
  - PGP, KeyNote, ...
- Requires Infrastructure
  - Provisioning
  - Revocation
- Support From Bad to Nonexistent
  - Proprietary Systems

## IKE Phase 1 - Authentication (3)

- Public Key Encryption
  - Similar Issues to Signatures
- Requires Initiator Knows Responder PKey
- Limited Support
- No Clear Advantages Over Signatures

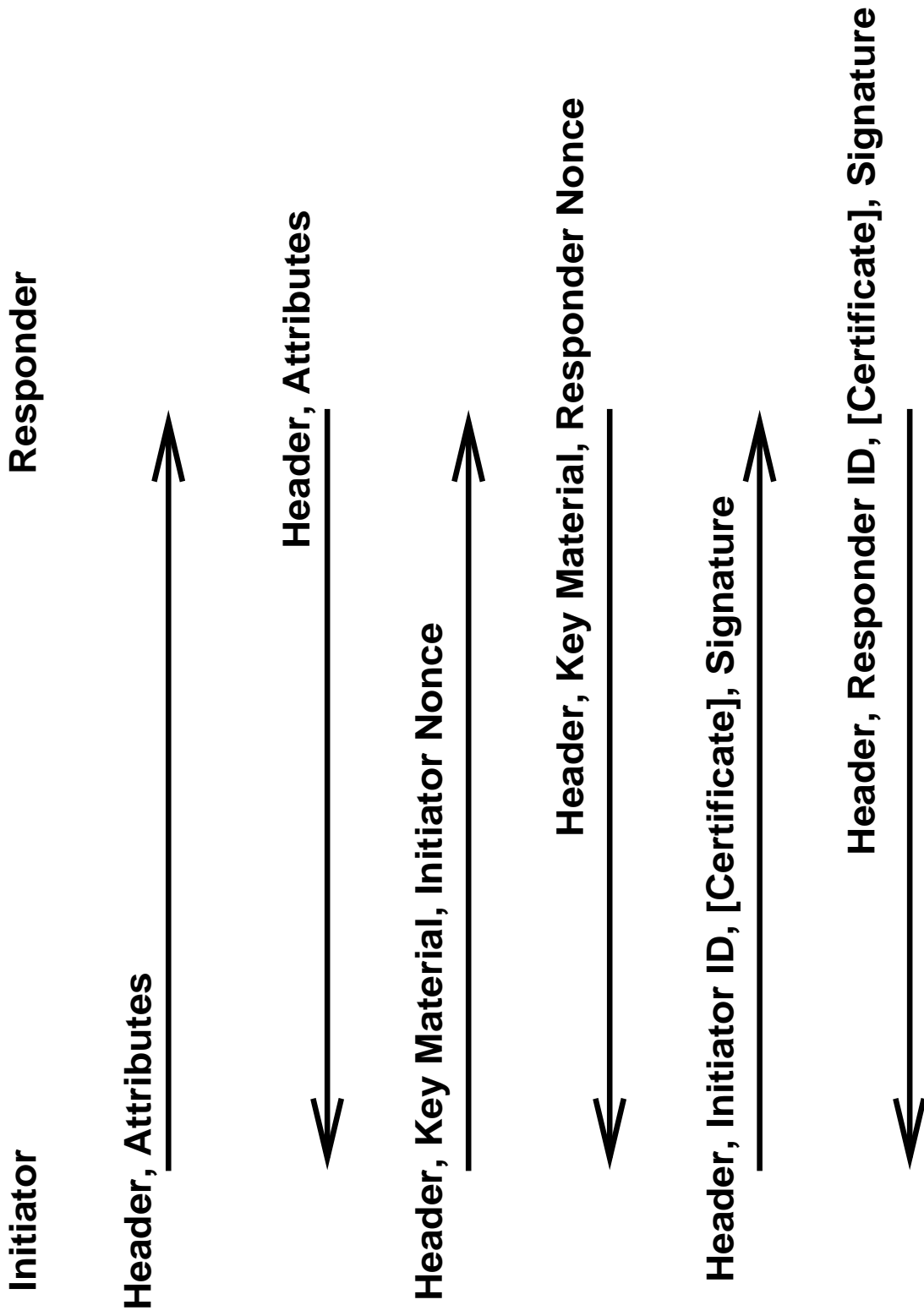
# PKIX

- Attempt To Solve Authentication Problem
  - Trusted Third Parties (CAs)
