Application of Cryptography: IPsec & IKE

Guevara Noubir
http://www.cs.uu.edu/home/noubir/Courses/CIS235/94
Textbook: "Network Security",
Kaufman - Perlman - Speciner
Reading: Chapters 17-18

Outline
n Introduction to secure networking
  n IPsec:
    o Authentication Header and Encapsulated Secure Payload
  n Internet Key Exchange (IKE)

Approach to Secure Networking
n Key Distribution Centers: Trusted Third Parties
  or
n Public Key Systems + Public Key Infrastructure
Key Distribution Center

- Solve the scalability problem of a set of $n$ nodes using secret keys $n^2(n+1)/2$ keys
- New nodes are configured with a key to the KDC
  - e.g., $K_n$ for node $A$
- If node $A$ wants to communicate with node $B$
  - $A$ sends a request to the KDC
  - The KDC securely sends to $A$: $E_{K_n}(R_{AB})$ and $E_{K_n}(R_{BA}, A, B)$
- Advantage:
  - Single location for updates, single key to be remembered
- Drawbacks:
  - If the KDC is compromised
  - Single point of failure/performance bottleneck $\Rightarrow$ multiple KDCs?
- Example of systems: Kerberos

Public Keys and Certification Authorities

- How do you know the public key of a node?
- Typical solution:
  - Use a trusted node as a certification authority (CA)
  - The CA generates certificates: Signed(A, public key, valid information)
  - Every body needs to know the CA public key
  - Certificates can be stored in a directory service or exchanged during the authentication process
- Advantages:
  - The CA doesn't have to be online $\Rightarrow$ more physical protection
  - Not a performance bottleneck, not a single point of failure
  - Certificates are not security sensitive: only threat is DoS
  - A compromised CA cannot decrypt conversation but can lead to impersonation
  - A certification hierarchy can be used: e.g., X.509

Securing Network Stacks

- Where to put the security in a protocol stack?
- Practical considerations:
  - End to end security
  - No modification to OS/network stack

<table>
<thead>
<tr>
<th>Application Layer</th>
<th>Security</th>
<th>Transport Layer (TCP)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL/TLS, SSH</td>
<td></td>
<td>(IPSec, IKE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network Layer (IP)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Link Layer (IEEE802.11)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(spread-spectrum, quantum crypto, etc.)</td>
<td></td>
</tr>
</tbody>
</table>
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

Some Issues with Real-time Communication

- Session key establishment
- Perfect Forward Secrecy
  - Diffie-Hellman based PFS
  - Elman-Okaru:
    - If keys are compromised, Diffie-Hellman protects against passive attacks
  - Signature keys are usually not reserved
- Preventing Denial of Service
  - SYN attack on TCP: use stateless cookies = hash(IP addr, secret)
  - Puzzles: e.g., what is the 27-bit number has an MD = x?
  - These techniques do not fully protect against DDOS launched through viruses
- Hiding endpoint identity:
  - DH + authentication allows anonymous connection or detects man-in-the-middle
- Live partner reassurance:
  - Modify DH to include a nonce in the computation of the session key
  - Optimization using parallel computation, session resumption, deniability

IPsec Protocol Suite (IETF Standard)

- Provides inter-operable cryptographically based security services:
  - Services: confidentiality, authentication, integrity, and key management
- Protocols:
  - Authentication Header (AH): RFC2402
  - Encapsulated Security Payload (ESP): RFC2406
  - Internet Key Exchange (IKE)
- Environments: IPv4 and IPv6
- Modes:
  - Transport (between two hosts)
  - Tunnel (between hosts/firewalls)
IPsec

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

AH Formatting

AH Protocol Number = 51
ESP Formatting

ESP Protocol Number = 50

<table>
<thead>
<tr>
<th>IP hdr</th>
<th>ESP</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP hdr</td>
<td>TCP</td>
<td>Data</td>
</tr>
<tr>
<td>IP hdr</td>
<td>ESP</td>
<td>Data</td>
</tr>
<tr>
<td>IP hdr</td>
<td>ESP</td>
<td>Data</td>
</tr>
</tbody>
</table>

Transport mode

Tunneled mode

ESP Header

Security Protocol Index (0-255)
Authentication Parameter Index (0-255)
Encryption Parameter Index (0-255)
Authentication Data
Encryption Data
Padding (0-255 bytes)

ESP Protocol Number = 50

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
- $A \rightarrow B; C_A$
- $B \rightarrow A; C_B, C_B^* \text{ crypto offered}$
- $A \rightarrow B; C_B, C_B^* \text{ Crypto selected}$
- $B \rightarrow A; C_B, g^a \text{ mod } p$
- $A \rightarrow B; C_B, g^a \text{ mod } p \text{ (A's signature of previous message)}$
- $B \rightarrow A; C_B, g^b \text{ mod } p \text{ (B's signature of previous message)}$
- Role of $C_B, C_B^*$ 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., $g^x \text{ mod } p$) or exchange certificates
- Session key $X = g^{ab} \text{ mod } p$ 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:
  - Mutual authentication and session key establishment (also called ISAKMP SA or IKE SA)
  - AH/ESP SAs establishment
  - Each source/destination/port has its own SA/keys otherwise ESP traffic not using integrity could be decrypted
- IKE uses default port 500
Phase 1 IKE

