IPsec (AH, ESP), IKE

Guevara Noubir
CSG254: Network Security
noubir@ccs.neu.edu
## Securing Networks

<table>
<thead>
<tr>
<th>Layer</th>
<th>Protocols/Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications Layer</td>
<td>Applications Layer</td>
</tr>
<tr>
<td></td>
<td>telnet/ftp: ssh, http: <strong>https</strong>, mail: <strong>PGP</strong></td>
</tr>
<tr>
<td>(SSL/TLS)</td>
<td></td>
</tr>
<tr>
<td>Transport Layer (TCP)</td>
<td></td>
</tr>
<tr>
<td>(IPSec, IKE)</td>
<td></td>
</tr>
<tr>
<td>Network Layer (IP)</td>
<td></td>
</tr>
<tr>
<td>Link Layer</td>
<td></td>
</tr>
<tr>
<td>(IEEE802.1x/IEEE802.10)</td>
<td></td>
</tr>
<tr>
<td>Physical Layer</td>
<td></td>
</tr>
<tr>
<td>(spread-Spectrum, quantum crypto, etc.)</td>
<td></td>
</tr>
</tbody>
</table>

**Control/Management (configuration):**

**Network Security Tools:**

- Monitoring/Logging
- Intrusion Detection

---

**CSG254: Network Security**

**IPsec - IKE**
SSL vs. IPsec

SSL:
- Avoids modifying “TCP stack” and requires minimum changes to the application
- Mostly used to authenticate servers

IPsec:
- Transparent to the application and requires modification of the network stack
- Authenticates network nodes and establishes a secure channel between nodes
- Application still needs to authenticate the users
IPsec Protocol Suite
(IETF Standard)

- Provides inter-operable crypto-based security services:
  - Services: confidentiality, authentication, integrity, and key management

- Protocols:
  - Authentication Header (AH): RFC2402
  - Encapsulated Security Payload (ESP): 2406
  - Internet Key Exchange (IKE)

- Environments: IPv4 and IPv6

- Modes:
  - Transport (between two hosts)
  - Tunnel (between hosts/firewalls)
Assumption:
- End nodes already established a shared session key:
  - Manually or IKE

Security Association:
- Each secure connection is called a security association (SA)
- For each SA: key, end-node, sequence number, services, algorithms
- SA is unidirectional and identified by:
  - (destination-address, SPI = Security Parameter Index)

Protocols:
- Authentication Header: integrity protection
- Encapsulated Security Payload: encryption and/or integrity
### IP Packets

