IP Security

Network Mgmt/Sec.

Outline

- intro
- bulk encryption
- sessions and dynamic key mgmt.
- config examples

IP level security/bibliography

- Stallings - Cryptography and Network Security, Prentice Hall
- RFC 2401, “Security Architecture for the Internet Protocol”, Kent/Atkinson, 1998
- RFC 2402, “IP Authentication Header”, Kent/Atkinson, 1998
- RFC 2406, “IP Encapsulating Security Payload (ESP)”, Kent/Atkinson, 1998
- RFC 2407, “The Internet IP Security Domain of Interpretation for ISAKMP”, Piper, 1998.

we are not done yet ...

- RFC 2408, "Internet Security Association and Key Management Protocol" (ISAKMP), Maughan and others, 1998
- RFC 2409, "The Internet Key Exchange(IKE)", Harkins, Carrel, 1998
- RFC 2412, "The OAKLEY Key Determination Protocol", Orman, 1998
- RFC 2411, "IP Security Document Roadmap", Thayer, others, 1998
- per crypto "transform" documents for AH/ESP, e.g., md5/sha/des, etc.

network layer

- various research attempts to bind security in ABOVE IP header

  IP <security header> <TCP>

  - e.g., swipe, U.S. govt. ISO work, Sun SKIP, etc.
- might apply to routes or to end to end transport
- current IETF work called IPSEC - IP security
- must apply to IPv6, and can apply to IPv4
  - NOT IPV6 SPECIFIC !!!!

network layer pros/cons

- pros:
  - can be end to end or at least multi-link unlike link layer
  - could be hw/sw supported (hw support for encryption)
  - can shield unmodified host apps giving them crypto (nets/hosts/and possibly users)
  - can extend secure enclave across insecure areas
- cons:
  - harder to do as may be INSIDE O.S.
  - if not end to end, subject to certain kinds of attacks'
    - proposed plaintext attack

one big pro

- IETF ... and open, NOT enterprise-oriented
- Many national and international security experts and well-known IETF engineers have had their noses in it
- and argued about it for a long time
  - a loooong time ... :->

ipsec big picture

- AH/ESP new IP layer protocols (50/51) with either
  - 1. an IP datagram encapsulated in them (tunnel mode)
  - 2. TCP/UDP and the rest above them (transport mode)
- every packet may have AH/ESP applied to them
- AH for authentication; ESP for encryption (although ESP can have a combined authentication in it now)
- this is bulk/per-packet encryption/authentication

big picture

- key management may be manual (look up keys at boot say and load in kernel, ip must somehow bind keys to AH/ESP actions as packets go through it)
  - e.g., access-list as in current Cisco IOS/OpenBSD
  - or different mechanism/routes in PSU/FreeBSD
- or dynamic, sessions and session-keys negotiated using ISAKMP/OAKLEY protocols
  - session-keys and attributes dynamically bound to AH/ESP packets

big picture, cont.

- exact crypto algorithms can change over time
- new ones introduced / old ones retired
- e.g., AH may use hmac-md5/sha
- esp may use DES/3-DES, etc.
- OAKLEY can be DH authenticated by some public key protocol (e.g., RSA)
  - but a new session-key protocol might be introduced too

