IPSEC TRAFFIC PROTOCOLS

- security extensions for IPv4 and IPv6
- IP Authentication Header (AH)
  - authentication and integrity of payload and header
- IP Encapsulating Security Protocol (ESP)
  - confidentiality of payload
- ESP with optional ICV (integrity check value)
  - confidentiality, authentication and integrity of payload

SECURITY DEPENDS UPON

- secure protocols but also much more
  - cryptographic strength
  - implementation quality
  - good random number sources
  - end system security
  - system management
  - .................
**IP Tunneling**

- **Public Internet**
- **Internal Network 1**
- **Security Gateway**
- **Internal Network 2**

**IPSEC Security Association (SA)**

- Referenced by a 32 bit security parameter index (SPI) carried in each IPSEC packet
- SA for an IP packet is uniquely identified by:
  - SPI
  - Destination address
  - Security protocol (AH or ESP)

**IPSEC Security Association (SA)**

- Sequence number counter: 32 bit
- Overflow flag: indicating abort or not on overflow
- Anti-replay window
- AH information: algorithm, key, key lifetime
- ESP information:
  - Encryption: algorithm, key, key lifetime
  - Authentication: algorithm, key, key lifetime
- Lifetime of SA
- IPSEC protocol mode: transport, tunnel, wildcard
- Path MTU (maximum transmission unit)

**IPSEC Keying (SA) Granularity**

- Host Oriented
- User Oriented
- Session Unique
**IP AUTHENTICATION HEADER**

- IPv4 and IPv6 packets
  - data origin authentication
  - data integrity
  - replay prevention (optional as per SA)
- MAC on IP packet header and data payload
- IP header fields that change hop-by-hop set to 0 for MAC computation

**IP AUTHENTICATION HEADER FIELDS**

- next header: 8 bit protocol field
- length: 8 bit field specifying length of authentication data in 32 bit words
- unused: 16 bit set to 0
- SPI: 32 bit
- sequence number: 32 bit
- integrity check value (ICV): some multiple of 32 bits, e.g., 96, 128, 160
  - must support HMAC-MDS-96, HMAC-SHA-1-96

**IP AH TRANSPORT MODE**

- Original IP HEADER
- AH
- TCP/UDP/ICMP/IP payload
- protocol field of IP header is 51 (for AH payload)
- AH in turn contains protocol field specifying protocol of actual payload, e.g., TCP or UDP or ICMP or IP

**IP AUTHENTICATION HEADER**

- prevents IP spoofing attacks
  - at performance cost
- prevents replay attacks
  - sequence number added in revision
- can be widely and strongly deployed without concern of crypto-politics

**IP AH TUNNEL MODE**

- New IP HEADER
- AH
- Original IP HEADER
- TCP/UDP/ICMP/IP payload
- IP AH is a single protocol
- transport or tunnel mode is determined by SA
  - actually SA can allow both

**ANTI-REPLAY MECHANISM**

- Sequence number starts at 1 and cannot go past $2^{32}-1$
- receiver keeps a window of min size 32 (64 preferred, larger is ok)
  - packets to left of window are discarded
  - repeated packets within window are discarded
  - authentic packets to right of window cause window to move right
**IP ENCAPSULATING SECURITY PAYLOAD (ESP)**

- IPv4 and IPv6
  - ESP: data confidentiality
  - ESP w/Auth: data confidentiality, authentication, integrity
  - ESP w/Auth is an option within ESP
- ESP header (cleartext)
  - SPI, IV in cleartext
  - sequence number: 32 bit
  - Initial Value for CRC
- ESP trailer (encrypted)
  - padding
  - next header (identifies payload protocol)
- ESP w/Auth authentication
  - ICV: for authentication option
  - applies only to encrypted payload and not to header

**ESP**

- protocol 50
  - ESP w/Auth determined by SA
- ESP header
  - SPI, IV in cleartext
- ESP trailer
  - padding info, payload protocol is encrypted
  - tunnel mode provides partial traffic flow confidentiality

**ESP TUNNELING MODE**

**INTERNET KEY EXCHANGE**

- Hybrid protocol

  - ISAKMP
  - PHOTURIS
  - SKEME
  - MKMP
  - SKIP
  - OAKLEY

**ISAKMP**

- Internet security association and key management protocol
- separates key management from key exchanges
- complex general protocol used in a specific way in IKE
  - can apply to protocols other than IPSEC
- for IPSEC uses UDP over IP
### IKE

- **ISAKMP phase 1:** establishes ISAKMP SA
  - Main mode (DH with identity protection)
  - Aggressive mode (DH without identity protection)
- **Between phases:**
  - New group mode
- **ISAKMP phase 2:** establishes SA for target protocol
  - Quick mode

### PHASE 1 AUTHENTICATION ALTERNATIVES

- public-key signature
- preshared-key
- public-key encryption
- revised public-key encryption

