WILLIAM & MARY

CSCI 454/554 Computer and Network Security

Topic 8.2 Internet Key Management

WILLIAM & MARY

Outline

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

WILLIAM & MARY

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.

Escrow Foilage Protection

- Key escrow: communicating parties have to store their long-term keys with a third-party (authorities, etc.)
- Escrow-foilage: key stored at the third party is used maliciously
- Escrow Foilage Protection: the conversation between Alice and Bob can still be made secret against a passive eavesdropper with prior knowledge of Alice and Bob's long-term keys.
- Anything with PFS will also have escrow-foilage against a passive attacker.

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)

A Note about IKE

- IKE v2 was introduced in RFC 4306 (December 2005)
- IKE v2 does not interoperate with IKE v1
  - Both version can unambiguously run over the same UDP port
- IKE v2 combines the contents of previously separate documents
  - ISAKMP
  - IKE v1
  - DOI
  - NAT
  - …

Automatic Key Management

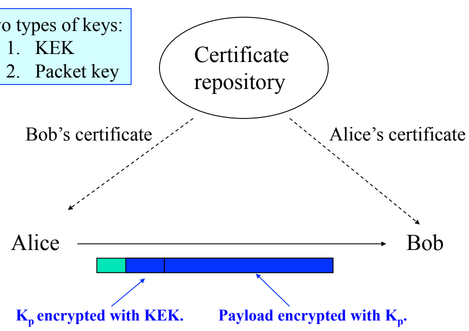
- Key establishment and management combined
  - SKIP
- Key establishment protocol
  - Oakley
    - focus on key exchange
- Key management
  - Internet Security Association & Key Management Protocol (ISAKMP)
    - Focus on SA and key management
    - Clearly separated from key exchange.

SKIP

- Simple Key-Management for Internet Prococols
- 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



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.

13

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.

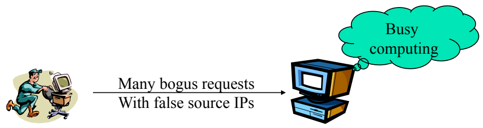
14

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

15

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

Resource Clogging Attack (Cont'd)

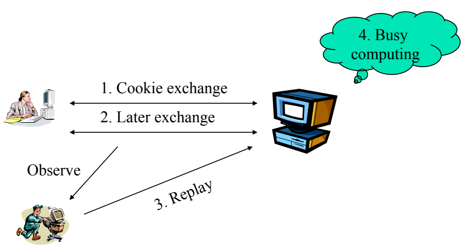
- Cookie
  - Each side sends a pseudo-random number, the cookie, in the initial message, which the other side acknowledges.
  - The acknowledgement must be repeated in the following messages.
  - Do not begin D-H calculation until getting acknowledgement for the other side.

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



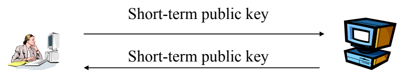
Man-In-The-Middle Attack

- 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?

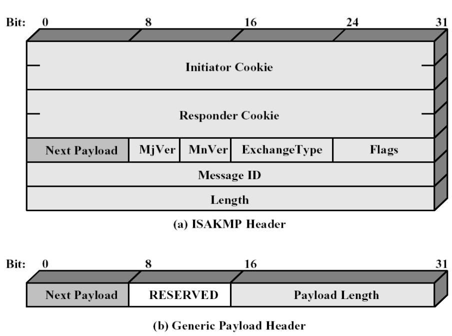
ISAKMP

- 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 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
- 240-255private 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

- 0 none
- 1 SA security association
- 2 P proposal
- 3 T transform
- 4 KE key exchange
- 5 ID identification
- 6 CERT certificate
- 7 CR certificate request

ISAKMP Payload Types

WILLIAM & MARY

- 8 H — hash
- 9 SIG — signature
- 10 NONCE — nonce
- 11 N — notification
- 12 D — delete
- 13 VID — vender ID
- 14-127 — reserved
- 128-255 — private use

ISAKMP Payload Types

WILLIAM & MARY

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

ISAKMP Exchanges

WILLIAM & MARY

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)

Identity Protection Exchange

1.	I→R: SA	Begin ISAKMP-SA negotiation
2.	R→I: SA	Basic SA agreed upon
3.	I→R: KE; NONCE	Key generated;
4.	R→I: KE; NONCE	key generated;
5.	I→R: ID_I; AUTH	Initiator id verified by responder
6.	R→I: ID_R; AUTH	Responder id verified by initiator; SA established

Blue messages: Payload encrypted after ISAKMP header

ISAKMP Exchanges (Cont'd)

Authentication Only Exchange

1.	I→R: SA; NONCE	Begin ISAKMP-SA negotiation
2.	R→I: SA; NONCE; ID_R; AUTH	Basic SA agreed upon; Responder id verified by initiator
3.	I→R: ID_I; AUTH	Initiator id verified by responder; SA established