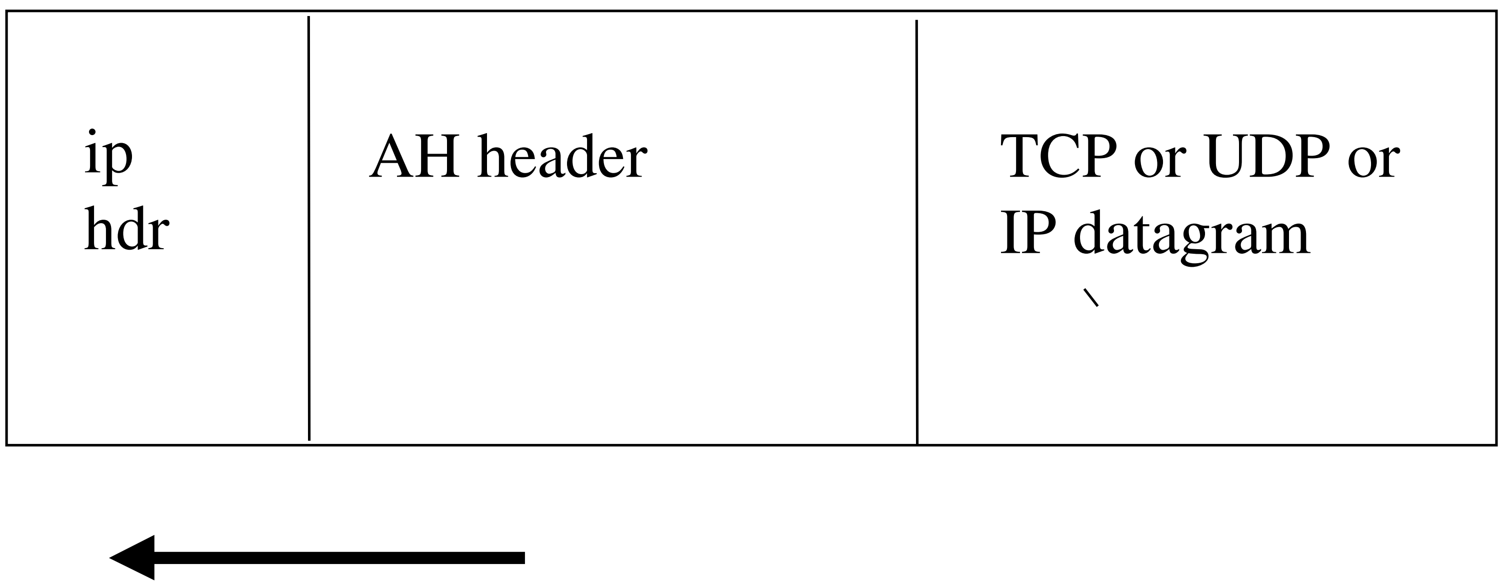
exactly how keys are stored ...

- is not part of the picture
- ISAKMP/OAKLEY could tie to DH public keys and RSA/DSS keys
  – which might be stored in nvram/local files/CA system/kerberos-like KDC, DNS, whatever
  – it's an implementation "detail"
  – true for manual keys used with just AH/ESP for that matter (how loaded is TBD ...)

IPSEC players

- tunnel-mode means one outgoing IP packet is encapsulated in another IP packet with typically (but not necessarily) a different IP dst (a router)
  - can be router to router
  - router to host or host to router (dialup ...)
  - host to host
- end to end may be tunnel or transport modes

AH



AH bound to parts of IP field

ESP

ip hdr	ESP header	TCP/UDP or IP datagram	esp next proto
	esp	tcp/data	trailer

encrypted parts ....

AH header breakdown (v2)

next hdr	length	reserved
Security Parameters Index (SPI)		
Sequence Number		
hash from one-way function (variable)		

ESP header breakdown

SPI (SPY vs. SPY?)
Sequence Number
payload data (variable)
padding 0.255 bytes + pad len + next hdr
optional authentication bits (variable)

