CSC 774 -- Network Security

Topic 5: Internet Key Management

Outline

• Key Management
  – Security Principles
  – Center-based Key Management
  – Certificate-based Key Management

• Internet Key Management
  – Manual Exchange
  – SKIP
  – Oakley
  – ISAKMP
Key Management

- Why do we need key management
  - AH and ESP require encryption and authentication keys
- Process to negotiate and establish IPSec SA’s between two entities

Security Principles

- Basic security principle for session keys
  - Compromise of a session key
    - Doesn’t permit reuse of the compromised session key.
    - Doesn’t compromise future session keys and long-term keys.
Security Principles (Cont’d)

- Perfect forward secrecy (PFS)
  - Compromise of current keys (session key or long-term key) doesn’t compromise past session keys.
  - Concern for encryption keys but not for authentication keys.
  - Not really “perfect” in the same sense as perfect secrecy for one-time pad.

Center-Based Key Management

- Key Distribution Center (KDC)
  - Communication parties depend on KDC to establish a pair-wise key.
  - The KDC generates the cryptographic key
  - Pull based
    - Alice communicates with the KDC before she communicates with Bob
  - Push based
    - Alice communicates with Bob, and it’s Bob’s responsibility to contact the KDC to get the pair-wise key.
Center-Based Key Management (Cont’d)

- Key Translation Center (KTC)
  - Similar to KDC
  - Difference
    - One of the participants generates the cryptographic; KTC only translates and forwards it to the other participant.

An Example of KDC: Kerberos

1. Request TGT
2. TGT + session key
3. Request SGT
4. Ticket + session key
5. Request service
6. Server authenticator

Keberos
Authentication Server (AS)
Ticket-Granting Server (TGS)

Server
Certificate-Based Key Distribution

- Public key certificate
  - Bind an identity and a public key together
  - Verify the authenticity of a party’s public key
- Key transfer
  - Transfer a locally generated pair-wise key to a remote entity
- Key exchange (agreement)
  - Both communication parties generate a key cooperatively.
  - Example: Diffie-Hellman key exchange

Internet Key Management

- Two major competing proposals
  - Simple Key Management for Internet Protocols (SKIP)
  - ISAKMP/OAKLEY
    - Photuris
      - Ephemeral D-H + authentication + Cookie
      - The first to use cookie to thwart DOS attacks
    - SKEME
    - Oakley
    - ISAKMP
    - ISAKMP/OAKLEY [IKE]
Manual Key Management

- Mandatory
- Useful when IPSec developers are debugging
- Keys exchanged offline (phone, email, etc.)
- Set up SPI and negotiate parameters

Automatic Key Management

- Key Distribution and Management combined
  - SKIP
- Key establishment protocol
  - Oakley
    - focus on key exchange
- Key management
  - Internet Security Association & Key Management Protocol (ISAKMP)
    - Focus on SA and key management
    - Clearly separated from key exchange.
• Idea
  – IP is connectionless in nature
  – Using security association forces a pseudo session layer underneath IP
  – Proposal: use session-less key establishment and management
    • Pre-distributed and authenticated D-H public key
    • Packet-specific encryption keys are included in the IP packets

**SKIP (Cont’d)**

Certificate repository

Bob’s certificate  Alice’s certificate

Alice  Bob

$K_p$ encrypted with KEK.  Payload encrypted with $K_p$. 
SKIP (Cont’d)

• KEK should be changed periodically
  – Minimize the exposure of KEK
  – Prevent the reuse of compromised packet keys

• SKIP’s approach
  – KEK = \( h(K_{AB}, n) \), where \( h \) is a one-way hash function, \( K_{AB} \) is the long term key between A and B, and \( n \) is a counter.

SKIP (Cont’d)

• Limitations
  – No Perfect Forward Secrecy
    • Can be modified to provide PFS, but it will lose the sessionless property.
  – No concept of SA; difficult to work with the current IPSec architecture

• Not the standard, but remains as an alternative.
Oakley

- Oakley is a refinement of the basic Diffie-Hellman key exchange protocol.
- Why need refinement?
  - Resource clogging attack
  - Replay attack
  - Man-in-the-middle attack
  - Choice of D-H groups

Resource Clogging Attack

- Stopping requests is difficult
  - We need to provide services.
- Ignoring requests is dangerous
  - Denial of service attacks
Resource Clogging Attack (Cont’d)

- **Counter measure**
  - If we cannot stop bogus requests, at least we should know from where the requests are sent.
  - **Cookie**
    - Each side sends a pseudo-random number, the cookie, in the initial message, which the other side acknowledges.
    - The acknowledgement must be repeated in the following messages.
    - Do not begin D-H calculation until getting acknowledgement for the other side.
    - Cookies are used to thwart resource clogging attack
      - Thwart, not prevent.