  - Issue Certificates To "Users"
  - Users Verify Others' Certificates
- Scalability
- Global PKI Improbable
  - Consistently Failed Deployment
- Organizational PKIs
  - Web Of Trust
    - Arbitrarily Complex

# PKI - The Practical Side

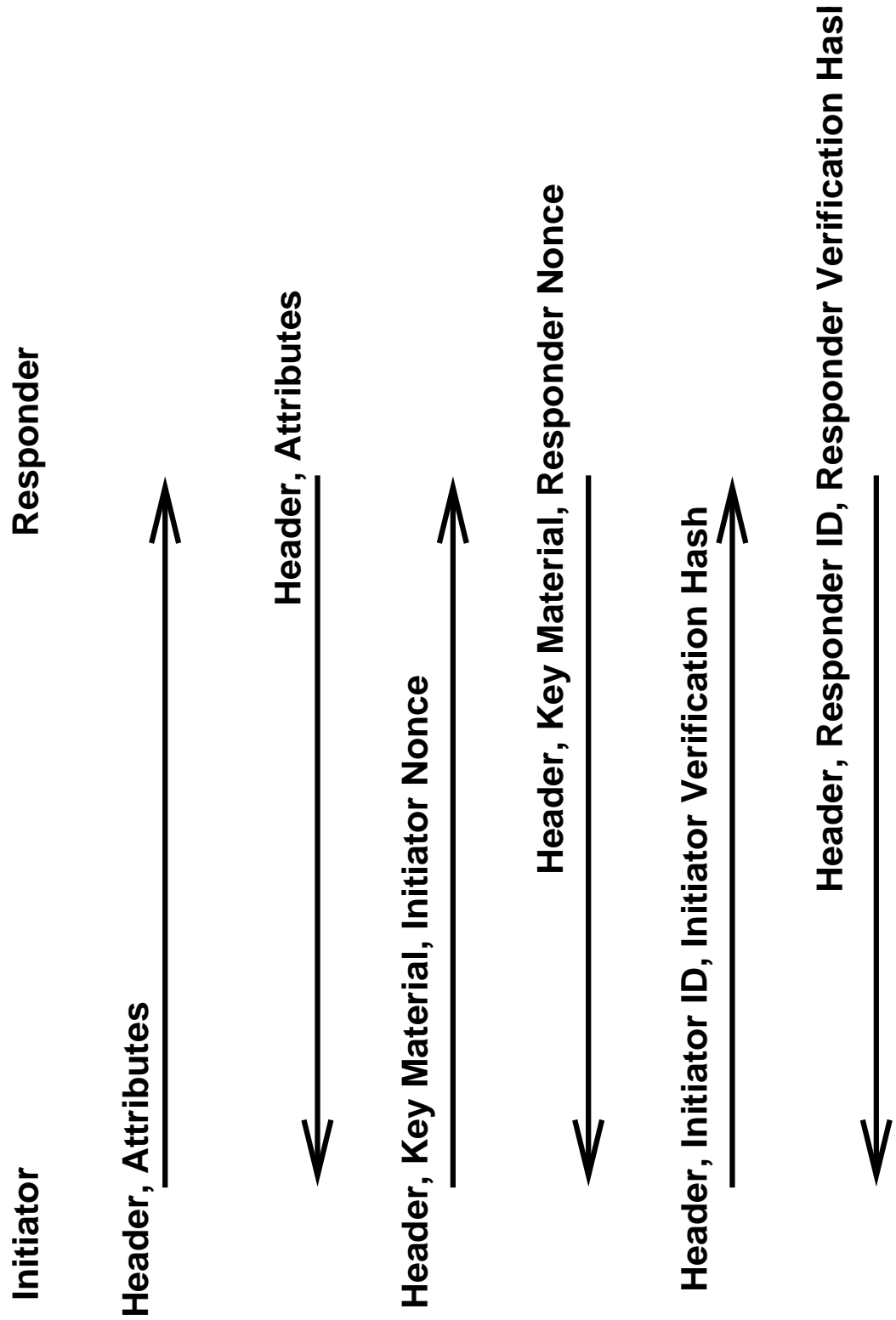
- Oversold By Vendors/Marketing
- Flexible, Powerful (In Theory)
- In Practice
  - Proprietary Solutions
  - Insufficient Integration
  - Lack Of Support
  - Dubious Support Of Standards
- Be Careful

