IP Security
- PartIII

Professor Yeali S. Sun
Information Management Department
National Taiwan University

Key Management

Key Management

- Goal
  - To determine and distribute secret keys for such as IPsec Encryption and Authentication.
  - Need mechanisms for communicating peers to agree on algorithms, key sizes, and other minutiae (small details).
- Typical scenario
  - Four keys: Transmit and receive pairs of keys for both AH and ESP between two communicating applications.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Two types of key management

- Manual
  - System administrator manually configures each system with its own keys and the keys of the other party
  - For small, static environments
- Automated
  - An automated system provides on-demand creation of keys for SAs
  - For large, dynamic environments

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

The Internet Key Exchange (IKE)

RFC2409

D. Harkins and D. Carrel November 1998

Standards Track

Preface

- ISAKMP – framework
- Oakley – key exchange protocol
- SKEME – key exchange protocol

ISAKMP

- Internet Security Association and Key Management Protocol
- A framework for peer <u>authentication</u> and <u>key exchange</u>
- Define a set of message types to
  - enable the use of a variety of key exchange algorithm; and
  - allow negotiation of security attributes

  between communicating parties.
- The default automated key management protocol.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley

- A key exchange protocol
- Enabling two users to exchange a key ***securely.***
- Based on Diffie-Hellman algorithm with added security.
- Mandated for use with the initial version of ISAKMP

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

SKEME (**S**ecure **K**ey **E**xchange **ME**chanism protocol)

- A versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Internet Key Exchange (IKE)

- IKE is a protocol using *part of* Oakley and *part of* SKEME in conjunction with ISAKMP.
- The goal is to **obtain authenticated keying material** for use with ISAKMP, and for other security associations such as AH and ESP for the IETF IPsec DOI.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley

Diffie-Hellman Key Determination Protocol (1/6)

- It allows two parties to agree on a *shared* value *without* requiring encryption.
- Users A and B
- In priori agreement on two global parameters: q and α
  - q: a large prime number
  - α: a primitive root of q
- Primitive root
  - If α is the primitive root of a very large prime number *q*, then the following numbers are distinct

    $$\alpha \bmod q, \alpha^2 \bmod q, \ldots, \alpha^{q-1} \bmod q$$

  - For any integer *b*, one can find a *unique* exponent *i* such that

    $$b = \alpha^i \bmod q \quad \text{where } 0 <= i <= (q-1)$$

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Diffie-Hellman Key Determination Protocol (2/6)

- Procedure
  - User A selects a random integer X_A as its private key and sends User B its public key Y_A $(= \alpha^{XA})$
  - User B selects a random integer X_B as its private key and sends User A its public key Y_B $(= \alpha^{XB})$
  - Compute the secret session key.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Diffie-Hellman Key Determination Protocol (3/6)

Global Public Element

q - prime number

α - $\alpha < q$ and α is a primitive root of q

User A Key Generation

Select private X_A $X_A < q$

Calculate public Y_A $Y_A = \alpha^{XA}$

User B Key Generation

Select private X_B $X_B < q$

Calculate public Y_B $Y_B = \alpha^{XB}$

Exchange Public Keys Y_A and Y_B

Generation of secret key by A

$K = (Y_B)^{XA} \bmod q$

Generation of secret key by B

$K = (Y_A)^{XB} \bmod q$

. Sun. All rights reserved. No part
stored in a retrieval system, or transmitted
or by any means without the prior written permission of the author.

Diffie-Hellman Key Determination Protocol: Algorithm (4/6)

- $K = (Y_B)^{X_A} \bmod q$
  $= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$
  $= (\alpha^{X_B})^{X_A} \bmod q$
  $= \alpha^{X_B X_A} \bmod q$
  $= (\alpha^{X_A})^{X_B} \bmod q$
  $= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$
  $= (Y_A)^{X_B} \bmod q$