note: IV may appear at front of payload

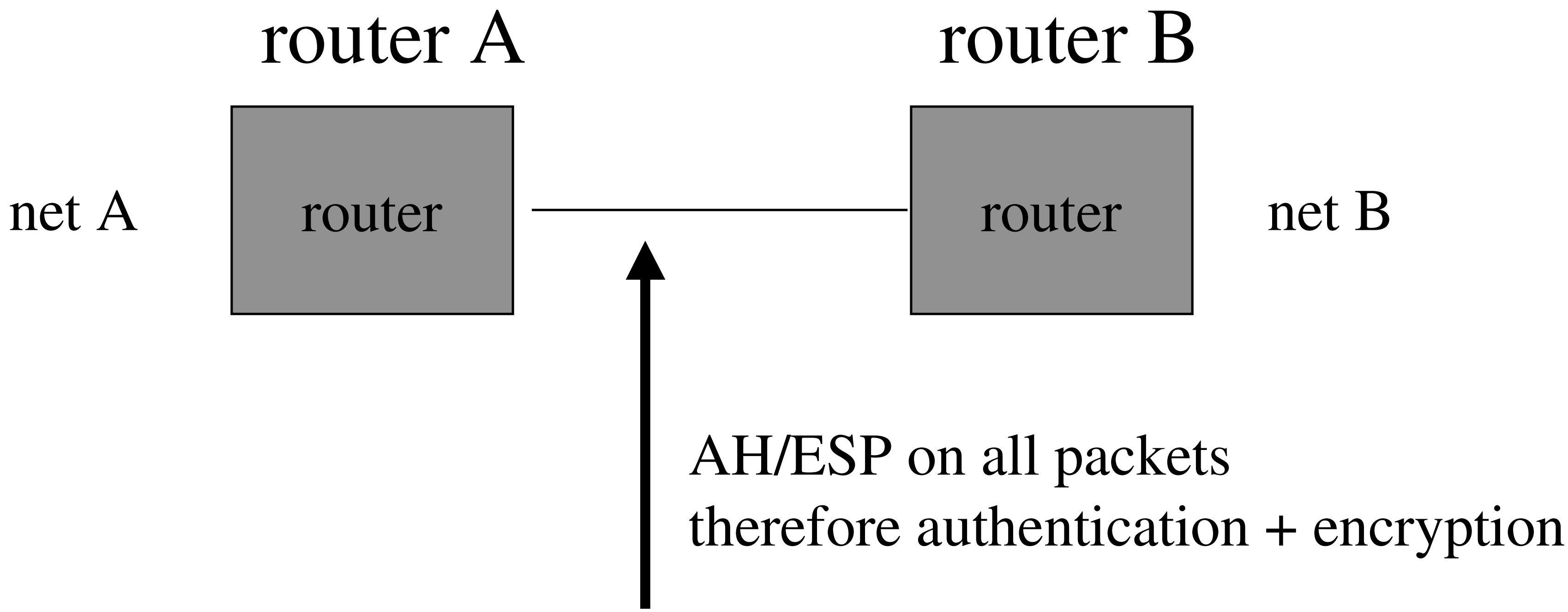
note two versions of IPSEC

- old (or v1) and new (v2)
- old associated with original RFCS, 1825 and up which have been replaced
- v1 AH and ESP lack replay fields
- ESP did not have authentication built-in
- transforms permitted non hmac-md5

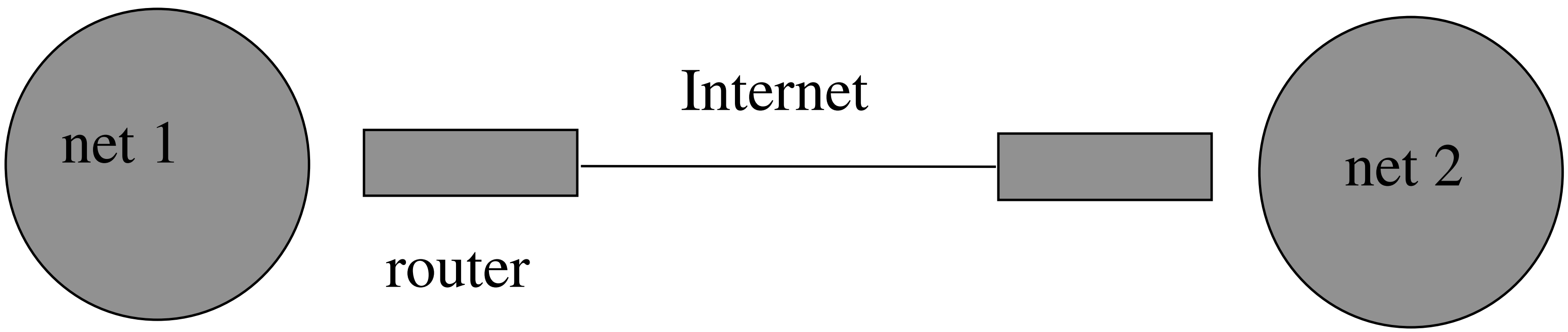
anti-replay

- initial sequence # is 0.
- dest tries to make sure packets are within replay window
- if overflow, SA should be reestablished (problematic for static SA ...)
- essentially window is large as IP pkts may be out of order (default size == 64 pkts)
  - if pkt is outside window, discard and log

IPSEC router/router architecture



Virtual Private Network



all pkts from net 1 to net 2 subject to
authentication/confidentiality
(and vice versa)

tunnel-mode process

- router A takes packet from IP node ip src = 1.1.1.1 to ip dst 2.2.2.2
- A is 1.1.1.2 and B is 2.2.2.1
- A adds new IP header and required AH and/or ESP headers encapsulating entire datagram
- new outer IP hdr, ip src = 1.1.1.2, dst = 2.2.2.1
- A sends packet across IP <IPSEC> IP tunnel to B as destination
- note outer IP and IPSEC bound together, inner datagram including its ip hdr encrypted