# IKE Main Mode - Signatures



The payloads on the last two messages (except for the Header are encrypted.

# IKE Main Mode - PassPhrase



The payloads of the last two messages (except for the Header) are encrypted.

# IKE - Aggressive Mode Signature

Initiator

Responder

Header, Attributes, Key Material, Initiator Nonce, Initiator ID



Header, Attributes, Key Material, Responder Nonce,

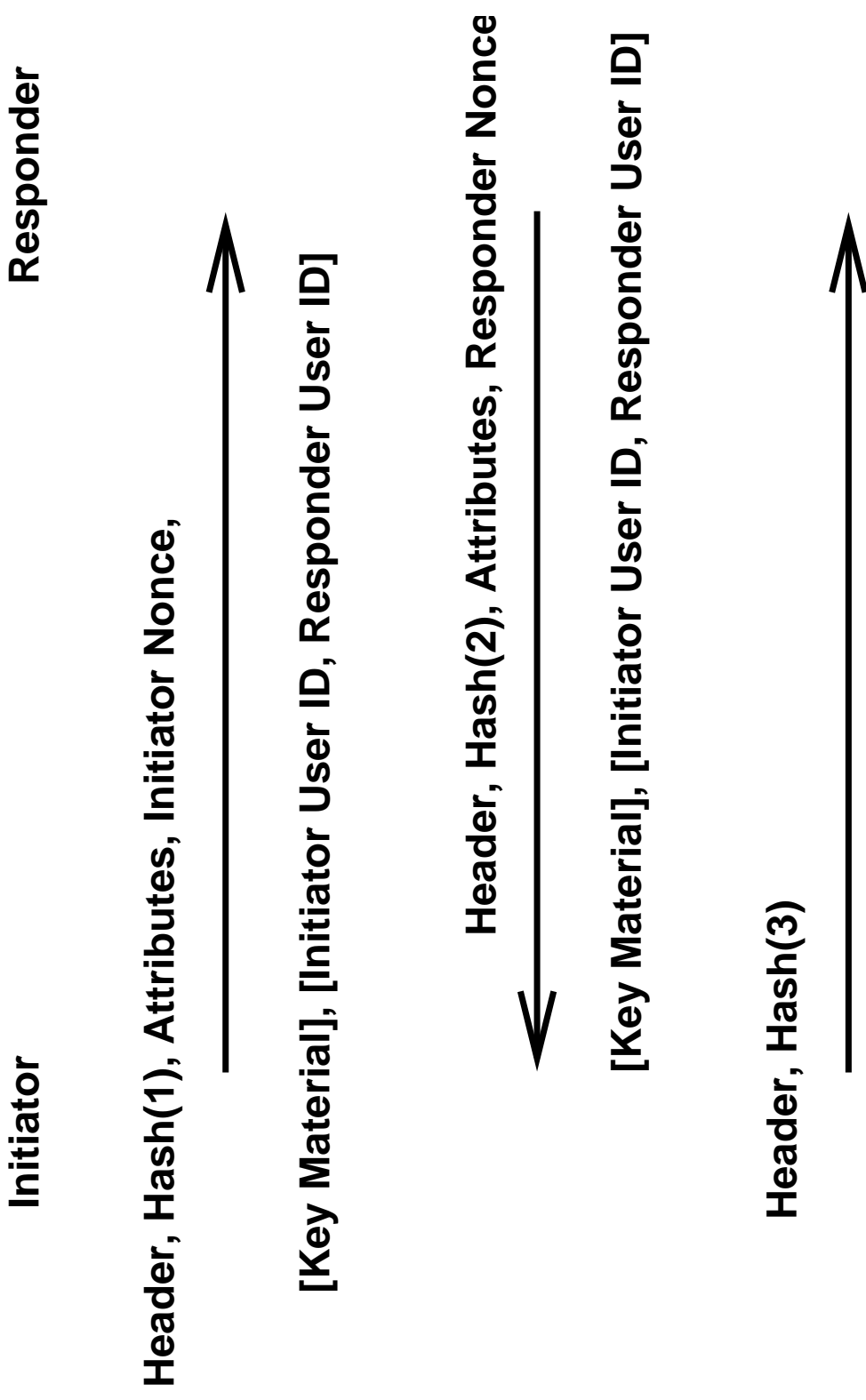


Responder ID, [Certificate], Signature

Header, [Certificate], Signature



# IKE Phase 2 - Quick Mode



**All payloads (except for the Headers) are encrypted.**

# IKE Informational Exchanges

- Mostly Used for Error Notification
- Some Proposals for
  - Dead Peer Detection
  - Compatibility/Versioning
- Nothing Exciting Here
- Other Modes Exist (New Group)

# Problems With IKE

- **Complicated**
  - **Feature Bloat**
  - **Obscure Documentation**
- **Interoperability Problems**
- **Denial of Service Problems**
  - **Bad Cookies**
- **Other Issues**
  - **NAT, Firewalls**

# Photuris

- Simple Key Agreement Protocol
- Adaptation of Station-to-Station Protocol
- Introduced Concept of Cookies
- Less Extensible
- Cleaner Specification
  - Cleaner Implementations
- Can Be Used Simultaneously With IKE
- Limited Use/Availability

# Policy

- Different Aspects/Layers
  - SPD
  - Attribute Negotiation
  - Domain Traversal
- Some Policy Exchanged In IKE
  - Insufficient For Most Scenarios
- IETF IPSP Working Group