Requirements for cookie generation

- The cookie must depend on the specific parties.
  - Prevent an attacker from reusing cookies.
- Impossible to forge
  - Use secret values
- Efficient
- Cookies are also used for key naming
  - Each key is uniquely identified by the initiator’s cookie and the responder’s cookie.
Replay Attack

- Counter measure
  - Use nonce

1. Cookie exchange
2. Later exchange
3. Replay
4. Busy computing

Man-in-the-middle-attack

- Counter measure
  - Authentication
  - Depend on other mechanisms.
    - Pre-shared key.
    - Public key certificates.
Oakley Groups

- 0  no group (placeholder or non-DH)
- 1  MODP, 768-bit modulus
- 2  MODP, 1024-bit modulus
- 3  MODP, 1536-bit modulus
- 4  EC2N over GF(2^{155})
- 5  EC2N over GF(2^{185})

Ephemeral Diffie-Hellman

- Session key is computed on the basis of short-term DH public-private keys.
- Exchange of these short-term public keys requires authentication and integrity.
  - Digital signatures.
  - Keyed message digests.
- The only protocol known to support Perfect Forward Secrecy.
ISAKMP

- Oakley
  - Key exchange protocol
  - Developed to use with ISAKMP
- ISAKMP
  - Security association and key management protocol
  - Defines procedures and packet formats to establish, negotiate, modify, and delete security associations.
  - Defines payloads for security association, key exchange, etc.

ISAKMP Message

- Fixed format header
  - 64 bit initiator and responder cookies
  - Exchange type (8 bits)
  - Next payload type (8 bits)
  - Flags: encryption, commit, authentication, etc.
  - 32 bit message ID
    - Resolve multiple phase 2 SAs being negotiated simultaneously
  - Variable number of payloads
    - Each has a generic header with
      - Payload boundaries
      - Next payload type (possible none)
ISAKMP Formats

(a) ISAKMP Header

(b) Generic Payload Header

ISAKMP Phases

• Phase 1
  – Establish ISAKMP SA to protect further ISAKMP exchanges
  – Or use pre-established ISAKMP SA
  – ISAKMP SA identified by initiator cookie and responder cookie

• Phase 2
  – Negotiate security services in SA for target security protocol or application.
ISAKMP

- Disadvantage
  - Additional overhead due to 2 phases
- Advantages
  - Same ISAKMP SA can be used to negotiate phase 2 for multiple protocols
  - ISAKMP SA can be used to facilitate maintenance of SAs.
  - ISAKMP SA can simplify phase 2.

ISAKMP Domain Of Interpretation (DOI)

- DOI defines
  - Payload format
  - Exchange types
  - Naming conventions for security policies, cryptographic algorithms
- DOI for IPSEC has been defined.
ISAKMP Exchange Types

- 0  none
- 1  base
- 2  identity protection
- 3  authentication only
- 4  aggressive
- 5  informational
- 6-31 reserved
- 32-239 DOI specific use
- 240-255 private use

ISAKMP Exchange Types

- Base exchange
  - reveals identities
- Identity protection exchange
  - Protects identities at cost of extra messages.
- Authentication only exchange
  - No key exchange
- Aggressive exchange
  - Reduce number of message, but reveals identity
- Informational exchange
  - One-way transmission of information.
### ISAKMP Payload Types