B gets packets

- B verifies contents acc to AH/ESP, decrypts in latter case
- strips outer IP and associated IPSEC headers
- routes packet (remaining datagram) which may or may not have interior IPSEC/application security to final local net destination

major note:

- routing is 2 one-way streams
- therefore we have have setup in reverse for packets from B to A
  - and their interior networks/hosts

end to end IPSEC

host a

host b

ip | ipsec | tcp/udp | apps

telnet to telnetd

netscape to httpd

could be tunnel mode too ...
(one SA for all apps/instances)

SA and SPI

- **SA - security association**: classically one way (as is routing):
  - (ip dst, AH or ESP, SPI) is recv. side index
- SPI is opaque number that is mapped to a particular algorithm and keys (DES or IDEA say)
- **SPI - security parameter index**
- AH/ESP by themselves assume keys are placed in kernel manually or via ISAKMP/OAKLEY
- when packet arrives, IP must use SA as key to find appropriate crypto algorithms and keys

SAD - security ass. database

- logical database in o.s. that SA is mapped to called SAD
- includes following parameters:
  – sequence # counter:
  – sequence counter overflow: should sequence overflow cause log entry and prevent retransmission
  – anti-replay window info

SAD., cont.

- SAD db. cont:
  - AH keys/lifetimes/related parameters
  - ESP keys/lifetimes/related parameters
  - lifetime of SA, time or byte count after which SA should be renegotiated or terminated
  - IPSEC protocol mode: tunnel/transport or mumble
  - PATH MTU: what can we send sans fragmentation

SA selector

- can logically state that at higher level we can have policy attribute database
- contains possible selectors used to determine characteristics of IPSEC traffic
- SA **selectors** (or **Security Policy Database**) thus map to SAs which shape IPSEC traffic

SPD entries might be:

- destination IP address: could be host/net/range/list/wildcard (when in doubt do this ...)
- source ip address
- userid: some token to id user
- data sensitivity level: (DOD ...)
- transport layer protocol: TCP, UDP, etc.
- IPSEC protocol, AH/ESP or both
- source/dst ports, tcp/udp

SPD entries cont.

- IPv6 class:
- IPv6 flow label:
- IPv4 Type of Service:
  - all real time traffic is to be authenticated ...
- all systems may not support this much flexibility
  - e.g., routers probably will not have end/end, user attributes, hosts might eventually