  - Integrate Aspects
- Very Early Stages

## Related Issues

- PF\_KEY
- Socket API
- DNS Security (DNSSEC)
- IPSRA
- Son of IKE
  
- SSH
- SSL/TLS
- SKIP
- Kerberos

# PF\_KEY

- Kernel Interface to Key Management
- Message-oriented Protocol
  - Based on BSD Routing Sockets
- Features:
  - Multiple Listeners
  - Different “Applications”
  - Kernel/User Initiated (Re)Keying
- Allows Portability in KMPs
- Does NOT Handle SPD
  - Hacked to do so

# IPSec Socket API

- Applications Can Take Advantage Of IPsec

## Directly

- Minor Layer Violation
- Complementary To PF\_KEY
- No Standard Way
  - Typically `getsockopt()/setsockopt()`
- Close Ties To Policy
- More In The Future

# DNS Security

- Protect DNS Information
  - Use Public Key Signatures
  - Existing Certification Structure
- Can Use Same Mechanism to Distribute

## Public Keys

- Allows Real “Opportunistic IPsec”
- Very Limited Support
  - Few Compliant Servers Running
  - Even Fewer Resolvers Support it

◦ Probably a Few Years Away Still

# IPSRRA

- Extend IKE to Provide Support for:
  - Legacy Authentication Systems
    - RADIUS, SecureID, etc.
  - Asymmetric Authentication
  - Configuration/Provisioning
    - Internal Addressing (DHCP!)
- Unclear If/When/What Will be Delivered

# Son of IKE

- Effort to Simplify IKE
- Potential Targets:
  - Keep Only Identity Protection
  - Remove Public Key Encryption
- Merge/Simplify Documentation
- A Year or Two Until Standardized (?)