Two modes:
- Aggressive mode: mutual authentication and session key establishment in three messages
  - $A \rightarrow B$: $g^m \pmod{p}$, crypto proposal
  - $B \rightarrow A$: $g^n \pmod{p}$, crypto choice, proof I'm $A$
  - $A \rightarrow B$: 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
  - $A \rightarrow B$: $g^m \pmod{p}$
  - $B \rightarrow A$: $g^n \pmod{p}$
  - $A \rightarrow B$: $g^m \pmod{p}$ ($A$, proof I'm $A$)
  - $B \rightarrow A$: 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 specific suite of acceptable algorithms:
  - (SHA, AES, MD5, RSA as key exchange, DH, CRAM-MD5, MD5, ECDH)
- The standard specifies a MUST be implemented set of algorithms:
  - Encryption/MD5, hash/MD5, authentication-pin-based key derivation
- The lifetime of the SA can also be negotiated

Session keys:
- Key seed: SKYIELD
  - Signature public keys: SKYIELD = $g^s \pmod{p}$, prime $p$.
  - Encryption public keys: $p$ (transaction), signature, and session key.
- Secret to generate other keys: $SKYIELD \_sk = prf(SKYIELD, (g^s, cookies, 0))$
- Integrity key: $SKYIELD \_i = prf(SKYIELD \_sk, (SKYIELD \_a, (g^a, cookies, 0)))$
- Encryption key: $SKYIELD \_e = prf(SKYIELD \_sk, (SKYIELD \_a, (g^e, cookies, 2)))$
- Message ID:
  - Random 32-bits serves the purpose of a SA 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:
- \( A \rightarrow B \): \( CP \)
- \( B \rightarrow A \): \( CPA \)
- \( A \rightarrow B \): \( g^r \mod p \), \( \text{nonce}_A \)
- \( B \rightarrow A \): \( g^r \mod p \), \( \text{nonce}_B \)
- \( K = (g^r \mod p \text{ nonce}_A \text{ nonce}_B) \)
- \( A \rightarrow B \): \( K_{\text{proof I'm } A} \) (certificate)
- \( B \rightarrow A \): \( K_{\text{proof I'm } B} \) (certificate)

Questions:
- What is the purpose of the nonces?
- Can we make this protocol shorter (5 messages)? At what expense?

IKE Phase 1: Public Signature Keys, Aggressive Mode

Protocol:
- \( A \rightarrow B \): \( CP \), \( g^r \mod p \text{ nonce}_A \text{ A} \)
- \( B \rightarrow A \): \( CPA \), \( g^r \mod p \text{ nonce}_B \text{ B} \text{ proof I'm } B \)
- \( A \rightarrow B \): \( K \text{ proof I'm } A \) (certificate)
- \( B \rightarrow A \): \( K \text{ proof I'm } B \) (certificate)

IKE Phase 1: Public Encryption Keys, Main Mode, Original

Protocol:
- \( A \rightarrow B \): \( CP \)
- \( B \rightarrow A \): \( CPA \)
- \( A \rightarrow B \): \( g^r \mod p \text{ nonce}_A \text{ (A)} \text{ B}_A \)
- \( B \rightarrow A \): \( g^r \mod p \text{ nonce}_B \text{ (B)} \text{ A} \)
- \( K = (g^r \mod p \text{ nonce}_A \text{ nonce}_B) \)
- \( A \rightarrow B \): \( K \text{ (proof I'm } A) \)
- \( B \rightarrow A \): \( K \text{ (proof I'm } B) \)
IKE Phase 1:
Public Encryption Keys, Aggressive Mode, Original

Protocol:
- $A \rightarrow B$: $CP, g^r \mod n, \{\text{nonce}_A\}_A, \{A\}_B$
- $B \rightarrow A$: $CP, g^r \mod n, \{\text{nonce}_B\}_B, \{B\}_A$, proof I'm $B$
- $A \rightarrow B$: proof I'm $A$

IKE Phase 1:
Public Encryption Keys, Main Mode, Revised

Protocol:
- $A \rightarrow B$: $CP$
- $B \rightarrow A$: $CPA$
  $K_A = \text{hash}(\text{nonce}_A, \text{cookie}_A)$
  $A \rightarrow B$: $\{\text{nonce}_B\}_B, K_A(g^r \mod n), K_A(A), [K_A(A\text{cert})]$  
  $K_B = \text{hash}(\text{nonce}_B, \text{cookie}_B)$
  $B \rightarrow A$: $\{\text{nonce}_A\}_A, K_B(g^r \mod n), K_B(B)$
  $K = (g^m \mod n, \text{nonce}_A, \text{nonce}_B, \text{cookie}_A, \text{cookie}_B)$
  $A \rightarrow B$: $K(\text{proof I'm A})$
  $B \rightarrow A$: $K(\text{proof I'm B})$
IKE Phase 1:  
Shared Secret Keys, Main Mode  
Assumption  and  share a secret  
Protocol:  
-  \rightarrow  \cdot  \begin{array}{l} \left( \right. \\ \left. \right. \end{array}  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
\left( \right)  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  

IKE Phase 1:  
Shared Secret Keys, Aggressive Mode  
Protocol:  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  

IKE: Phase 2  
Also known as "Quick Mode": 3- message protocol  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  \cdot  
-  \rightarrow  
\begin{array}{l} \text{All messages are encrypted using SKEYID, e, and integrity protected} \\
\text{using SKEYID, a (except X, Y) } \\
\text{Parameters: } \\
\text{A pair of cookies generated during phase 1} \\
\text{B: 32-bit number unique to this phase 2 session chosen by the initiator} \\
\text{CP: Crypto Proposal, CPA: Crypto Proposal Accepted} \\
\text{DH is optional and could be used to provide PFS} \\
\text{Nonces and cookies get shuffled into SKEYID to produce the SA encryption and integrity keys} 
\end{array}