Internet Key Exchange (IKE)

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

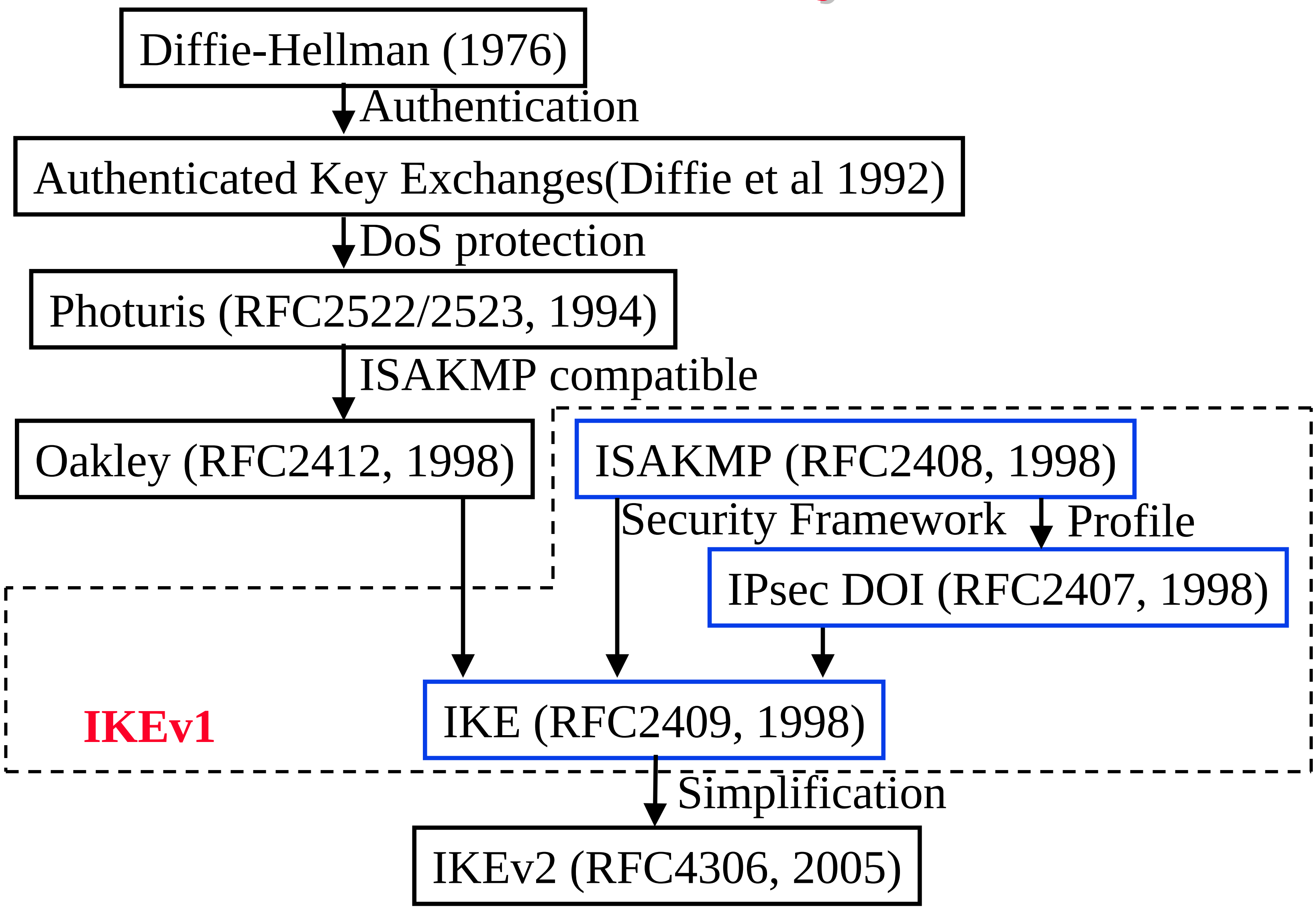
http://www.cse.wustl.edu/~jain/cse571-07/

- IKE Phases
- Main Mode and Aggressive Mode
- Authentication Methods
- Session Keys
- ISAKMP/IKE Encoding and Payload types
- IKE Version 2

Internet Key Exchange (IKE)