# Kerberos

- Some User Base
- W2K Uses Tickets For IKE Authentication
  - Not Much Interoperability
- KINK Based on Kerberos
  - Use an Existing Secure Channel
  - Distribute IPsec Keys/Parameters
  - Very Recently Started

# SSH

- Application Layer Protocol
- Replacement for r- utilities
  - Port/X Forwarding
  - PPP Over SSH Tunnels
- Mainly Intended For Interactive Use
- Not As Flexible As IPsec
  - Mostly Because of Different Layer
- Can Work With IPsec

# SSL/TLS

- Application Layer Protocol
- Three Sub-protocols:
  - Handshake (everything except..)
  - Change CipherSpec (coordination)
  - Alert (errors)
- Mostly Used For Web Transactions
  - Session Caching -- Reuse

# SSL/TLS (2)

- Various Key Exchange Methods
  - RSA-encrypted Key Exchange
  - Fixed Diffie-Hellman
  - Ephemeral Diffie-Hellman (StS)
  - Anonymous Diffie-Hellman
    - No Authentication
- Wide Mix Of Security Trade-Offs

# SSL/TLS (3)

- Conclusions:
  - Flexible, Adaptable
  - Used in New Technologies (e.g., WAP)
  - Wrong Layer
  - TCP Dependence
    - May Be Addressed

# SKIP

- Older Key Management Proposal for IPsec
- Keying Occurs In-Band
  - Protocol Inside IP
  - Encrypted Session Key in Every Packet
  - Faster Session Setup
  - Overhead Per Packet
  - Does Not Provide Forward Secrecy
  - Good For Multicast (?)
- Not Widely Used

# Comparisons ? (1)

- Different Technologies
  - Different Concerns
- IPsec
  - Powerful
  - Transparent
  - Network Layer
- Requires Infrastructure Changes
- SSH
  - User Tool
  - Convenient Features

