Topic 4.3: Internet Key Management

Outline

- Key Management
  - Security Principles
- Internet Key Management
  - Manual Exchange
  - SKIP
  - Oakley
  - ISAKMP
  - IKE
Key Management

- Why do we need Internet key management
  - AH and ESP require encryption and authentication keys
- Process to negotiate and establish IPsec SAs 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.

Internet Key Management

• Manual key management
  – Mandatory
  – Useful when IPsec developers are debugging
  – Keys exchanged offline (phone, email, etc.)
  – Set up SPI and negotiate parameters
Internet Key Management (Cont’d)

- Automatic 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 (extension to Photuris)
    - Oakley (RFC 2412)
    - ISAKMP (RFC 2408)
    - ISAKMP/OAKLEY → IKE (RFC 2409)

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 sessionless 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)

Two types of keys:
1. KEK
2. Packet key

Certificate repository

Bob’s certificate
Alice’s certificate

Alice ————> Bob

K_p encrypted with KEK.
Payload encrypted with K_p.

NC STATE UNIVERSITY  Computer Science  CSC 474  Dr. Peng Ning  10
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 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.
  – Cookies are used to thwart resource clogging attack
    • Thwart, not prevent

• 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.
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

- How to choose the DH 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.

Ephemeral Diffie-Hellman

- Question: What happens if the long term key is compromised?
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

![ISAKMP Header Diagram]

![Generic Payload Header Diagram]

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, Transform-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 a SA that is no longer valid.</td>
</tr>
</tbody>
</table>

### ISAKMP Exchanges

#### Basic Exchange

1. **I→R**: SA; NONCE  
   - Begin ISAKMP-SA negotiation

2. **R→I**: SA; NONCE  
   - Basic SA agreed upon

3. **I→R**: KE; ID_I; AUTH  
   - Key generated; Initiator id verified by responder

4. **R→I**: KE; ID_R; AUTH  
   - Responder id verified by initiator; key generated; SA established
### ISAKMP Exchanges (Cont’d)

#### Identify Protection Exchange

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>I→R: SA</td>
<td>• Begin ISAKMP-SA negotiation</td>
</tr>
<tr>
<td>2.</td>
<td>R→I: SA</td>
<td>• Basic SA agreed upon</td>
</tr>
<tr>
<td>3.</td>
<td>I→R: KE; NONCE</td>
<td>• Key generated;</td>
</tr>
<tr>
<td>4.</td>
<td>R→I: KE; NONCE</td>
<td>• Key generated;</td>
</tr>
<tr>
<td>5.</td>
<td>I→R: ID; AUTH</td>
<td>• Initiator id verified by responder</td>
</tr>
<tr>
<td>6.</td>
<td>R→I: ID; AUTH</td>
<td>• Responder id verified by initiator; SA established</td>
</tr>
</tbody>
</table>

Red messages: Payload encrypted after ISAKMP header

#### Authentication Only Exchange

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>I→R: SA; NONCE</td>
<td>• Begin ISAKMP-SA negotiation</td>
</tr>
<tr>
<td>2.</td>
<td>R→I: SA; NONCE; ID; AUTH</td>
<td>• Basic SA agreed upon; Responder id verified by initiator</td>
</tr>
<tr>
<td>3.</td>
<td>I→R: ID; AUTH</td>
<td>• Initiator id verified by responder; SA established</td>
</tr>
</tbody>
</table>
ISAKMP Exchanges (Cont’d)

Aggressive Exchange

1. $I \rightarrow R$: SA; KE; NONCE; $ID_I$
   - Begin ISAKMP-SA negotiation and key exchange

2. $R \rightarrow I$: SA; KE; NONCE; $ID_R$; AUTH
   - Responder identity verified by responder; Key generated; Basic SA agreed upon;

3. $I \rightarrow R$: AUTH
   - Initiator id verified by responder; SA established

Red messages: Payload encrypted after ISAKMP header

---

ISAKMP Exchanges (Cont’d)

Informational Exchange

1. $I \rightarrow R$: N/D
   - Error or status notification, or deletion.

Red message: Payload encrypted after ISAKMP header
IKE Overview

• IKE = ISAKMP + part of OAKLEY + part of SKEME
  – ISAKMP determines
    • How two peers communicate
    • How these messages are constructed
    • How to secure the communication between the two peers
    • No actual key exchange
  – Oakley
    • Key exchange protocol
  – Combining these two requires a Domain of Interpretation (DOI)
    • RFC 2407

IKE Overview (Cont’d)

• A separate RFC has been published for IKE
  – RFC 2409
• Request-response protocol
  – Initiator
  – Responder
• Two phases
  – Phase 1: Establish an IKE (ISAKMP) SA
    • Essentially the ISAKMP phase 1
    • Bi-directional
  – Phase 2: Use the IKE SA to establish IPsec SAs
    • Key exchange phase
    • Directional
IKE Overview (Cont’d)