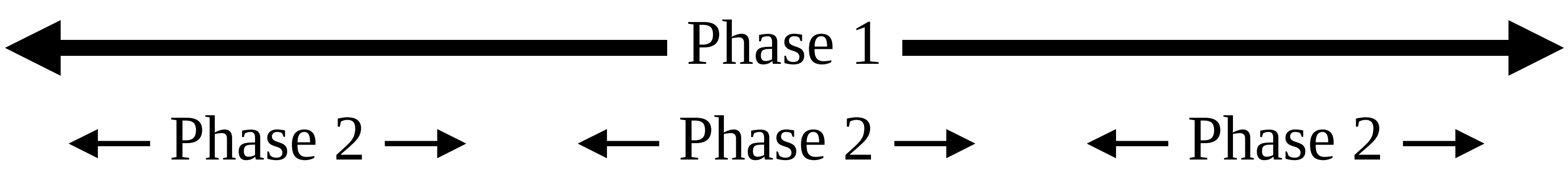
- Mutual authentication and establish a shared secret
- Features: Hiding end point identifiers, crypto algorithm negotiation
- Many modes and phases

IKE History


Diffie-Hellman (1976)
Authentication
Authenticated Key Exchanges(Diffie et al 1992)
DoS protection
Photuris (RFC2522/2523, 1994)
ISAKMP compatible
Oakley (RFC2412, 1998)
ISAKMP (RFC2408, 1998)
Security Framework
Profile
IPsec DOI (RFC2407, 1998)
IKEv1
IKE (RFC2409, 1998)
Simplification
IKEv2 (RFC4306, 2005)