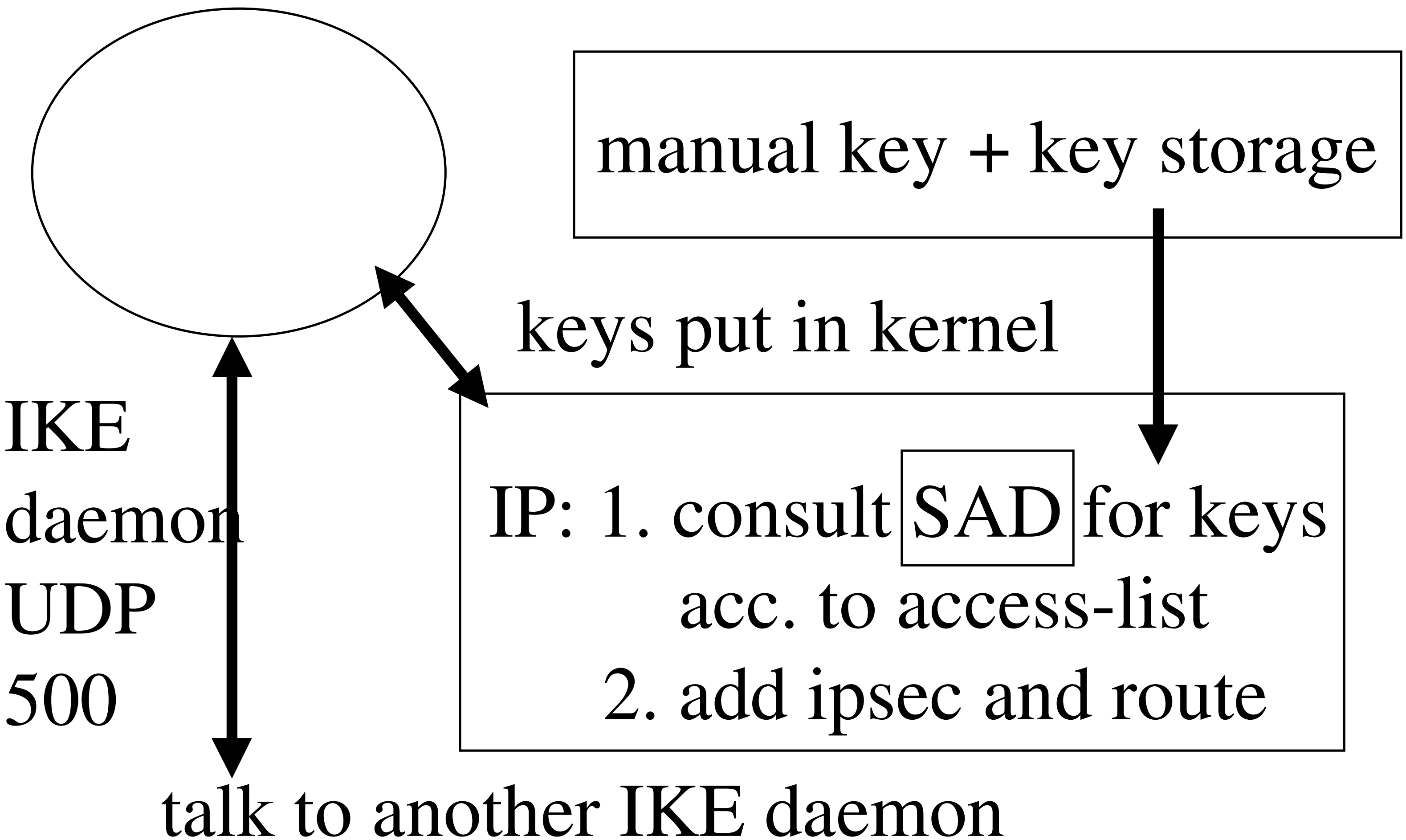
crypto

- HMAC-MD5-96/AH (and ESP too ...)
  - meaning hash is chopped off at 96 bits, not key is 96 bits
- HMAC-SHA-1-96/AH
- IPSEC IP DOI document includes:
- DES-CBC
- 3DES
- RC5
- IDEA, and triple IDEA

more crypto

- CAST
- Blowfish (which can have big keys ...)

