INFS 766
Internet Security Protocols

Lectures 7 and 8
IPSEC

Prof. Ravi Sandhu

IPSEC ROADMAP

- Security Association
- IP AH (Authentication Header) Protocol
- IP ESP (Encapsulating Security Protocol)
- Authentication Algorithm
- Encryption Algorithm
- IKE (Key Exchange)
- [IP Compression Protocol and Algorithms]

SECURITY DEPENDS UPON

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

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

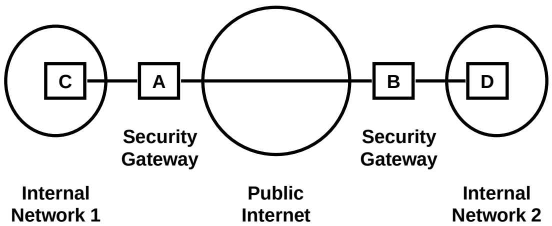
IPSEC TRAFFIC PROTOCOLS

- security services
  - authentication and integrity
  - confidentiality
  - replay prevention
  - partial traffic flow confidentiality
  - compression
- algorithm-independent with standard defaults
- secret-key technology

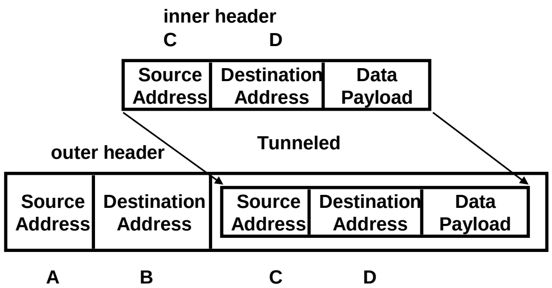
IPSEC TRAFFIC PROTOCOLS

- both IP AH and IP ESP can operate in
  - transport mode
    - end-to-end
  - tunnel mode
    - security-gateway to security-gateway
- transport mode and tunnel model can coexist

IP TUNNELING



IP TUNNELING



IPSEC SECURITY ASSOCIATION (SA)

- **SA is a one-directional relationship between sender and receiver**
- **SA applies to AH or ESP but not both**
- **two-way secure exchange of IP packets requires two (or more) SAs**
- **unicast (multicast will come later)**
- **SAs are established by**
  - **management protocols (IKE)**
  - **manually**

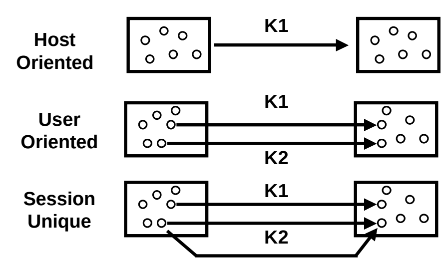
IPSEC SECURITY ASSOCIATION (SA)

- **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



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 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 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

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-MD5-96, HMAC-SHA-1-96

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

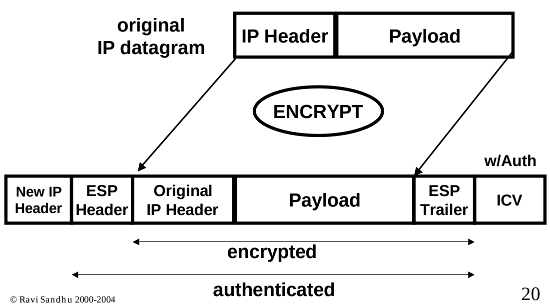
ANTI-REPLAY MECHANISM

- 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

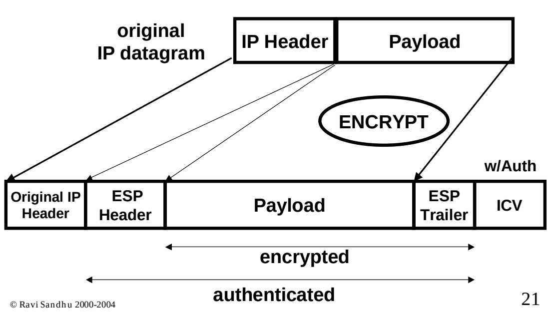
IP ENCAPSULATING SECURITY PAYLOAD (ESP)

- IPv4 and IPv6
  - ESP: data confidentiality
  - ESP w/Auth: data confidentiality, authentication, integrity
  - ESP w/Auth is an option within ESP
- ESP header (cleartext)
  - security parameter index (SPI)
  - sequence number: 32 bit
  - Initial Value for CBC
- 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 TUNNELING MODE