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>04</td>
</tr>
<tr>
<td>HLen</td>
<td>08</td>
</tr>
<tr>
<td>TOS</td>
<td>01</td>
</tr>
<tr>
<td>Length</td>
<td>09</td>
</tr>
<tr>
<td>Flags</td>
<td>03</td>
</tr>
<tr>
<td>Offset</td>
<td>01</td>
</tr>
<tr>
<td>TTL</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>SourceAddr</td>
<td></td>
</tr>
<tr>
<td>DestinationAddr</td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td></td>
</tr>
<tr>
<td>Pad (variable)</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**

```plaintext
0 4 8 16 19 31

<table>
<thead>
<tr>
<th>Version</th>
<th>HLen</th>
<th>TOS</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>04</td>
<td>08</td>
<td>01</td>
</tr>
<tr>
<td></td>
<td>09</td>
<td>03</td>
<td>01</td>
</tr>
<tr>
<td>TTL</td>
<td>Protocol</td>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>SourceAddr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DestinationAddr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td>Pad (variable)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
AH Formatting

AH Protocol Number = 51

Transport mode

Tunnel mode

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Length (8)</th>
<th>Reserved (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Parameters Index (32)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number Field (32)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication Data (N*32)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SN: for replay detection
ESP Formatting

ESP Protocol Number = 50

Transport mode

Tunnel mode
ESP Header

- Security Parameters Index (32)
- Sequence Number Field (32)
- Initialization Vector (variable)
- Payload Data (variable)
  - Padding (0-255 bytes)
  - Pad Length (8)
  - Next Header
- Authentication Data

Auth/Integrity coverage

Confidentiality coverage
Issues

- NAT boxes:
  - IPsec tunnel mode doesn’t easily work

- Firewalls
  - IPsec encrypts information used by firewalls to filter traffic (e.g., port number)

- AH mutable/immutable/predictable fields:
  - Some fields get modified by the intermediate routers and can’t be protected by the AH
  - Mutable: type of service, flags, fragment offset, TTL, header checksum
  - Why is PAYLOAD-LENGTH considered immutable (even if packets can be fragmented)? Why not fragment offset. Inconsistency!
  - Mutable but predictable fields are included in the AH computation using their expected value at the destination (e.g., destination address even when using source routing)
IPsec: Internet Key Exchange

- **Goal:**
  - Mutual authentication and establishment of a shared secret session key using:
    - Pre-shared secret key or public signature-only key, or public encryption key
    - Negotiation of features and cryptographic algorithms

- **Specification documents:**
  - ISAKMP (Internet Security Association and Key Management Protocol): RFC 2408
  - IKE: RFC 2409
  - DOI (Domain Of Interpretation): RFC 2407
Photuris

- Photuris goal: signed Diffie-Hellman exchange
  1. $A \rightarrow B: C_A$
  2. $B \rightarrow A: C_A, C_B'$ crypto offered
  3. $A \rightarrow B: C_A, C_B g^a \pmod p$, crypto selected
  4. $B \rightarrow A: C_A, C_B', g^b \pmod p$
  5. $A \rightarrow B: C_A, C_B g^{ab} \pmod p$\{A, signature on previous messages\}
  6. $B \rightarrow A: C_A, C_B, g^{ab} \pmod p$\{B, signature on previous messages\}

- Role of $C_A$, $C_B$, and messages
- Additional features: SPI selection
- Why not sign messages 3 & 4...?
Simple Key-Management for Internet Protocol (SKIP)

- Uses long term Diffie-Hellman keys