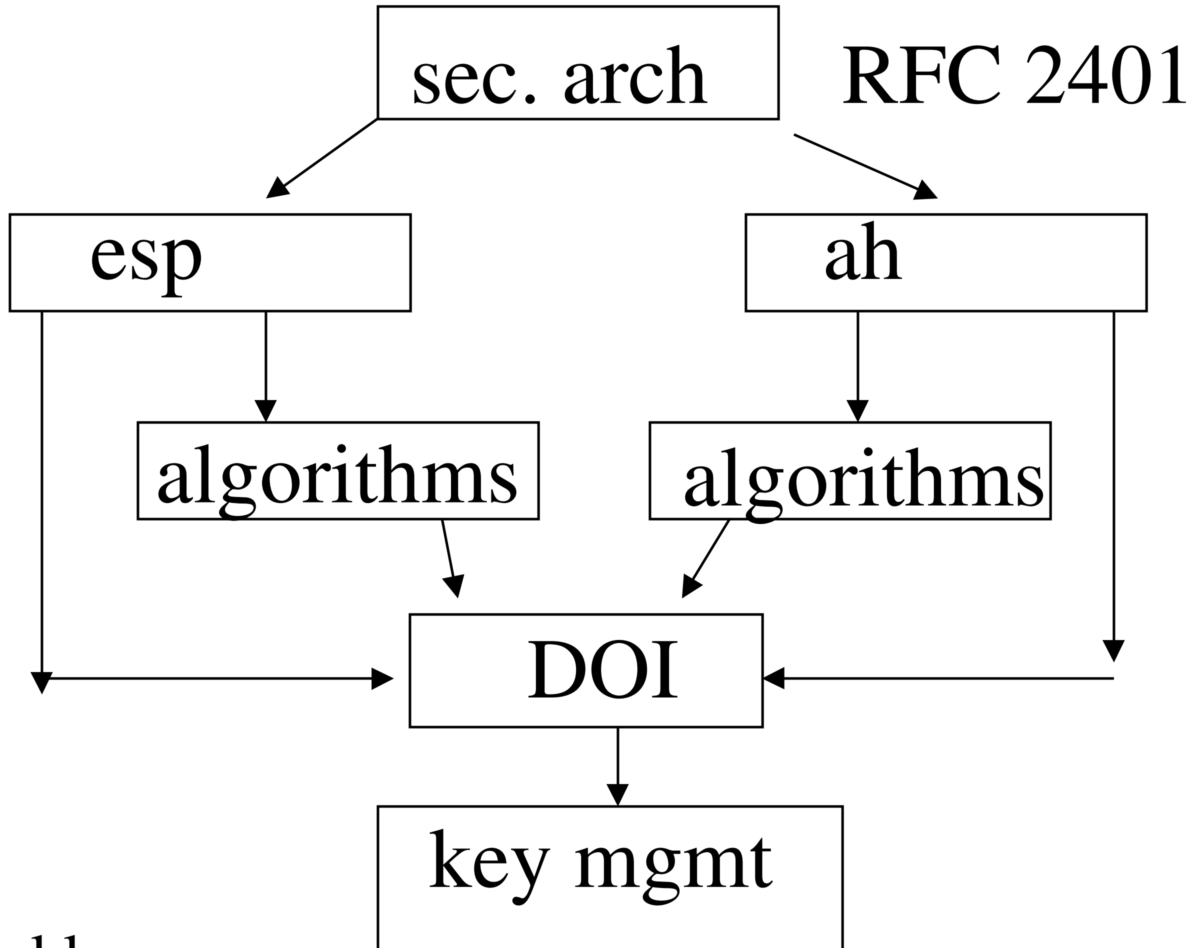
key mgmt. and outbound keys



IKE = ISAKMP + OAKLEY

- ISAKMP is general and wishes to enable
  - an *unspecified* key exchange protocol (e.g., OAKLEY)
  - very general format parameters for cookies, nonces, key material (certificates), etc.
  - and above all, the SA itself
    - » here's the SPI, ESP/AH algorithms desired, etc.
  - therefore also enables dynamic SA generation
  - DOI document exists in part to specify known values to be used with ISAKMP/AH/ESP, etc.

RFC document roadmap picture



DOI includes

- one specific DOI to start with == "IP"
- "instantiates ISAKMP for use with IP when IP uses ISAKMP to negotiate S.A.s" ...
- must define naming scheme for protocol identifiers
- define SA attributes, key exchange types, etc.

e.g.

- IPSEC AH transform values
  - md5(2),sha(3)
- IPSEC ESP transform values
  - descbc(1)/3des(3)/rc5/idea/cast/rc4/null, etc.
- SA attributes including auth/encryption algorithms, groups, key length, compression
- identification payload - id initiator of SA
  - recv must use to somehow determine policy

naming types thus include:

- IPv4 addr
- FQDN
- USER_FQDN (joe@blackhat.com)
- IPv4 subnet
- IPv6 addr/subnet
- IPv4 range/IPv6 range
- ASN1/Distinguished name (X.501)
- ASN1/General name (X.509)
- KEY_ID - opaque string

technical terms:

- **perfect forward secrecy** - regarding session keys, if K(N) is cracked, it should not be possible to use that to crack previous or subsequent session keys
- **group** - set of mathematical attributes used as basis for session-key algorithm; e.g., Diffie-Hellman exponential + specific algorithm
- **identity secrecy** - optional encryption of the identity DOI values in the ISAKMP exchange

IKE = ISAKMP + OAKLEY

- two modes for establishment of authenticated keys
  - main mode (6 messages)
  - aggressive mode (less messages)
- main mode is not optional, aggressive mode is optional
  - difference is in identity protection (not in aggressive mode)
- additional modes include Quick mode and New Group mode
  - quick is for rekeying

overview

- we do policy 1st (SA) which includes:
  - encryption/hash algorithms
  - authentication method
  - information about DH/groups
  - cookies in ISAKMP header
  - this is done first in either mode
- we also have to do session key negotiation
- we may optionally do identity protection (encrypt a “name”) if desired

cookies

- from Photuris KE protocol (Karn/Simpson) comes notion of **anti-clogging** defense
  - not pesky Dutch dance troupe
- each side in initial exchange sends and receives 64 bit random number before DH computation
  - prevents one possible D.O.S. attack using IP spoofing
  - attacker will not be able to ACK your cookie

decode:

- HDR - ISAKMP header
- SA - security association material
- HDR* - encrypted payload in header
- KE - Key Exchange material (for DH)
- Nx - a nonce
- CERT - certificate
- IDx - identification payload
- [] - it's optional

main mode with certificates

- 1st two messages are “policy” (SAs)
  ISAKMP HDR, SA --->
  <-- HDR, SA
- 2nd two messages do DH data and nonces
  HDR, KE, Ni -->
  <-- HDR, KE, Nr
- 3rd send authenticated bits
  HDR*, IDii, [CERT], SIG_I -->
  <-- HDR*, Idir, [CERT], SIG_R

aggressive mode with signatures

- HDR, SA, KE, Ni, IDii -->
  <-- HDR, SA, KE, Nr, Idir, [CERT], SIG_R
- HDR, [CERT,] SIG_I -->

other modes in IKE include

- main/aggressive mode with public key encryption
  - plus a cheaper form with less public key operations
  - cannot prove as with digital signature that conversation occurred
- authentication done with pre-shared keys
  - basically manual key based, id is IP address

details: Oakley groups

- 1. DH prime, generator, 768 bits
- 2. DH prime, generator, 1024 bits
- 3/4 based on elliptical curves (see OAKLEY RFC)

ISAKMP format

- variable-length, generic header + various possible payload headers appended on
- header has (initiator cookie, responder cookie, next payload, Major/Minor, exchange type, flags, message id, length)
- payload (next payload, reserved, length)
  - 0 in next payload means last
  - followed by payload specific data
- multiple payloads in a message

payload types include:

- SA == DOI and other bits
- proposal == some SA specifics, e.g., SPI
- transform == crypto algorithm info
- Key Exchange (data)
- Identification
- Certificate
- Certificate Request
- Hash
- Signature
- Nonce
- Notification (errors) and Delete (burn that one)

sample implementations

- FreeBSD/PSU manual IPSEC keys
- FreeBSD 4.0 - Kame IPv6/IPSEC
  - included/free as is next
- Linux Redhat S/WAN (handout) config
- Cisco IOS 12.0 sample configuration
  - commercial, of course