ISAKMP Exchanges (Cont'd)

Aggressive Exchange

1.	I→R: SA; KE; NONCE; ID_I	Begin ISAKMP-SA negotiation and key exchange
2.	R→I: SA; KE; NONCE; ID_R; AUTH	Responder identity verified by responder; Key generated; Basic SA agreed upon;
3.	I→R: AUTH	Initiator id verified by responder; SA established

Red messages: Payload encrypted after ISAKMP header

ISAKMP Exchanges (Cont'd)

Informational Exchange

1.	I→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

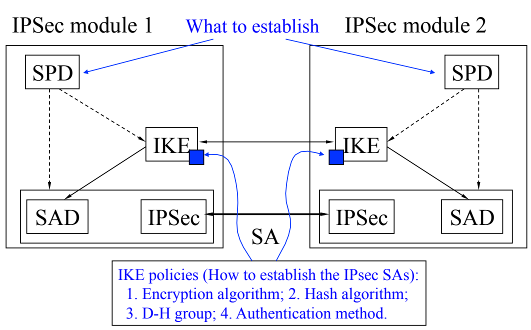
IKE Overview (Cont'd)

WILLIAM & MARY

- 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

WILLIAM & MARY



IKE Phase 1

WILLIAM & MARY

- 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
    - SKEYID = prf(Ni_b | Nr_b, g^{xy})
  - With public key encryption
    - SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)
  - With pre-shared key
    - SKEYID = prf(pre-shared-key, Ni_b | Nr_b)
  - Notations:
    - prf: keyed pseudo random function prf(key, message)
    - CKY-I/CKY-R: I's (or R's) cookie
    - Ni_b/Nr_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
    - SKEYID_d = prf(SKEYID, g^{xy} | CKY-I | CKY-R | 0)
  - Authentication key
    - SKEYID_a = prf(SKEYID, SKEYID_d | g^{xy} | CKY-I | CKY-R | 1)
  - Encryption key
    - SKEYID_e = prf(SKEYID, SKEYID_a | g^{xy} | CKY-I | CKY-R | 2)

IKE Phase 1 (Cont'd)

- To authenticate the established key
  - Initiator generates
    - HASH_I = prf(SKEYID, g^{xi} | g^{xr} | CKY-I | CKY-R | SAi_b | IDii_b)
  - Responder generates
    - HASH_R = prf(SKEYID, g^{xr} | g^{xi} | CKY-R | CKY-I | SAi_b | IDir_b)
  - Authentication with digital signatures
    - HASH_I and HASH_R are signed and verified
  - Public key encryption or pre-shared key
    - HASH_I and HASH_R directly authenticate the exchange.

IKE Phase 1 Authenticated with Signatures

Main Mode

Initiator		Responder
HDR, SA	→	
	←	HDR, SA
HDR, KE, Ni	→	
	←	HDR, KE, Nr
HDR*, IDii, [CERT,] SIG_I	→	
	←	HDR*, IDir, [CERT,] SIG_R

IKE Phase 1 Authenticated with Signatures

Aggressive Mode

Initiator		Responder
HDR, SA, KE, Ni, IDii	→	
	←	HDR, SA, KE, Nr, IDir, [CERT,] SIG_R
HDR, [CERT,] SIG_I	→	

IKE Phase 1 Authenticated with Public Key Encryption

Main Mode

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

IKE Phase 1 Authenticated with Public Key Encryption

Aggressive Mode

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	→	

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

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

IKE Phase 1 Authenticated with A Revised Mode of Public Key Encryption

Aggressive Mode

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 →

Further Details

Ne_i=prf(Ni_b, CKY-I)
Ne_r=prf(Nr_b, CKY-R)

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

IKE Phase 1 Authenticated with Pre-Shared Key

Main Mode

Initiator → Responder

HDR, SA →

← HDR, SA

HDR, KE, Ni →

← HDR, KE, Nr

HDR*, IDii, HASH_I →

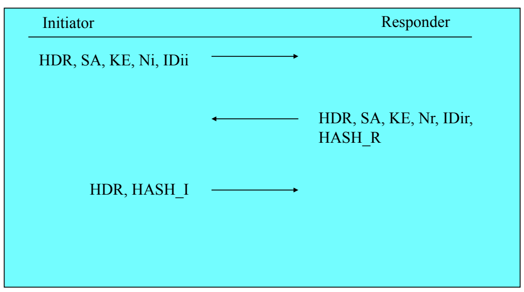
← HDR*, IDir, HASH_R

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



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 nonce
  - 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)

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)

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

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

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

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

where $g(qm)^{xy}$ 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.