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>SA security association</td>
</tr>
<tr>
<td>2</td>
<td>P proposal</td>
</tr>
<tr>
<td>3</td>
<td>T transform</td>
</tr>
<tr>
<td>4</td>
<td>KE key exchange</td>
</tr>
<tr>
<td>5</td>
<td>ID identification</td>
</tr>
<tr>
<td>6</td>
<td>CERT certificate</td>
</tr>
<tr>
<td>7</td>
<td>CR certificate request</td>
</tr>
<tr>
<td>8</td>
<td>H hash</td>
</tr>
<tr>
<td>9</td>
<td>SIG signature</td>
</tr>
<tr>
<td>10</td>
<td>NONCE nonce</td>
</tr>
<tr>
<td>11</td>
<td>N notification</td>
</tr>
<tr>
<td>12</td>
<td>D delete</td>
</tr>
<tr>
<td>13</td>
<td>VID vender ID</td>
</tr>
<tr>
<td>14-127</td>
<td>reserved</td>
</tr>
<tr>
<td>128-255</td>
<td>private use</td>
</tr>
</tbody>
</table>
ISAKMP Payload Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Association (SA)</td>
<td>Domain of Interpretation, Situation</td>
<td>Used to negotiate security attributes and indicate the DOI and Situation under which negotiation is taking place.</td>
</tr>
<tr>
<td>Proposal (P)</td>
<td>Proposal ID, Protocol-ID, SPI Size, # of Transforms, SPI</td>
<td>Used during SA negotiation, indicates protocol to be used and number of transforms.</td>
</tr>
<tr>
<td>Transform (T)</td>
<td>Transform ID, Transforms-ID, SA Attributes</td>
<td>Used during SA negotiation, indicates transform and related SA attributes.</td>
</tr>
<tr>
<td>Key Exchange (KE)</td>
<td>Key Exchange Data</td>
<td>Supports a variety of key exchange techniques.</td>
</tr>
<tr>
<td>Identification (ID)</td>
<td>ID Type, ID Data</td>
<td>Used to exchange identification information.</td>
</tr>
<tr>
<td>Certificate (CERT)</td>
<td>Cert Encoding, Certificate Data</td>
<td>Used to transport certificates and other certificate-related information.</td>
</tr>
<tr>
<td>Certificate Request (CR)</td>
<td>Cert Types, Certificate Types, Cert Auths, Certificate Authorities</td>
<td>Used to request certificates, indicates the types of certificates requested and the acceptable certificate authorities.</td>
</tr>
<tr>
<td>Hash (HASH)</td>
<td>Hash Data</td>
<td>Contains data generated by a hash function.</td>
</tr>
<tr>
<td>Signature (SIG)</td>
<td>Signature Data</td>
<td>Contains data generated by a digital signature function.</td>
</tr>
<tr>
<td>Nonce (NONCE)</td>
<td>Nonce Data</td>
<td>Contains a nonce.</td>
</tr>
<tr>
<td>Notification (N)</td>
<td>DOI, Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data</td>
<td>Used to transmit notification data, such as an error condition.</td>
</tr>
<tr>
<td>Delete (D)</td>
<td>DOI, Protocol-ID, SPI Size, # of SPIs, SPI (one or more)</td>
<td>Indicates an SA that is no longer valid.</td>
</tr>
</tbody>
</table>

### ISAKMP Exchanges

#### (a) Base Exchange

1. \(I \rightarrow R: \text{SA, NONCE}\) Begin ISAKMP/SA negotiation
2. \(R \rightarrow I: \text{SA, NONCE}\) Basic SA agreed upon
3. \(I \rightarrow R: \text{KE, ID}_{I}, \text{AUTH}\) Key generated; Initiator identity verified by responder
4. \(R \rightarrow I: \text{KE, ID}_{E}, \text{AUTH}\) Responder identity verified by initiator; Key generated; SA established

#### (b) Identity Protection Exchange

1. \(I \rightarrow R: \text{SA}\) Begin ISAKMP/SA negotiation
2. \(R \rightarrow I: \text{SA}\) Basic SA agreed upon
3. \(I \rightarrow R: \text{KE, NONCE}\) Key generated
4. \(R \rightarrow I: \text{KE, NONCE}\) Key generated
5. * \(I \rightarrow R: \text{ID}_{E}, \text{AUTH}\) Initiator identity verified by responder
6. * \(R \rightarrow I: \text{ID}_{E}, \text{AUTH}\) Responder identity verified by initiator; SA established

#### (c) Authentication Only Exchange

1. \(I \rightarrow R: \text{SA, NONCE}\) Begin ISAKMP/SA negotiation
2. \(R \rightarrow I: \text{SA, NONCE, ID}_{E}, \text{AUTH}\) Basic SA agreed upon; Responder identity verified by initiator
3. \(I \rightarrow R: \text{ID}_{E}, \text{AUTH}\) Initiator identity verified by responder; SA established

#### (d) Aggressive Exchange

1. \(I \rightarrow R: \text{SA, KE, NONCE, ID}_{E}\) Begin ISAKMP/SA negotiation and key exchange
2. \(R \rightarrow I: \text{SA, KE, NONCE, ID}_{E}, \text{AUTH}\) Initiator identity verified by responder; Key generated; Basic SA agreed upon
3. * \(I \rightarrow R: \text{AUTH}\) Responder identity verified by initiator; SA established

#### (e) Informational Exchange

1. * \(I \rightarrow R: \text{N/D}\) Error or status notification, or deletion

Notation:

- \(I\) = initiator
- \(R\) = responder
- * = signifies payload encryption after the ISAKMP header