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

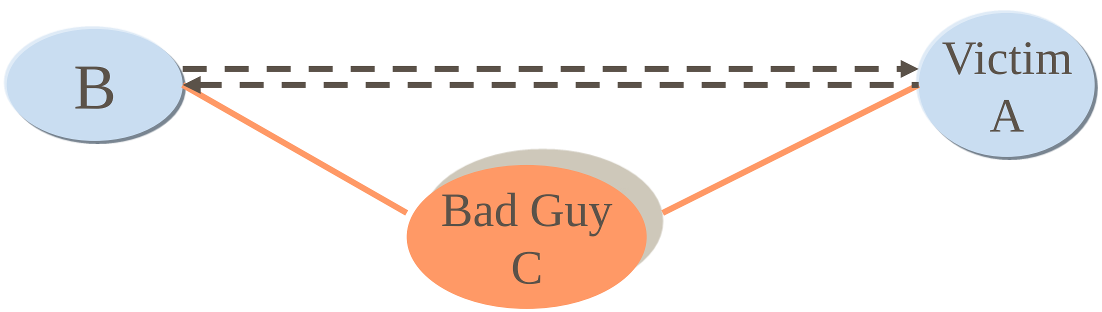
Diffie-Hellman Key Determination Protocol: Advantages (5/6)

- Secret keys are created only when needed.
- No need to store secret keys for a long period of time.
- No need to transfer secret keys over the Internet
- Authentication (secret keys are generated using the secret key of the public key's true owner)

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Diffie-Hellman Key Determination Protocol: Weakness (6/6)

- It does **NOT** provide information about the ***identities*** of the parties.
- Subject to a man-in-the-middle attack



Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: features (1/9)

- Add *authentication* to the Diffie-Hellman exchange
  - to prevent man-in-the-middle attacks.
- Cookies
  - anti-clogging tokens to prevent denial of service (clogging) attack
  - to protect computing resources from attack without spending excessive CPU resources to determine its authenticity.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: features (2/9)

- Support the use of different *groups* for the Diffie-Hellman key exchange.
  - Allow two parties to negotiate global parameters of the Diffie-Hellman key exchange (i.e., q and α) and the identity of the algorithm.
  - Three distinct group representations are defined
    - modular exponentiation groups (α=2, 768-bit modulus, 1024-bit modulus, etc.)
    - elliptic curve groups over 2^{155}
    - elliptic curve groups over 2^{185}

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

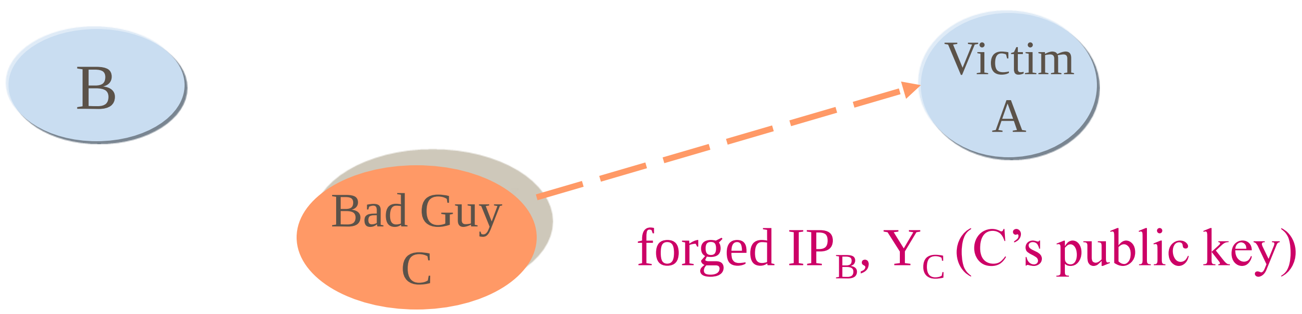
Oakley: features (3/9)

    - For each representation, many distinct realizations are possible, depending on parameter selection.
- Nonces
    - To ensure against reply attacks

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: Clogging (Denial of Service) Attack (4/9)

- Bad guy C **forges** the source address of a legitimate User B and sends a public key to the victim A.
- The victim A computes secret key. (no authentication)
- Repeated messages of this type can **clog** the victim's system.



Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: cookies for anti-clogging tokens (5/9)

The concept of cookie

- Cookies provide a weak form of source address identification for the two communicating parties.
- They do cookie exchange before to perform the computationally expensive part of the protocol (large integer exponentiations).

- Each party sends a *pseudorandom number* – cookie – in the initial message which the other side acknowledges.
- This acknowledgment is repeated in the *first* message of the Diffie-Hellman key exchange.
- If the source address is forged, the opponent gets no answer.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: Cookies Requirements (6/9)

- The cookie *must* depend on the specific parties (a weak form of authentication).
- The issuing entity will use local secret info in the generation and subsequent verification of a cookie.
- Fast cookie generation and verification to prevent attacks sabotaging processor resources.
  - e.g., use a fast hash (e.g., MD5) over the IP src/dst addr., the UDP src/dst ports and a locally generated secret value.
- Cookies are 64-bit pseudo-random numbers.
  - The generation method must ensure with high probability that the numbers used for each IP remote address are unique over some time period, such as one hour.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: Cookie Requirements (7/9)

- Protection
  - To <u>prevent</u> attacks that obtain a cookie using an real IP address and UDP port to swamp the victim with requests from randomly chosen IP addresses or ports.
  - It is impossible for anyone other than the issuing party to generate cookies that will be accepted by the entity.
- Note that absolute protection against denial of service is *impossible*, but this anti-clogging token provides a technique for making it easier to handle.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

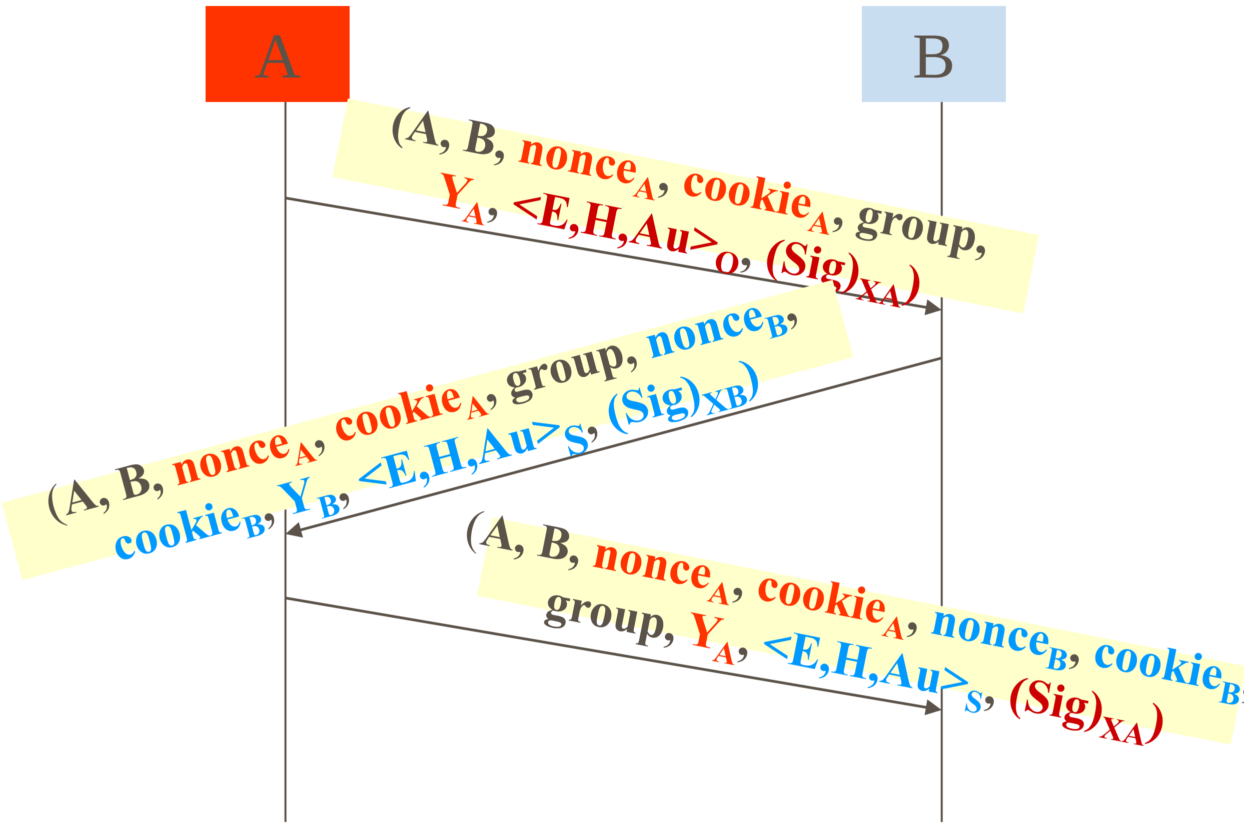
Oakley: Authentication Methods (8/9)