• Several Modes
  – Phase 1:
    • Main mode: identity protection
    • Aggressive mode
  – Phase 2:
    • Quick mode
  – Other modes
    • New group mode
      – Establish a new group to use in future negotiations
      – Not in phase 1 or 2;
      – Must only be used after phase 1
    • Informational exchanges
      – ISAKMP notify payload
      – ISAKMP delete payload

IPSEC Architecture Revisited

IKE policies (How to establish the IPsec SAs):
1. Encryption algorithm; 2. Hash algorithm;
A Clarification About PFS

- In RFC 2409:
  - When used in the memo Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key.
  - The key used to protect transmission of data MUST NOT be used to derive any additional keys.
  - If the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys.
- Is this consistent with what we discussed?

IKE Phase 1

- Four authentication methods
  - Digital signature
  - Authentication with public key encryption
  - The above method revised
  - Authentication with a pre-shared key
IKE Phase 1 (Cont’d)

- IKE Phase 1 goal:
  - Establish a shared secret SKEYID
  - With signature authentication
    - $$\text{SKEYID} = \text{prf}(N_i_b \mid N_r_b, g^{xy})$$
  - With public key encryption
    - $$\text{SKEYID} = \text{prf}(\text{hash}(N_i_b \mid N_r_b), \text{CKY-I} \mid \text{CKY-R})$$
  - With pre-shared key
    - $$\text{SKEYID} = \text{prf}($$pre-shared-key, $$N_i_b \mid N_r_b)$$
  - Notations:
    - prf: keyed pseudo random function $$\text{prf}(\text{key}, \text{message})$$
    - CKY-I/CKY-R: I’s (or R’s) cookie
    - $$N_i_b/N_r_b:$$ the body of I’s (or R’s) nonce

IKE Phase 1 (Cont’d)

- Three groups of keys
  - Derived key for non-ISAKMP negotiations
    - $$\text{SKEYID}_d = \text{prf}(\text{SKEYID}, g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 0)$$
  - Authentication key
    - $$\text{SKEYID}_a = \text{prf}(\text{SKEYID}, \text{SKEYID}_d \mid g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 1)$$
  - Encryption key
    - $$\text{SKEYID}_e = \text{prf}(\text{SKEYID}, \text{SKEYID}_a \mid g^{xy} \mid \text{CKY-I} \mid \text{CKY-R} \mid 2)$$
IKE Phase 1 (Cont’d)

• To authenticate the established key
  – Initiator generates
    • \( \text{HASH}_I = \text{prf}(\text{SKEYID}, g^{xi} | g^{xr} | \text{CKY-I} | \text{CKY-R} | \text{SAi}_b | \text{IDii}_b) \)
  – Responder generates
    • \( \text{HASH}_R = \text{prf}(\text{SKEYID}, g^{xr} | g^{xi} | \text{CKY-R} | \text{CKY-I} | \text{SAi}_b | \text{IDir}_b) \)
  – Authentication with digital signatures
    • \( \text{HASH}_I \) and \( \text{HASH}_R \) are signed and verified
  – Public key encryption or pre-shared key
    • \( \text{HASH}_I \) and \( \text{HASH}_R \) directly authenticate the exchange.

IKE Phase 1 Authenticated with Signatures

Main Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA</td>
<td>HDR, SA</td>
</tr>
<tr>
<td>HDR*, KE, Ni</td>
<td>HDR, KE, Nr</td>
</tr>
</tbody>
</table>
IKE Phase 1 Authenticated with Signatures

Aggressive Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA, KE, Ni, IDii</td>
<td>HDR, SA, KE, Nr, IDir, [CERT,] SIG_R</td>
</tr>
<tr>
<td>HDR, [CERT,] SIG_I</td>
<td></td>
</tr>
</tbody>
</table>

IKE Phase 1 Authenticated with Public Key Encryption

Main Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA</td>
<td>HDR, SA</td>
</tr>
<tr>
<td>HDR, KE, [HASH(1),]</td>
<td></td>
</tr>
<tr>
<td>&lt;IDii_b&gt;PubKey_r, &lt;Ni_b&gt;PubKey_r</td>
<td>HDR, KE, &lt;IDir_b&gt;PubKey_i, &lt;Nr_b&gt;PubKey_i</td>
</tr>
<tr>
<td>HDR*, HASH_I</td>
<td>HDR*, HASH_R</td>
</tr>
</tbody>
</table>


IKE Phase 1 Authenticated with Public Key Encryption

Aggressive Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA, [HASH(1), ] KE, &lt;IDii_b&gt;PubKey_r,</td>
<td>HDR, SA, KE, &lt;IDir_b&gt;PubKey_i, &lt;Nr_b&gt;PubKey_i,</td>
</tr>
<tr>
<td>&lt;Ni_b&gt;PubKey_r</td>
<td>HASH_R</td>
</tr>
<tr>
<td>HDR, HASH_I</td>
<td></td>
</tr>
</tbody>
</table>