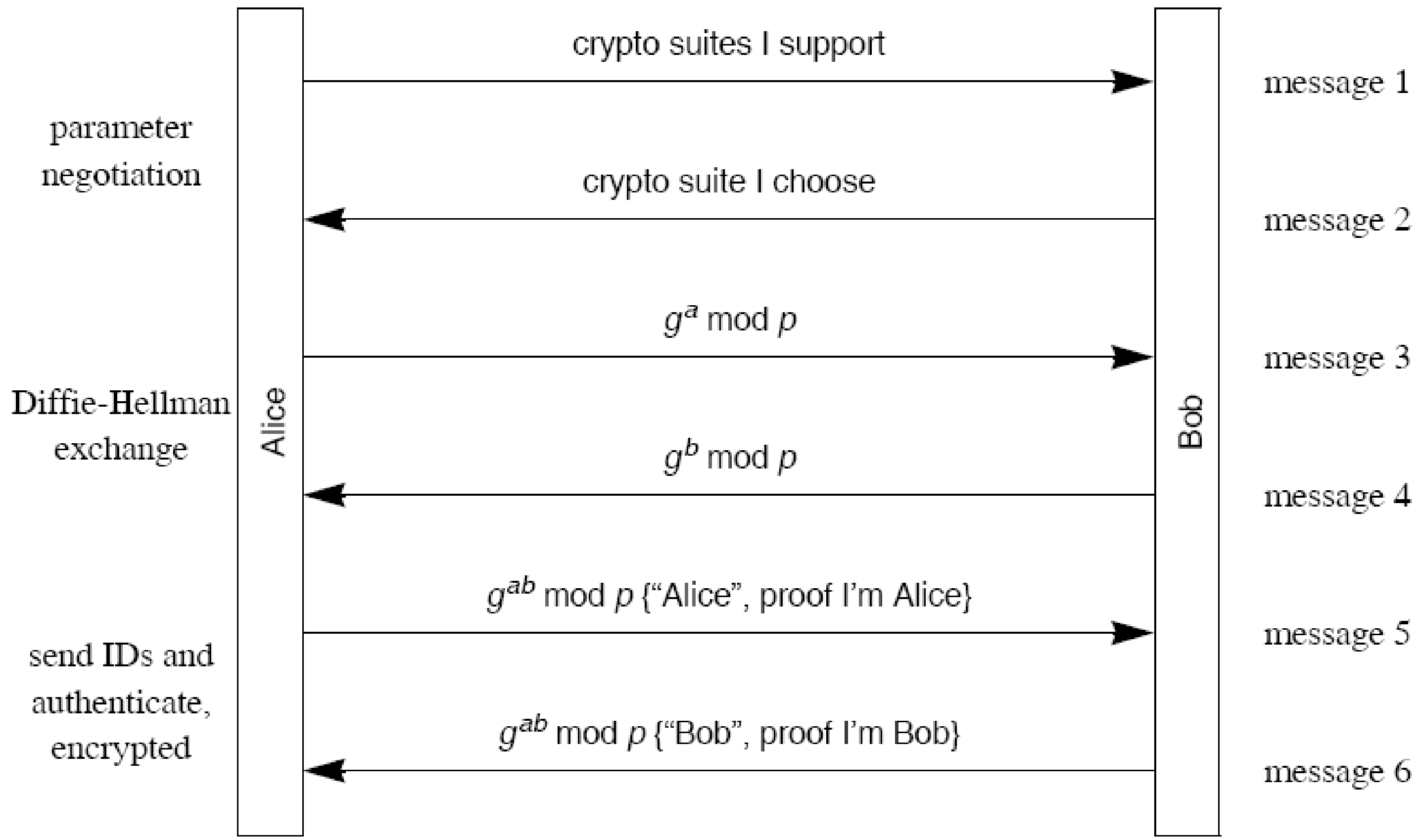
IKE Phases

- May need to setup multiple connections with different security properties ⇒ Two phases
- Phase 1: Mutual authentication and session keys = IKE SA
- Phase 2: Use results of phase 1 to create multiple associations between the same entities = ESP or AH SA
- IKE SA is bi-directional
- AH and ESP SAs are unidirectional



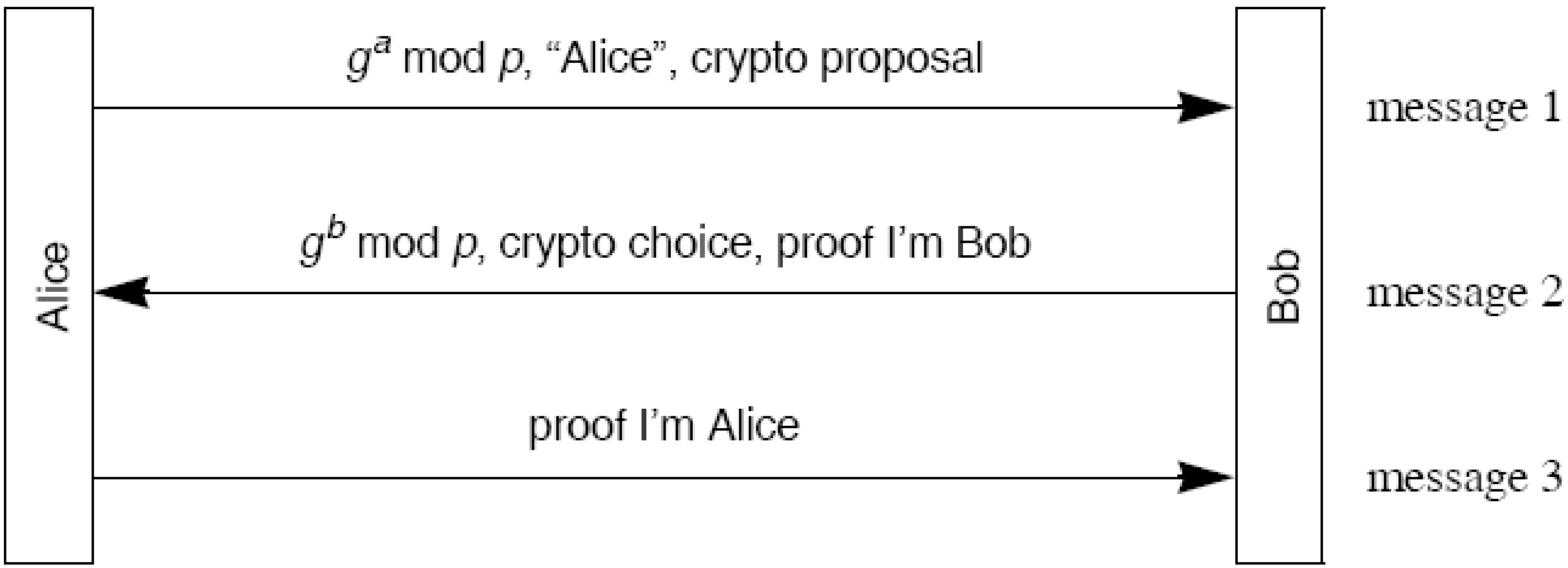
IKE Main Mode

- Allows ability to hide end-point identifiers and to select crypto algorithms ⇒ requires 6 messages