Three are specified

- Digital signatures
  - Generate *a mutual hash* over some parameters, e.g., user IDs and nonces.
  - Each party *encrypts the hash with its private key*.
  - The receiving party authenticates the sender using *sender's public key decryption*.
- Public-key encryption
  - A sending party encrypts information such as IDs and nonces with its private key.
  - The receiving party authenticates the sender using its public key decryption.
- Symmetric-key encryption
  - A key is derived from some out-of-band mechanism to authenticate both by symmetric encryption of exchange parameters.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Oakley: Aggressive Key Exchange (9/9)



- group: name of D-H grp for this exchange
- E: encryption
- H: Hashing
- Au: authentication
- Sig: IDs, nonces, group, Y's, EHAS

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Internet Security Association and Key Management Protocol (ISAKMP)

RFC 2408

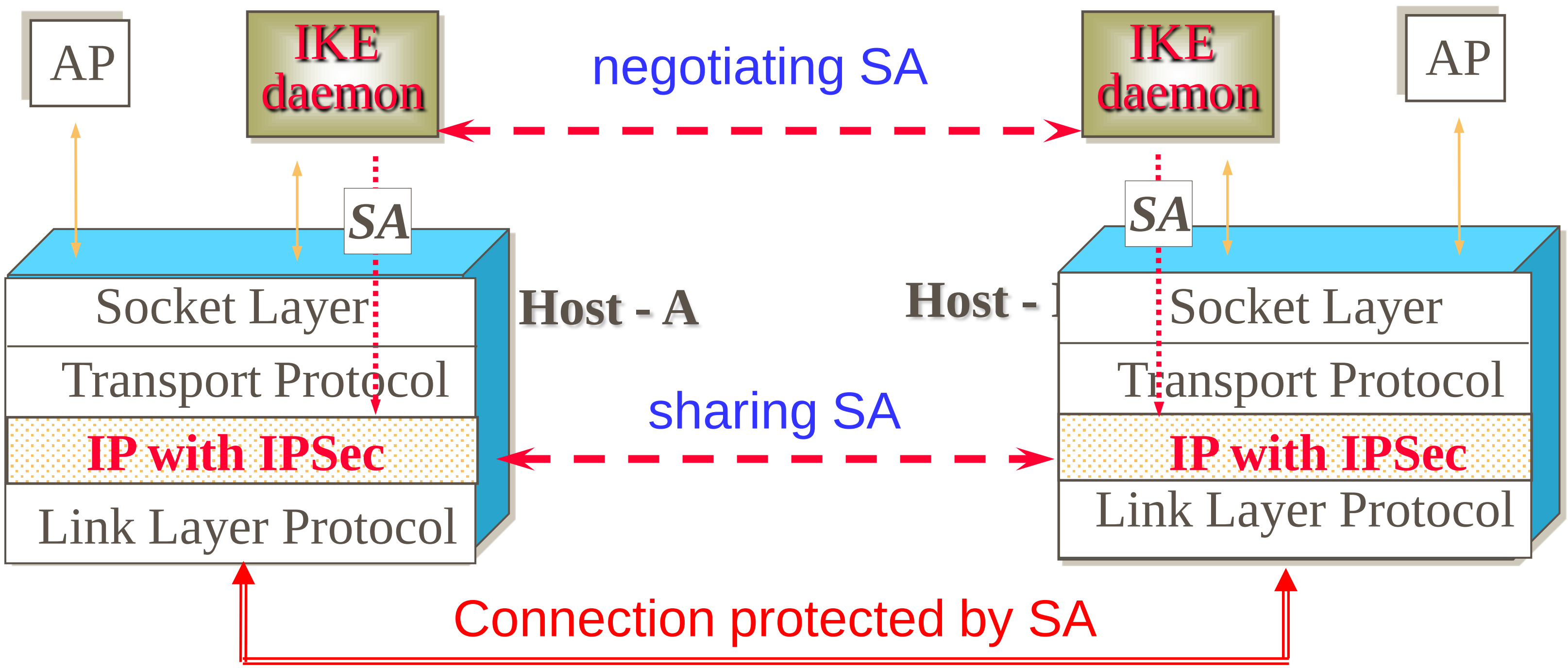
November 1998

Standards Track

ISAKMP: Goals

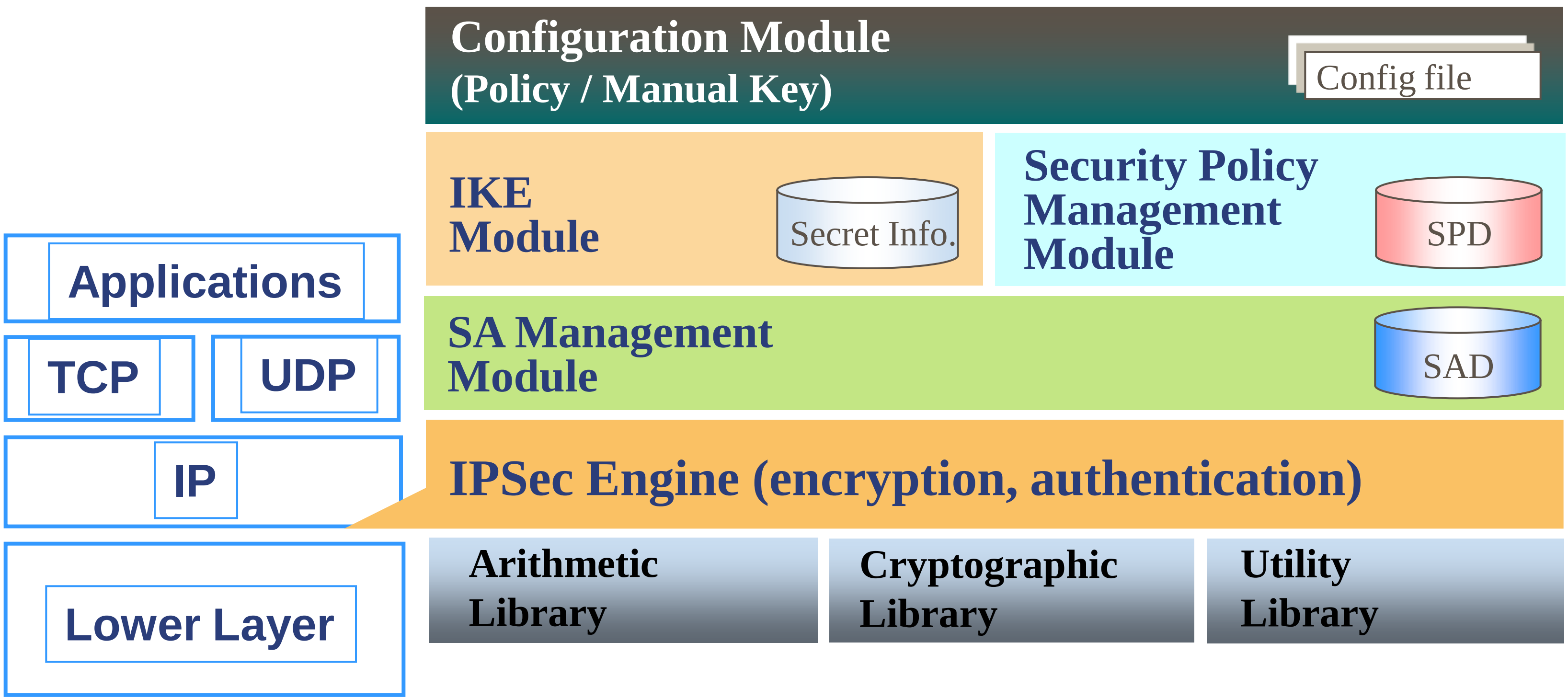
- Defines procedures and packet formats to establish, negotiate, modify and delete Security Associations (SA).
- At the establishment phase, payloads are defined to exchange key generation and authentication data.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.