Observations

- Authenticated using public key encryption
  - No non-repudiation
    - No evidence that shows the negotiation has taken place.
  - More difficult to break
    - An attacker has to break both DH and public key encryption
  - Identity protection is provided in aggressive mode.
  - Four public key operations
    - Two public key encryptions
    - Two public key decryptions
### IKE Phase 1 Authenticated with A Revised Mode of Public Key Encryption

#### Main Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA</td>
<td>HDR, SA</td>
</tr>
<tr>
<td>HDR, [HASH(1),]</td>
<td></td>
</tr>
<tr>
<td>&lt;Ni_b&gt;PubKey_r</td>
<td></td>
</tr>
<tr>
<td>&lt;KE_b&gt;Ke_i</td>
<td></td>
</tr>
<tr>
<td>&lt;IDii_b&gt;Ke_i,</td>
<td></td>
</tr>
<tr>
<td>[Cert-I_b]=Ke_i</td>
<td></td>
</tr>
<tr>
<td>HDR*, HASH_I</td>
<td>HDR*, HASH_R</td>
</tr>
</tbody>
</table>

#### Aggressive Mode

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA, [HASH(1),]</td>
<td>HDR, SA,</td>
</tr>
<tr>
<td>&lt;Ni_b&gt;PubKey_r,</td>
<td>HDR, &lt;Nr_b&gt;PubKey_i,</td>
</tr>
<tr>
<td>&lt;KE_b&gt;Ke_i,</td>
<td>&lt;KE_b&gt;Ke_r,</td>
</tr>
<tr>
<td>&lt;IDii_b&gt;Ke_i</td>
<td>&lt;IDir_b&gt;Ke_r</td>
</tr>
<tr>
<td>[, &lt;Cert-I_b&gt;=Ke_i]</td>
<td>HASH_R</td>
</tr>
<tr>
<td>HDR, HASH_I</td>
<td></td>
</tr>
</tbody>
</table>
Further Details

\[ Ne_i = \text{prf}(Ni_b, \text{CKY-I}) \]
\[ Ne_r = \text{prf}(Nr_b, \text{CKY-R}) \]

- Ke_i and Ke_r are taken from Ne_i and Ne_r, respectively.

IKE Phase 1 Authenticated with Pre-Shared Key

<table>
<thead>
<tr>
<th>Main Mode</th>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HDR, SA</td>
<td>HDR, SA</td>
</tr>
<tr>
<td></td>
<td>HDR, KE, Ni</td>
<td>HDR, KE, Nr</td>
</tr>
<tr>
<td></td>
<td>HDR*, IDi, HASH_I</td>
<td>HDR*, IDir, HASH_R</td>
</tr>
</tbody>
</table>
IKE Phase 1 Authenticated with Pre-Shared Key (Cont’d)

- What provide the authentication?
- Why does it work?

IKE Phase 1 Authenticated with Pre-Shared Key

**Aggressive Mode**

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA, KE, Ni, IDii</td>
<td>HDR, SA, KE, Nr, IDir, HASH_R</td>
</tr>
<tr>
<td>HDR, HASH_I</td>
<td></td>
</tr>
</tbody>
</table>
IKE Phase 2 -- Quick Mode

- Not a complete exchange itself
  - Must be bound to a phase 1 exchange
- Used to derive keying materials for IPsec SAs
- Information exchanged with quick mode must be protected by the ISAKMP SA
- Essentially a SA negotiation and an exchange of nonces
  - Generate fresh key material
  - Prevent replay attack

IKE Phase 2 -- Quick Mode (Cont’d)

- Basic Quick Mode
  - Refresh the keying material derived from phase 1
- Quick Mode with optional KE payload
  - Transport additional exponentiation
  - Provide PFS
IKE Phase 2 -- Quick Mode (Cont’d)

If PFS is not needed, and KE payloads are not exchanged, the new keying material is defined as

\[
\text{KEYMAT} = \text{prf(SKEYID}\_d, \text{ protocol | SPI | Ni}_b | \text{ Nr}_b)
\]

If PFS is desired and KE payloads were exchanged, the new keying material is defined as

\[
\text{KEYMAT} = \text{prf(SKEYID}\_d, g(qm)^{|g} | \text{ protocol | SPI | Ni}_b | \text{ Nr}_b)
\]

where \(g(qm)^{|g}\) is the shared secret from the ephemeral Diffie-Hellman exchange of this Quick Mode.

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

---

IKE Phase 2 -- Quick Mode (Cont’d)

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR*, HASH(3)</td>
<td></td>
</tr>
</tbody>
</table>