ESP TRANSPORT MODE

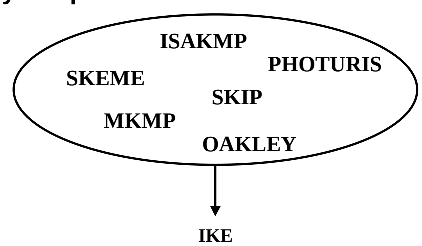


ESP

- 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

INTERNET KEY EXCHANGE (IKE)

- Hybrid protocol



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

DIFFIE-HELLMAN KEY ESTABLISHMENT

A — $y_A = a^{x_A} \bmod p$ public key; private key x_A

B — $y_B = a^{x_B} \bmod p$ public key; private key x_B

$$k = y_B^{x_A} \bmod p = y_A^{x_B} \bmod p = a^{x_A * x_B} \bmod p$$

system constants: p: prime number, a: integer

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

PHASE 1 AUTHENTICATION ALTERNATIVES

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

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

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
<P>_b    body of payload <P>
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
KE       key exchange containing DH public values
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
```


IKE NOTATION

```
prf(key, msg) keyed pseudo-random function (often MAC)

SKEYID   string derived from secret material known only to the active
         players in the exchange
SKEYID_e keying material used by the ISAKMP SA to protect confidentiality
         of its messages.
SKEYID_a keying material used by the ISAKMP SA to protect authentication
         of its messages.
SKEYID_d keying material used to derive keys for non-ISAKMP SAs


<x>y     "x" is encrypted with the key "y"

-->      initiator to responder
<--      responder to initiator

|        concatenation of information
[x]      indicates that x is optional
```


SKEYS, HASH AND SIG

```
SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2)


HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b )
HASH_R = prf(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b )

HASH_I and HASH_R used directly for MAC authentication OR
digitally signed by SIG_I and SIG_R
```


MAIN MODE WITH DIGITAL SIGNATURES

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


            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,
                                            [ CERT, ] SIG_R
 HDR, [ CERT, ] SIG_I           -->


            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                   -->

 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), ]
   <IDii_b>PubKey_r,
     <Ni_b>PubKey_r            -->
                                      HDR, KE, <IDir_b>PubKey_i,
                               <--             <Nr_b>PubKey_i
 HDR*, HASH_I                  -->
                               <--    HDR*, HASH_R


      HASH(1) is hash of responder's certificate

    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,
    <Ni_b>Pubkey_r             -->
                                      HDR, SA, KE, <IDir_b>PubKey_i,
                               <--         <Nr_b>PubKey_i, HASH_R
 HDR, HASH_I                   -->


             Provides identity protection

      HASH(1) is hash of responder's certificate

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


AUTHENTICATION WITH PUBLIC-KEY ENCRYPTION

- **does not provide non-repudiation**
- **provides additional security since attacked must break both**
  - **DH key exchange**
  - **public-key encryption**
- **provides identity protection in aggressive mode**
- **revised protocol reduces public-key operations**

MAIN MODE WITH REVISED PUBLIC KEY ENCRYPTION

```
 Initiator                              Responder
-----------                            -----------
 HDR, SA                       -->
                               <--    HDR, SA
 HDR, [ HASH(1), ]
   <Ni_b>Pubkey_r,
   <KE_b>Ke_i,
   <IDii_b>Ke_i,
   [<Cert-I_b>Ke_i]           -->
                                      HDR, <Nr_b>PubKey_i,
                                           <KE_b>Ke_r,
                               <--         <IDir_b>Ke_r,
 HDR*, HASH_I                  -->
                               <--    HDR*, HASH_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 leftomost 320 bits of K1 | K2 | K3 where
K1 = prf(Ne_i, 0)
K2 = prf(Ne_i, K1)
K3 = prf(Ne_i, K2)

Similarly for Ke_r
```

AGGRESSIVE MODE WITH REVISED PUBLIC KEY ENCRYPTION

```
 Initiator                        Responder
-----------                      -----------
 HDR, SA, [ HASH(1),]
   <Ni_b>Pubkey_r,
   <KE_b>Ke_i, <IDii_b>Ke_i
   [, <Cert-I_b>Ke_i ]      -->
                                  HDR, SA, <Nr_b>PubKey_i,
                                       <KE_b>Ke_r, <IDir_b>Ke_r,
                            <--        HASH_R
 HDR, HASH_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)             -->
```

```
HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [ | KE ] [ | IDci | IDcr ])
HASH(2) = prf(SKEYID_a, M-ID | Ni_b | SA | Nr [ | KE ] [ | IDci | Idcr ] )
HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)
```


PHASE 2 QUICK MODE

```
If no PFS there is no KE payload and new keying material is

    KEYMAT = prf(SKEYID_d, protocol | SPI | Ni_b | Nr_b).