### DIFFIE-HELLMAN KEY ESTABLISHMENT

<table>
<thead>
<tr>
<th>A</th>
<th>y_A = a^x_A mod p</th>
<th>B</th>
<th>y_B = a^x_B mod p</th>
</tr>
</thead>
<tbody>
<tr>
<td>public key</td>
<td>private key</td>
<td>public key</td>
<td>private key</td>
</tr>
</tbody>
</table>

\[ k = y_B^{x_A} \mod p = y_A^{x_B} \mod p = a^{x_A \cdot x_B} \mod p \]

**System constants:** p: prime number, a: integer

### COOKIE EXCHANGE

- Phase 1 employs cookie exchange to thwart (not prevent) denial of service attacks
- A -> B: Cookie_Request
  - A’s cookie, 64 bit random number
- B -> A: Cookie_Response
  - Includes A and B’s cookies
- All further Phase 1 and Phase 2 messages include both cookies
  - ISAKMP SA is identified by both cookies
  - IPSEC protocol SA is identified by SPI

### PERFECT FORWARD SECRECY

- Use a different DH key-pair on each exchange
- DH public keys need to be authenticated
  - Authentication can be done by many techniques
- Loss of long-term (authentication) keys does not disclose session keys

### COOKIE GENERATION

- Hash over
  - IP Source and Destination Address
  - UDP Source and Destination Ports
  - A locally generated random secret
  - Timestamp
IKE DEFAULT OAKLEY DH GROUPS

- Group 1
  - MODP, 768 bit prime p, g=2
- Group 2
  - MODP, 1024 bit prime p, g=2
- Group 3
  - EC2N, 155 bit field size
- Group 4
  - EC2N, 185 bit field size
- private groups can be used

IKE NOTATION

HDR ISAKMP header whose exchange type is the mode
HDR* indicates payload encryption
SA SA negotiation payload, initiator MAY provide multiple
proposals, responder replies with one
SAi_b body of the SA payload (minus generic headers)
CKY-I Initiator’s cookie
CKY-R Responder’s cookie
g^xi initiator’s DH public value
g^xr responder’s DH public value
g^xy Diffie-Hellman shared secret
SAi_b body of the SA payload (minus generic headers)
Ni initiator nonce
Nr responder nonce
Idii identification payload for ISAKMP initiator
Idir identification payload for ISAKMP responder
SIG signature payload, data signed varies
CERT certificate payload
HASH hash payload

SKEYS, HASH AND SIG

SKEYID_e = prf(SKEYID, g^xy | CKY-I | CKY-R | Idii | CERT | SIG_I)
SKEYID_a = prf(SKEYID, SKEYID_e | g^xy | CKY-I | CKY-R | Idii | CERT | SIG_I)
SKEYID_d = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 0)

HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b | CERT) or
      digitally signed by SIG_I

SKEYID = prf(Ni b | Nr b, g^xy)

MAIN MODE WITH DIGITAL SIGNATURES

Initiator                          Responder
----------                        ----------
HDR, SA                     -->    HDR, SA
HDR, KE, Ni                 -->    HDR, KE, Nr

SKEYID = prf(Ni b | Nr b, g^xy)

AGGRESSIVE MODE WITH DIGITAL SIGNATURES

Initiator                          Responder
----------                        ----------
HDR, SA, KE, Ni, IDii      -->    HDR, SA, KE, Nr, IDir,
HDR, [ CERT, ] SIG_I     -->    HDR, [ CERT, ] SIG_R

SKEYID = prf(Ni b | Nr b, g^xy)
MAIN AND AGGRESSIVE MODE WITH PRE-SHARED KEY

MAIN MODE

Initiator Responder
----------  ------------
HDR, SA --> HDR, SA
HDR, KE, Ni --> HDR, KE, Nr
HDR*, IDii, HASH_I --> HDR*, IDir, HASH_R

AGGRESSIVE MODE

Initiator Responder
----------  ------------
HDR, SA, KE, Ni, IDii --> HDR, SA, KE, Nr, IDir, HASH_R
HDR, HASH_I --> HDR, HASH_R

SKEYID = prf(pre-shared-key, Ni_b | Nr_b)

MAIN MODE WITH PUBLIC KEY ENCRYPTION

Initiator Responder
----------  ------------
HDR, SA --> HDR, SA
HDR, KE, [HASH(1),] <KE_b>PubKey_r, <Ni_b>PubKey_r --> HDR, KE, <DIR_b>PubKey_i, <Nr_b>PubKey_i
HDR*, HASH_I --> HDR*, HASH_R

SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)

AGGRESSIVE MODE WITH PUBLIC KEY ENCRYPTION

Initiator Responder
----------  ------------
HDR, SA, [HASH(1),] KE, <IDii_b>PubKey_r, <KE_b>PubKey_r --> HDR, KE, <DIR_b>PubKey_i, <Nr_b>PubKey_i
HDR, HASH_I --> HDR, HASH_R

SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)

MAIN MODE WITH REVISED PUBLIC KEY ENCRYPTION

Initiator Responder
----------  ------------
HDR, SA --> HDR, SA
HDR, [HASH(1),] <KE_b>PubKey_r, <Ni_b>PubKey_r --> HDR, KE, <DIR_b>PubKey_i, <Nr_b>PubKey_i
HDR*, HASH_I --> HDR*, HASH_R

SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)

MAIN MODE WITH REVISED PUBLIC KEY ENCRYPTION

Ne_i = prf(Ni_b, CKY-I)
Ne_r = prf(Nr_b, CKY-R)

Ke_i is leftmost 320 bits of K1 | K2 | K3 where
K1 = prf(Ke_i, 0)
K2 = prf(Ke_i, K1)
K3 = prf(Ke_i, K2)

Similarly for Ke_r
AGGRESSIVE MODE WITH REVISED PUBLIC KEY ENCRYPTION

Initiator                        Responder
-----------                      -----------
HDR, ESK, <HASH(1), <KEK_b<Ke_i, <IDii_b<Ke_i, <cert_b<Ke_i -->
HDR, ESK, <KEK_r<Ke_r, <IDir_b<Ke_r, HASH_R -->
HDR, ESH, HSH_I -->

PHASE 2 QUICK MODE

Initiator                        Responder
-----------                      -----------
HDR*, HASH(1), SA, Ni |- KE |- IDci, IDcr | -->
<-- HDR*, HASH(2), SA, Nr |- KE |- IDci, IDcr |
<-- HDR*, HASH(3) -->

Results in 4 security associations-- 2 each way for both SA0 and SA1

PHASE 2 QUICK MODE

Additional key material can be generated if needed as follows:

KEYMAT = K1 | K2 | K3 | ... where

K1 = prf(SKEYID_d, g(qm)^xy | protocol | SPI | M-ID | Ni | Nr)
K2 = prf(SKEYID_d, g(qm)^xy | protocol | SPI | M-ID | Ni | Nr)

In either case, "protocol" and "SPI" are from the ISAKMP Proposal Payload that contained the negotiated Transform.

Two SAs are established
One in each direction
Keys are different because of different SPIs

NEW GROUP MODE

$v$ sandwiched between phase 1 and 2
$\triangleright$ group can be negotiated in phase 1
$\triangleright$ new group mode allows nature of group to be hidden
$\triangleright$ in phase 1 only group id is communicated in clear
NEW GROUP MODE

Initiator                        Responder
--------                        --------
HDR*, HASH(1), SA        -->     HDR*, HASH(2), SA
<--  HASH(1) = prf(KEYID_a, M-ID | SA)
HASH(2) = prf(KEYID_a, M-ID | SA)

VIRTUAL PRIVATE NETWORKS

VPNs

- VPNs are used to securely connect networks using tunnels (virtual circuits) over the Internet
- Secure remote access is used to securely connect a single computer using tunnels (virtual circuits) over the Internet

WHAT IS TUNNELED

- IPSEC tunnel can be used to tunnel
  - IP packets
    - IPSEC standard approach
  - Layer 2 packets
    - virtual switched LAN (VSLAN)
    - proprietary approaches

VIRTUAL PRIVATE NETWORKS

IPSEC tunnel

Public Internet

Internal Network 1

IP ESP w/ICV or IP AH or both tunnel mode

Internal Network 2

IPSEC tunnel

VPNs

- IPSEC tunnel can be used to tunnel
  - IP packets
    - IPSEC standard approach
  - Layer 2 packets
    - virtual switched LAN (VSLAN)
    - proprietary approaches

VIRTUAL PRIVATE NETWORKS

VPNs

- IPSEC tunnel can be used to tunnel
  - IP packets
    - IPSEC standard approach
  - Layer 2 packets
    - virtual switched LAN (VSLAN)
    - proprietary approaches
VIRTUAL PRIVATE NETWORKS

Public Internet

Internal Network 1

Internal Network 2

IPSEC tunnel

IP ESP w/ICV or IP AH or both tunnel mode

IPSEC tunnel

SECURE REMOTE ACCESS

Public Internet

Internal Network 1

Remote Laptop

IPSEC tunnel

IP ESP w/ICV or IP AH or both tunnel mode

PPTP VPNs

✧ Voluntary tunneling
  ✧ PPTP tunnel from client to network

✧ Compulsory tunneling
  ✧ PPTP tunnel from ISP to network
  ✧ client to ISP dial-in via PPP is unprotected

PPTP VPNs

✧ Originally intended for secure remote access

✧ enhancements for network to network VPNs

✧ known security flaws
  ✧ remedied in version 2