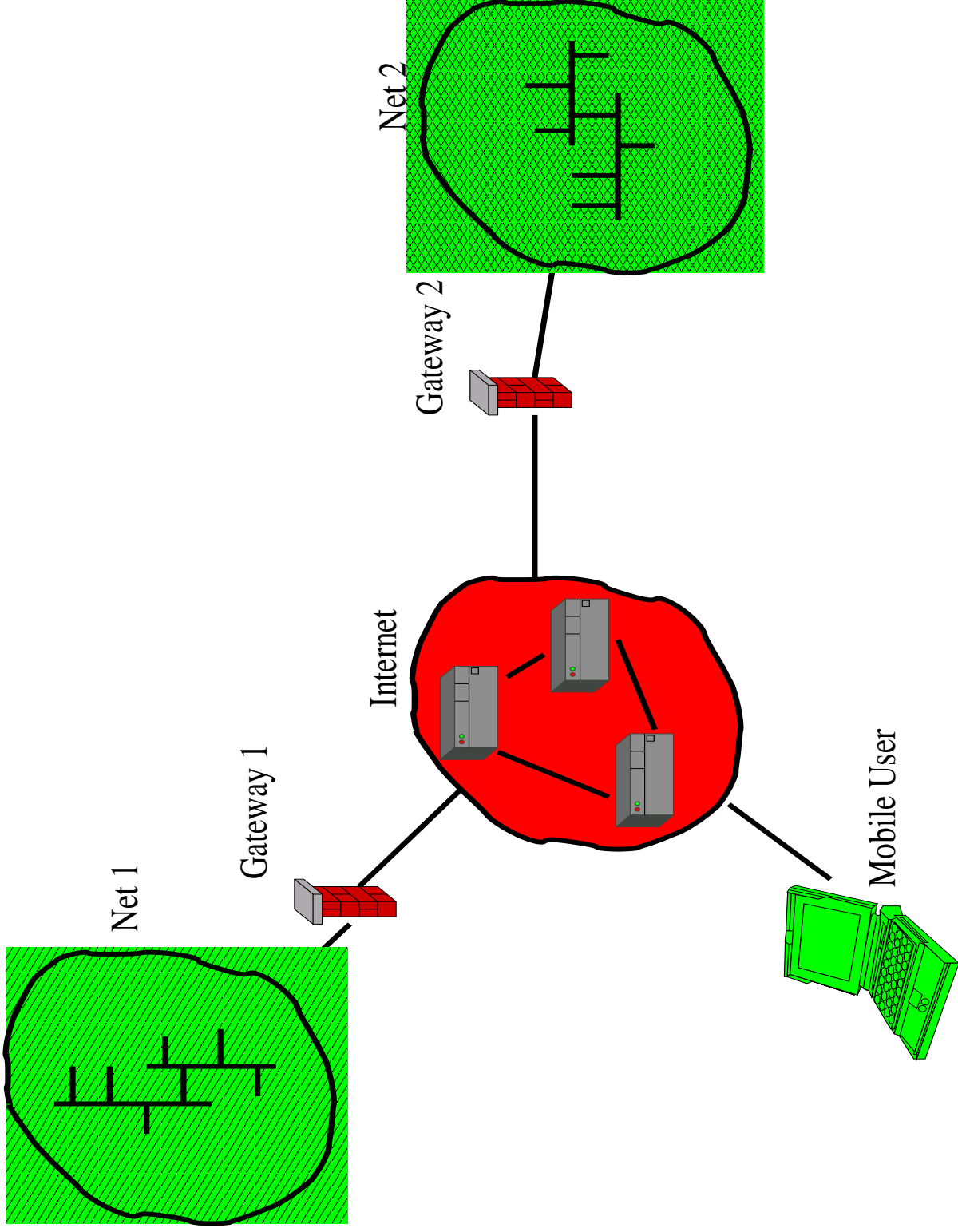
## Comparisons ? (2)

- SSL
  - Widely Used
  - No Infrastructure Changes
  - Limited To TCP
  - Not Transparent
- Kerberos
  - Organization Management
  - Problematic Cross-Domain
  - Needs Application Support
    - A Lot Of It Available

# Example Configurations

- VPN Between OpenBSD Boxes
  - Pre-shared Keys
  - Public Keys
- Road-Warrior Between OpenBSD Boxes

# Network Diagram



# VPN Using OpenBSD (1)

Setup The SPD, on Gateway1:

```
ipsecadm flow -addr Net1 Mask1 Net2 Mask2  
-out -require -dst Gateway2  
ipsecadm flow -addr Net2 Mask2 Net1 Mask1  
-in -require -dst Gateway2
```

Similarly For Gateway2:

```
ipsecadm flow -addr Net2 Mask2 Net1 Mask1  
-out -require -dst Gateway1  
ipsecadm flow -addr Net1 Mask1 Net2 Mask2  
-in -require -dst Gateway2
```

# VPN/OpenBSD - Passphrases

In /etc/isakmpd/isakmpd.conf:

```
[ISAKMP-Peer-Gateway2]  
Phase= 1  
Local-Address= Gateway1  
Address= Gateway2  
Configuration= Default-main-mode  
Authentication= secretpassword
```

```
[Default-main-mode]  
EXCHANGE_TYPE= ID_PROT  
Transforms= 3DES-SHA
```