IKE Aggressive Mode

- End-points ID not hidden ⇒ Requires only three messages



IKE Authentication Methods

1. Original Public Key Encryption (separately encrypt each field with other sides public key)
2. Revised Public Key Encryption (Encrypt session key with public key. Use session key to encrypt the rest)
3. Public key signature
4. Pre-shared secret key

4 Methods $\times$ 2 Modes = 8 variants of Phase 1

Authentication Methods: Comparison

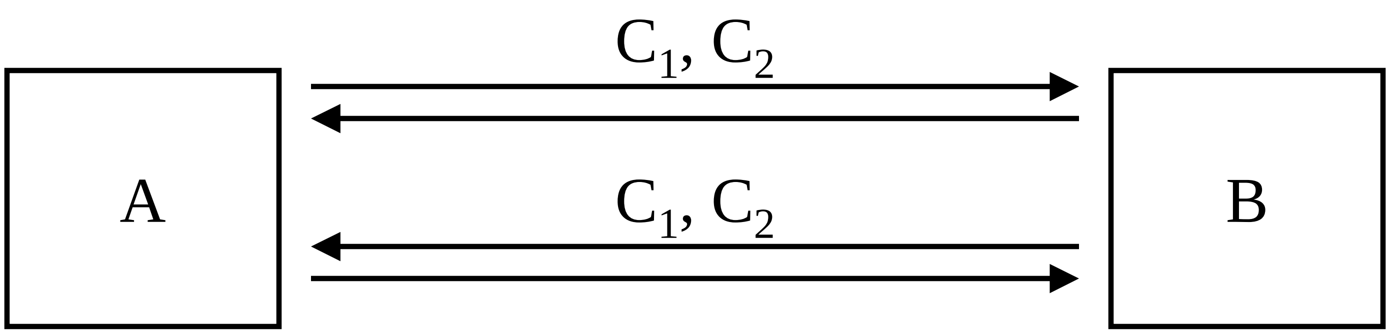
- Public vs. Pre-shared: Public requires sending the certificate first
- Public key: Need to reveal the identity
- Encryption vs. Signature keys: Encryption keys may be escrowed. Signature keys are not.
- With signature key, identity may be revealed to an imposter.
- With encryption keys, identity is revealed only to intended entity.

Proof of Identity

- Different for each authentication method
- Hash(key, DH value, nonces, crypto choices)
- Could have been the same for all authentication methods
- Integrity check does not cover selected crypto algorithm

IKE Phase 1 Cookies

- Proof of identity in the last message includes hashes of all previous messages
- Need to remember the crypto choices offered ⇒ State
- ISAKMP requires cookies to be unique for each connection from the same IP address ⇒ Cannot use stateless cookies
- Connection identifier = <Initiator cookie, responder cookie> ⇒ May end up with the same connection identifier for two connections



DH Parameters

- Modular exponentiation or Elliptic curves
- g^a mod p ⇒ Need to select a large prime p and generator g
- The group identifiers:
  - 0 = No group
  - 1 = A modular exp with a 768 bit modulus
  - 2 = A modular exp with a 1024 bit modulus
  - 3 = A modular exp with a 1536 bit modulus
  - 4 = An elliptic curve group over GF[2^{155}]
  - 5 = An elliptic curve group over GF[2^{185}]

Well-Known Group 1

- A 768 bit prime based on digits of π
- $2^{768} - 2^{704} - 1 + 2^{64} \times \{ [2^{638} \pi] + 149686\}$
- Decimal value:

1552518092300708935130918131258481755631334049434514313202351194902966239949102107258669453876591642442910007680288864229150803718918046342632727613031282983744380820890196288509170691316593175367469551763119843371637221007210577919

Representation in OAKLEY

- Type of group: "MODP"
- Size of field element (bits): 768
- Prime modulus:
  - Length (32 bit words): 24

Well-Known Group 1 (Cont)

- Data (hex):

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF

- Generator: 22 (decimal)
  - Length (32 bit words): 1
- See RFC2412 for other well-known groups.