- Parties assumed to know each other public keys (i.e., $g^a \mod p$) or exchange certificates
- Session key $X = g^{ab} \mod p$ is established in 0 messages
- Each packet is encrypted using data key $S$ and each packet contains: $X\{S\}$
  - Same $S$ can be used for several packets
- Later on PFS was added by periodically forgetting the keys and doing a new DH
ISAKMP (RFC2408)

- Proposed by NSA as a framework and accepted by IETF
  - Runs over UDP and allows to exchange fields to create a protocol
- IKE (RFC2409) based on OAKLEY & SKEME using ISAKMP syntax

- IKE phases:
  1. Mutual authentication and session key establishment (also called ISAKMP SA or IKE SA)
  2. AH/ESP SAs establishment

  Each source/destination/port has its own SA/keys otherwise ESP traffic not using integrity could be decrypted...
Phase 1 IKE

Two modes:

- Aggressive mode: mutual authentication and session key establishment in three messages
  - $A \rightarrow B$: $g^a \mod p$, $A$, crypto proposal
  - $B \rightarrow A$: $g^b \mod p$, crypto choice, proof I’m $B$
  - $A \rightarrow B$: proof I’m $A$

- Main: additional features such as hiding end-points identities and negotiating crypto DH algorithm
  - $A \rightarrow B$: crypto suite I support
  - $B \rightarrow A$: crypto suite I choose
  - $A \rightarrow B$: $g^a \mod p$
  - $B \rightarrow A$: $g^b \mod p$
  - $A \rightarrow B$: $g^{ab} \mod p$ \{A, proof I’m $A$\}
  - $B \rightarrow A$: $g^{ab} \mod p$ \{B, proof I’m $B$\}
Phase 1 IKE

- Key types:
  - Pre-shared secret key
  - Public encryption key: fields are separately encrypted using the public key
  - Optimized public encryption key: used to encrypt a random symmetric key, and then data is encrypted using the symmetric key
  - Public signature key: used only for signature purpose

⇒ 8 variants of IKE phase 1: 2 modes x 4 key types

- Proof of Identity:
  - Required in messages 2-3 aggressive mode and 5-6 main mode
  - Proves the sender knows the key associated with the identity
  - Depends on the key type
  - Hash of identity key, DH values, nonces, crypto choices, cookies
  - Alternative: MAC of previous messages
Phase 1 IKE

- Negotiating cryptographic parameters
  - A specifies suites of acceptable algorithms:
    - \{(3DES, MD5, RSA public key encryption, DH),
      (AES, SHA-1, pre-shared key, elliptic curve), \ldots\}
  - The standard specifies a MUST be implemented set of algorithms:
    - Encryption=DES, hash=MD5/SHA-1, authentication=pre-shared key/DH
  - The lifetime of the SA can also be negotiated

- Session keys:
  - Key seed: SKEYID
    - Signature public keys: SKEYID = prf(nonces, \( g^x \mod p \))
    - Encryption public keys: prf(hash(nonces), cookies)
    - Pre-shared secret key: prf(pre-shared secret key, nonces)
  - Secret to generate other keys: SKEYID_d = prf(SKEYID, (g^x, cookies, 0))
  - Integrity key: SKEYID_a = prf(SKEYID, (SKEYID_d, (g^x, cookies, 1)))
  - Encryption key: SKEYID_e = prf(SKEYID, (SKEYID_a, (g^x, cookies, 2))

- Message IDs:
  - Random 32-bits serves the purpose of a SN but in an inefficient manner because they have to be remembered
IKE Phase 1: Public Signature Keys, Main Mode

Description:
- Both parties have public keys for signatures
- Hidden endpoint identity (except for …?)

Protocol:
- \( A \rightarrow B: CP \)
- \( B \rightarrow A: CPA \)
- \( A \rightarrow B: g^a \mod p, \text{nonce}_A \)
- \( B \rightarrow A: g^b \mod p, \text{nonce}_B \)
- \( K = f(g^{ab} \mod p, \text{nonce}_A, \text{nonce}_B) \)
- \( A \rightarrow B: K\{A, \text{proof I'm A, [certificate]}\} \)
- \( B \rightarrow A: K\{B, \text{proof I'm B, [certificate]}\} \)