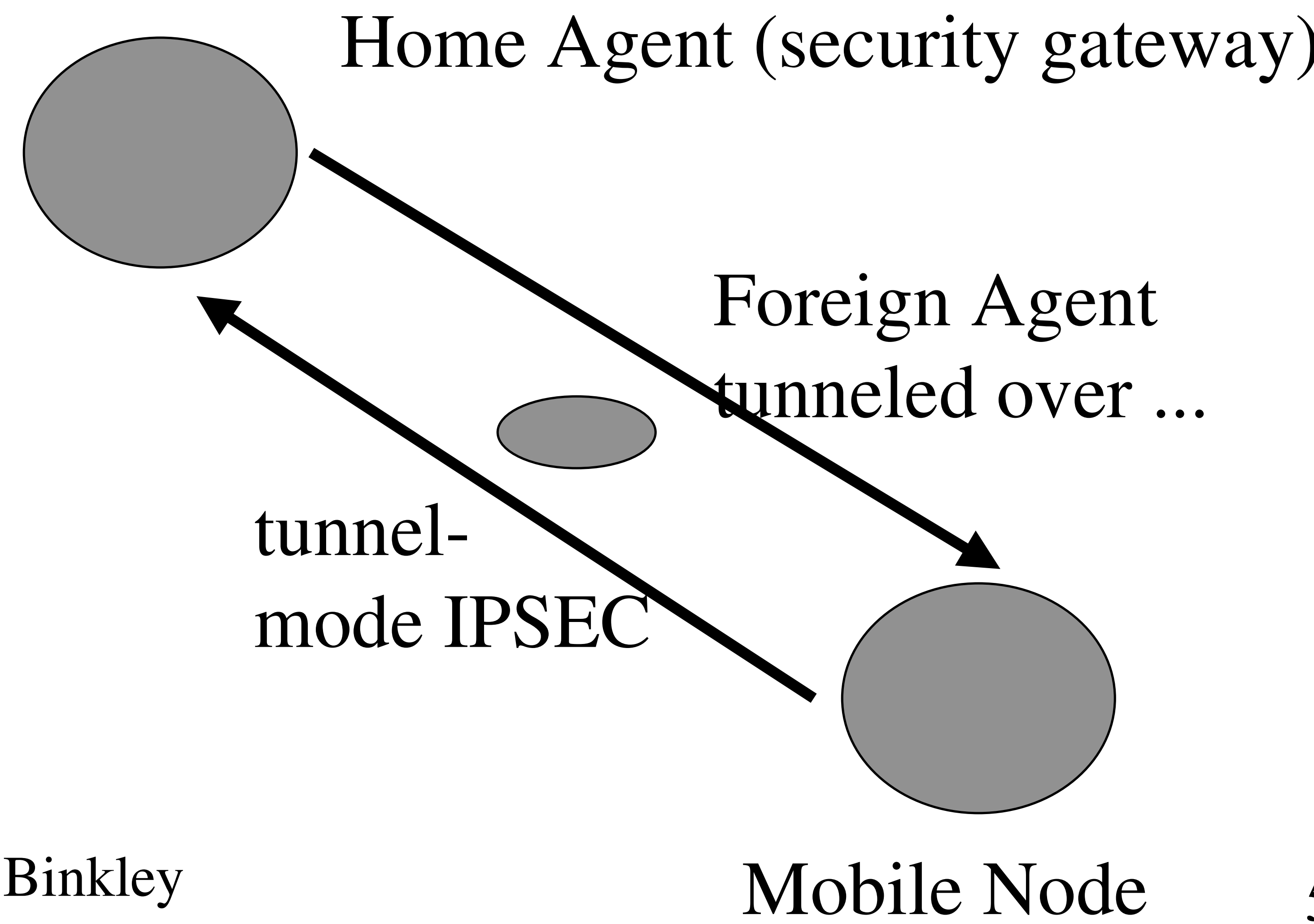
FreeBSD/NRL/PSU route-based

- 1st specify manual keys in /etc/keys
  two lines for two-way exchange
  ah 1234 ip-src ip-dst md5 128-bit-key
  ah 1235 ip-dst ip-src md5 128-bit-key
  esp (similar)
- 2nd load above at boot from /etc/rc.local
  # keyadmin load /etc/keys
- now SAs are in kernel

FreeBSD/NRL/PSU cont.

- statically (or dynamically) add routes which invoke existing SPIs versus SAs
- route add type -spi SPI -itsrc SA -itdst SA destination
- e.g., type might be -ah, -esp, -ahtunnel -esptunnel
- routing daemon may dynamically use route socket to make IPSEC binding to route

IPSEC/Mobile-IP basis



Cisco simple config example

- from www.cisco.com (search on ipsec configuration)
- supported on Cisco 1600/2500/2600/3600/4000/7200/7500/AS5300
- IKE, ah old and new, esp old and new forms supported
- key material still needs to be supplied in next slide
- anti-replay supported only with IKE, not manual setup (makes sense as it can’t rollover ...)

simple ipsec config on cisco

- 1. access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
- 2. crypto ipsec transform-set *myset* esp-des esp-sha
- 3. crypto map *toRemotesite* 10 ipsec-isakmp
- match address 101
- set transform-set *myset*
- set peer 10.2.2.5
- 4. interface Serial0
  - ip address 10.0.0.2
  - crypto map *toRemoteSite*

summary

- IPSEC does not rule out firewalls
  - but may be viewed as a way to talk to a new secure class of bastion hosts or “security gateways”
  - certainly can be firewall feature, talk IPSEC to X
- router to router (or host to host) with application transparency (or host or network) is extreme PRO
  - may be internal or external
- network tunnels (IPIP or GRE) should be combined with IPSEC (else risk 1-way ip spoof attacks thru an unprotected tunnel)

TBD and bottom line

- implementations so far lack user-specific key mechanisms
  – host to host setup by root
  – not ruled out though acc. to RFC 2401 selectors
- compared to certain other mechanisms
  – strong and publicly reviewed by experts
  – transforms can change over time
  – IKE flexible
  – still are key storage/access considerations