Negotiating Cryptographic Parameters

- Allows negotiating encryption (DES, 3DES, IDEA), hash (MD5, SHA), authentication method (Pre-shared keys, DSS), DH parameters
- Must implement: DES, MD5 and SHA, pre-shared key, modp
- Need to send allowed combinations
  $\Rightarrow$ Large number of choices
- In the aggressive mode, initiator selects a combination. Responder can only reject
- Can also specify a lifetime in terms of time or number of bytes

IKE Session Keys

- Phase 1 $\Rightarrow$ Integrity key and Encryption Key
- The two keys are used in the last phase 1 message and all phase 2 messages
- Note the same keys are used in both directions
  $\Rightarrow$ Reflection attack can cause DoS
- SKEYID = hash (DH values, nonces, cookies, pre-shared secret if any) $\Rightarrow$ Key seeds
- prf = pseudo random function (e.g., DES CBC, or HMAC) with two parameters - key and data

IKE Session Keys (Cont)

- Public Signature Authentication:
  SKEYID = prf(nonces, g^{xy} mod p)
- Public Encryption Authentication:
  SKEYID = prf(hash(nonces), cookies)
- Pre-share secret key authentication:
  SKEYID = prf(pre-shared secret key, nonces)

- SKEYID_d = prf(SKEYID, (g^{xy} mod p| cookies | 0))
- SKEYID_a = prf(SKEYID, (SKEYID_d | (g^{xy} mod p | cookies | 1)) = Integrity (Authentication) Protection Key
- SKEYID_e = prf(SKEYID, (SKEYID_a | (g^{xy} mod p | cookies | 2)) = Encryption Key

IKE Session Keys (Cont)

- Proof of identity for initiator = prf(SKEYID, (g^x mod p| g^y mod p|cookies|Initial crypto parameter proposal|Initiator's Identity))
- Proof of identity for responder = prf(SKEYID, (g^x mod p| g^y mod p|cookies|Initial crypto parameter proposal|Responder's Identity))

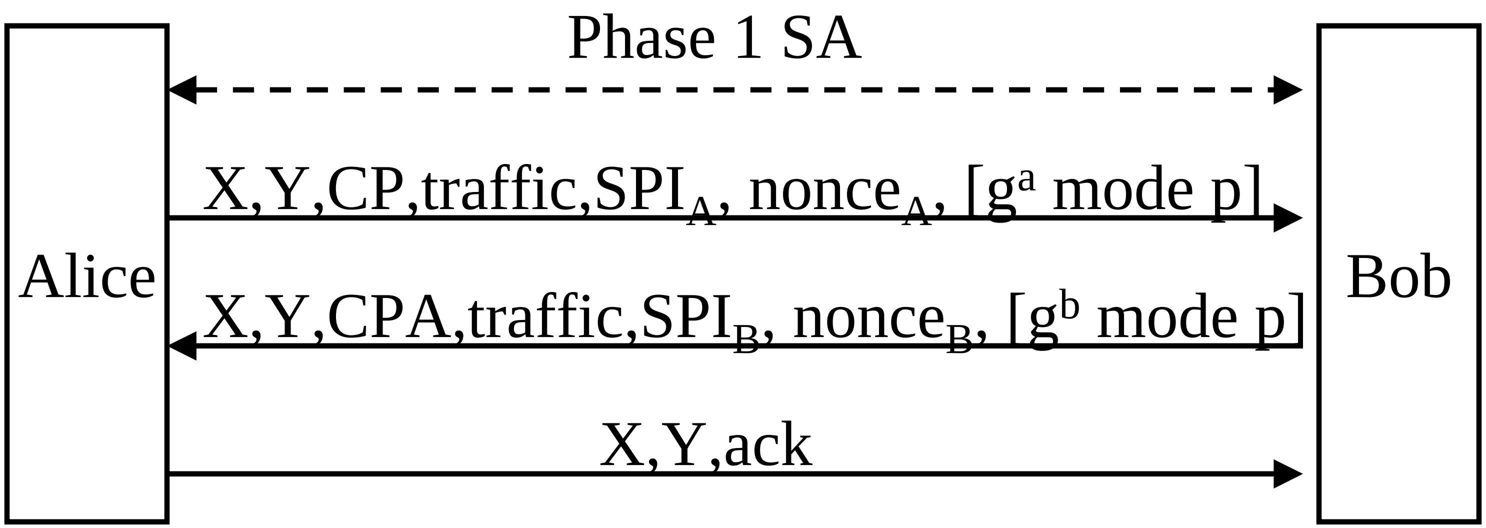
IKE Message IDs

- IKE messages contain a 32-bit message ID to avoid replay
- ISAKMP requires these IDs to be randomly chosen $\Rightarrow$ Difficult to check for replay
- Sequence numbers would have been better