Similarly For Gateway2

# VPN/OpenBSD - Certificates (1)

Create a private CA (Only Once):

```
openssl genrsa -out /etc/ssl/private/ca.key 1024
```

```
openssl req -new -key /etc/ssl/private/ca.key  
-out /etc/ssl/private/ca.csr
```

```
openssl x509 -req -days 365 -in /etc/ssl/private/ca.csr  
-signkey /etc/ssl/private/ca.key  
-out /etc/ssl/ca.crt
```

On Both Gateways:

```
cp ca.crt /etc/isakmpd/ca/
```

# VPN/OpenBSD - Certificates (2)

Create Keys For The Gateways:

```
openssl genrsa -out /etc/isakmpd/private/local.key 1024
```

```
openssl req -new -key /etc/isakmpd/private/local.key  
-out /etc/isakmpd/private/Gateway1.csr
```

# VPN/OpenBSD - Certificates (3)

Produce Certificates Signed By The CA:

```
openssl x509 -req -days 365 -in Gateway1.csr  
-out Gateway1.crt -CA /etc/ssl/ca.crt  
-CAkey /etc/ssl/private/ca.key -CAcreateserial  
-out Gateway1.crt  
  
certpatch -i Gateway1 -k /etc/ssl/private/ca.key  
Gateway1.crt Gateway1.crt  
  
mv Gateway1.crt /etc/isakmpd/certs/
```

# VPN/OpenBSD - Certificates (4)

Create Policy:

```
openssl x509 -in /etc/isakmpd/ca/ca.crt -text | grep Issuer
```

- e.g., "Issuer: CN=CA Certificate"

In `/etc/isakmpd/isakmpd.policy`:

Authorizer: "POLICY"

Licensees: "DN:/CN=CA Certificate"

# Road Warrior Using OpenBSD (1)

The SPD Entries On Client:

```
ipsecadm flow -addr 0.0.0.0 0.0.0.0 HomeNet Mask  
-out -require -dst Gateway  
ipsecadm flow -addr HomeNet Mask 0.0.0.0 0.0.0.0  
-in -require -dst Gateway
```

The SPD Entries on Gateway:

```
ipsecadm flow -addr 0.0.0.0 0.0.0.0 HomeNet Mask  
-in -require  
ipsecadm flow -addr HomeNet Mask 0.0.0.0 0.0.0.0  
-out -acquire
```

# Road Warrior Using OpenBSD (2)

Use Same Procedure To Create Certificates

When Generating Road Warrior Certificate:

```
certpatch -i -t ufqdn -i ji@att.com  
-k /etc/ssl/private/ca.key  
ji.crt ji.crt
```

# Road Warrior Using OpenBSD (3)

Let's Make The Policy More Involved:

Authorizer: "POLICY"

Licensees: "DN:/CN=CA Certificate"

Conditions: esp\_present == "yes" &&

(esp\_enc\_alg == "3des" ||

esp\_enc\_alg == "aes") &&

esp\_auth\_alg == "hmac-sha" &&

pfs == "yes" &&

@esp\_group\_desc == 2 &&

@esp\_life\_seconds < 1800 &&

remote\_negotiation\_address == remote\_filter;

# Pointers to Documents

[www.cis.upenn.edu/~angelos/Talks/CCS2000-tutorial.ps](http://www.cis.upenn.edu/~angelos/Talks/CCS2000-tutorial.ps)  
[www.ietf.org/html.charters/ipsec-charter.html](http://www.ietf.org/html.charters/ipsec-charter.html)

[www.openbsd.org](http://www.openbsd.org) (FAQ, manpages)

[www.freeswan.org](http://www.freeswan.org)

[www.kame.net](http://www.kame.net)

[www.ietf.org/html.charters/](http://www.ietf.org/html.charters/)

Further Questions:

[angelos@cis.upenn.edu](mailto:angelos@cis.upenn.edu)

[ji@research.att.com](mailto:ji@research.att.com)