If PFS there is KE payload and new keying material is

    KEYMAT = prf(SKEYID_d, g(qm)^xy | protocol | SPI | Ni_b | Nr_b)

where g(qm)^xy is the shared secret from the ephemeral DH
exchange of this Quick Mode (which must then be deleted)

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
```


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 | Ni_b | Nr_b)
       K2 = prf(SKEYID_d, K1 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
       K3 = prf(SKEYID_d, K2 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
       etc.
```


PHASE 2 QUICK MODE

```
Multiple SA's and keys can be negotiated with one exchange as follows:

     Initiator                        Responder
    -----------                      -----------
     HDR*, HASH(1), SA0, SA1, Ni,
       [, KE ] [, IDci, IDcr ] -->
                               <--    HDR*, HASH(2), SA0, SA1, Nr,
                                         [, KE ] [, IDci, IDcr ]
     HDR*, HASH(3)             -->

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


NEW GROUP MODE

- **sandwiched between phase 1 and 2**
- **group can be negotiated in phase 1**
- **new group mode allows nature of group to be hidden**
  - **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(SKEYID_a, M-ID | SA)
              HASH(2) = prf(SKEYID_a, M-ID | SA)
```


VIRTUAL PRIVATE NETWORKS

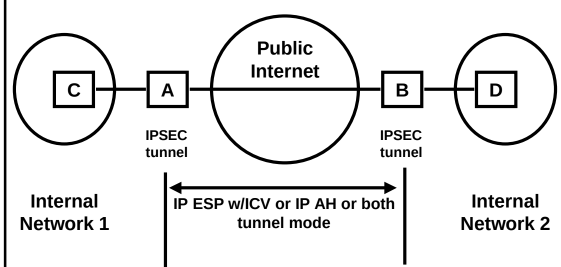
VPNs

- **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**

VPN TECHNOLOGIES

- **IPSEC**
  - **layer 3 VPN (standards based), layer 2 VPN (proprietary)**
- **PPTP (Point-to-point tunneling protocol)**
  - **Microsoft layer 2 VPN, built in security with known flaws**
- **L2F (layer 2 forwarding)**
  - **Cisco layer 2 VPN, no security, phasing out**
- **L2TP (layer 2 tunneling protocol)**
  - **emerging IETF standard, needs IPSEC security**
- **SSL (layer 4 tunnel)**
  - **proprietary approaches, tunnel IP over SSL-protected TCP**

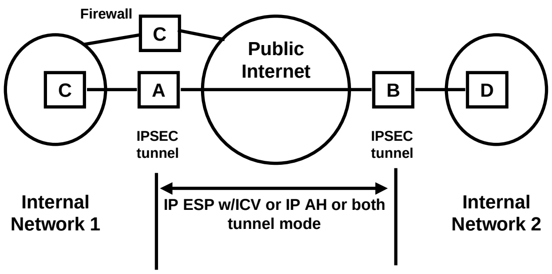
VIRTUAL PRIVATE NETWORKS



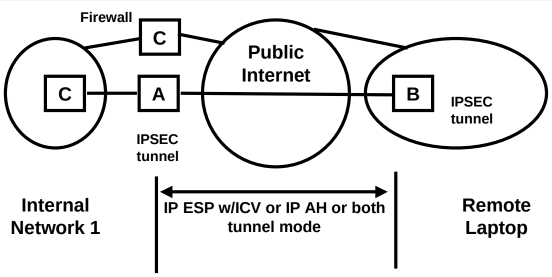
WHAT IS TUNNELED

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

VIRTUAL PRIVATE NETWORKS



SECURE REMOTE ACCESS



PPTP VPNs

- **Originally intended for secure remote access**
- **enhancements for network to network VPNs**
- **known security flaws**
  - **remedied in version 2**

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**