IKE Phase 2

- IPsec (AH or ESP) SA can be set up in Phase 2
  $\Rightarrow$ Negotiate crypto parameters, another optional DH (for perfect forward secrecy), traffic selectors
- Traffic selector = IP address or mask, IP protocol type, and TCP/UDP port #
- If traffic selector is wider than the acceptable, the request will be refused
- Phase 2 is also known as quick mode

IKE Phase 2 (Cont)



- X = pair of cookies generated in phase 1
- Y = a 32-bit number to distinguish different phase 2 sessions
- CP = Crypto Proposal, CPA = Crypto Proposal Accepted
- X and Y are in clear rest of the phase 2 messages are encrypted with SKEYID_e and integrity protected with SKEYID_a
- IV = final cipher text block of the previous message

ISAKMP/IKE Encoding

- All messages start with a 28-octet fixed header

# octets		
8	initiator's cookie	
8	responder's cookie	
1	next payload	
1	version number (major/minor)	
1	exchange type	
1	flags	
4	message ID	
4	message length (in units of octets)	(after encryption)

ISAKMP/IKE Encoding (Cont)

- Exchange type: 2=Main mode, 4=aggressive mode, ...
- Followed by a sequence of payloads

Type of Next Payload		Type of Next Payload = 0
Length of This Payload	…	Length of This Payload
Payload		Payload

- Payload type:1=SA, 2= Proposal, ...
- Each SA payload consists of multiple proposals (P) payloads.
- Each P payload consists of multiple transform (T) payloads
- In phase 1, only one P inside SA.
- In phase 2, there can be multiple proposals (protocols), e.g., AH, ESP, AH+ESP, AH+ESP+Compression
- T payload consists of a complete choice suite, E.g., Authentication, hash, encryption, DH combination in phase 1

ISAKMP Payload Types

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.

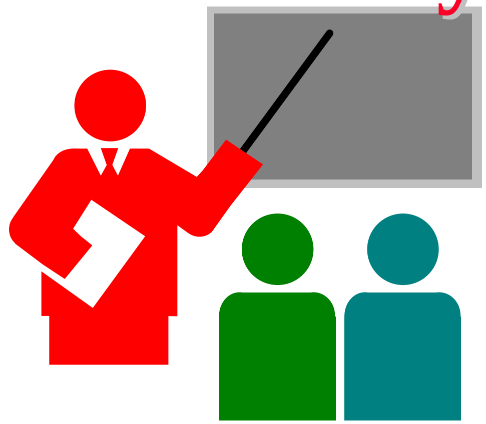
ISAKMP Payload Types (Cont)

Type	Parameters	Description
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.

IKE Version 2

- RFC 4306, December 2005 (V1 in RFC2407, RFC2408, RFC2409, November 1998)
- Replaces 8 negotiations methods by single method
- Easier to Implement $\Rightarrow$ Less Interoperability problems $\Rightarrow$ More deployment
- Less vulnerable to DoS

Summary



- ISAKMP is a framework for key exchange and IKE is a profile of ISAKMP.
- IKE consists of 3 documents: ISAKMP, DOI, IKE
- IKE consist of 2 Phases.
  Phase 1 generates SKEYID, SKEYID_a, SKEYID_d
  Phase 2 generates session keys. Multiple phase 2 per phase 1
- Two modes: Aggressive and Main
- Four authentication methods: Shared Secret, Public Encryption Key (Old, revised), Public Signature Key ⇒ Total 8 variants

Homework 14

- Read chapter 18 of the text book
- Submit answer to the Exercise 18.3
- Exercise 18.3: Show how someone who knows both Alice's and Bob's Public encryption keys (and neither side's private key) can construct an entire IKE exchange based on public encryption keys that appears to be between Alice and Bob. (Hint: Show that a third party can create a valid exchange shown in Figure 18-6, 18-7, 18-8 without knowing the private keys. Note that the proof of identity consists of a hash of the nonce sent by the other side.)