Questions:
- What is the purpose of the nonces?
- Can we make to protocol shorter (5 messages)? At what expense?
IKE Phase 1:
Public Signature Keys, Aggressive Mode

- Protocol:
  - $A \rightarrow B$: $CP, \ g^a \mod \ p, \ \text{nonce}_A, \ A$
  - $B \rightarrow A$: $CPA, \ g^b \mod \ p, \ \text{nonce}_B, \ B, \ \text{proof I’m B}, \ [\text{certificate}]$
  - $A \rightarrow B$: proof I’m $A, \ [\text{certificate}]$
IKE Phase 1:
Public Encryption Keys, Main Mode, Original

Protocol:
- $A \rightarrow B$: $CP$
- $B \rightarrow A$: $CPA$
- $A \rightarrow B$: $g^a \mod p$, ${\text{nonce}_A}_B$, ${A}_B$
- $B \rightarrow A$: $g^b \mod p$, ${\text{nonce}_B}_A$, ${B}_A$

$$K = f(g^{ab} \mod p, \text{nonce}_A, \text{nonce}_B)$$

- $A \rightarrow B$: $K$[proof I’m A]
- $B \rightarrow A$: $K$[proof I’m B]
IKE Phase 1:
Public Encryption Keys, Aggressive Mode, Original

Protocol:
- $A \rightarrow B$: $CP$, $g^a \mod p$, $\{\text{nonce}_A\}_B$, $\{A\}_B$
- $B \rightarrow A$: $CPA$, $g^b \mod p$, $\{\text{nonce}_B\}_A$, $\{B\}_A$, proof I’m $B$
- $A \rightarrow B$: proof I’m $A$
IKE Phase 1:
Public Encryption Keys, Main Mode, Revised

Protocol:
- $A \rightarrow B$: CP
- $B \rightarrow A$: CPA

\[ K_A = \text{hash}(\text{nonce}_A, \text{cookie}_A) \]

- $A \rightarrow B$: \{nonce$_A$\}_B, $K_A$\{g$^a$ mod p\}, $K_A$\{A\}, [ $K_A$\{A's cert\} ]

\[ K_B = \text{hash}(\text{nonce}_B, \text{cookie}_B) \]

- $B \rightarrow A$: \{nonce$_B$\}_A, $K_B$\{g$^b$ mod p\}, $K_B$\{B\}.

\[ K = f( g^{ab} \mod p, \text{nonce}_A, \text{nonce}_B, \text{cookie}_A, \text{cookie}_B ) \]

- $A \rightarrow B$: $K$\{proof I’m A\}
- $B \rightarrow A$: $K$\{proof I’m B\}
IKE Phase 1:
Public Encryption Keys, Aggressive Mode, Revised

Protocol:

- $K_A = \text{hash}(\text{nonce}_A, \text{cookie}_A)$
- $A \rightarrow B$: CPA, $\{\text{nonce}_A\}_B$, $K_A\{g^a \mod p\}$, $K_A\{A\}$, [$K_A\{A's \text{ cert}\}$]
- $K_B = \text{hash}(\text{nonce}_B, \text{cookie}_B)$
- $B \rightarrow A$: CPA, $\{\text{nonce}_B\}_A$, $K_B\{g^b \mod p\}$, $K_B\{B\}$, proof I’m B
- $K = f(g^{ab} \mod p, \text{nonce}_A, \text{nonce}_B, \text{cookie}_A, \text{cookie}_B)$
- $A \rightarrow B$: $K\{\text{proof I’m A}\}$
IKE Phase 1:
Shared Secret Keys, Main Mode

- Assumption $A$ and $B$ share a secret $J$.
- Protocol:
  - $A \rightarrow B$: $CP$
  - $B \rightarrow A$: $CPA$
  - $A \rightarrow B$: $g^a \mod p$, nonce$_A$
  - $B \rightarrow A$: $g^b \mod p$, nonce$_B$
  - $K = f(J, g^{ab} \mod p, nonce_A, nonce_B, cookie_A, cookie_B)$
  - $A \rightarrow B$: $K$\{proof I’m $A$\}
  - $B \rightarrow A$: $K$\{proof I’m $B$\}
IKE Phase 1:
Shared Secret Keys, Aggressive Mode

- **Protocol:**
  - $A \rightarrow B$: $CP, g^a \mod p, \text{nonce}_A, A$
  - $B \rightarrow A$: $CPA, g^b \mod p, \text{nonce}_B, B, \text{proof I’m } B$
  - $A \rightarrow B$: proof I’m $A$
IKE: Phase 2

- Also known as “Quick Mode”: 3- messages protocol
  - $A \rightarrow B$: $X, Y, CP, traffic, SPI_A, nonce_A^e, [g^a \mod p]_{optional}$
  - $B \rightarrow A$: $X, Y, CPA, traffic, SPI_B, nonce_B, [g^b \mod p]_{optional}$
  - $A \rightarrow B$: $X, Y, ack$

- All messages are encrypted using SKEYID_e, and integrity protected using SKEYID_a (except $X, Y$)

- Parameters:
  - $X$: pair of cookies generated during phase 1
  - $Y$: 32-bit number unique to this phase 2 session chosen by the initiator
  - CP: Crypto Proposal, CPA: Crypto Proposal Accepted
  - DH is optional and could be used to provide PFS
  - Nonces and cookies get shuffled into SKEYID to produce the SA encryption and integrity keys