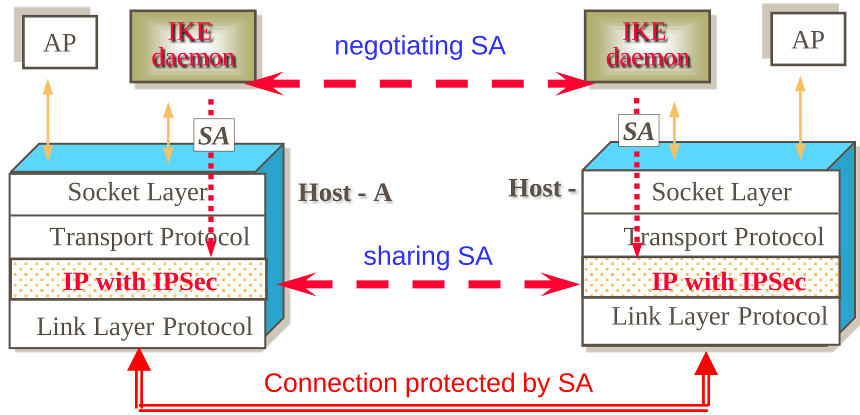
Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

IPsec Architecture

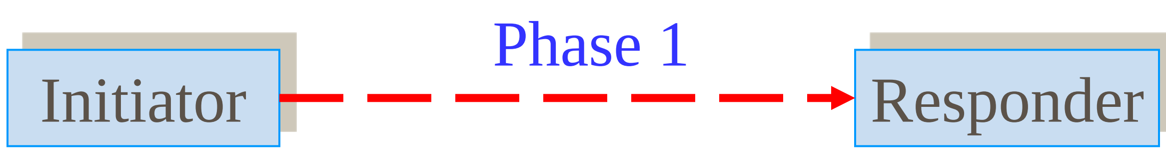


Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Two Phases of negotiation

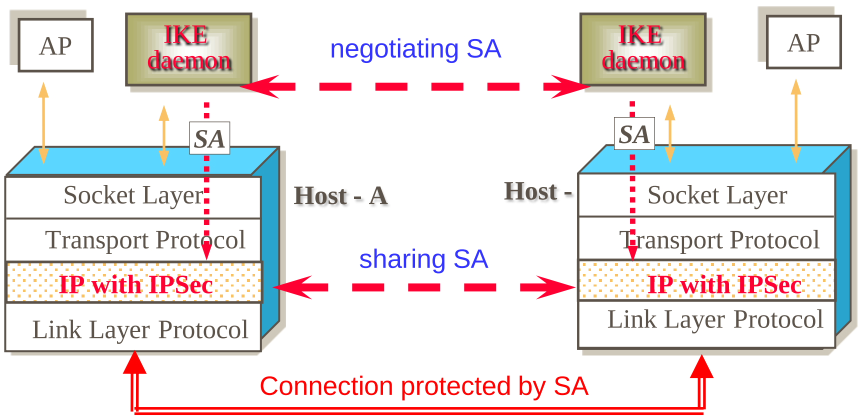


- The *establishment* of a secure communication channel between two parties consists of two phases:
  - Phase 1. Establish an ISAKMP SA.
  - Phase 2. Establish actual IPsec SA.
- Initiator and Responder



Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Two Phases of negotiation



Phase I: Goals

- *Negotiate* ISAKMP SA *parameters*
- Establish a *shared secret* for Phase II.
- *Authenticate identities of* servers/hosts.

Phase II: Goals

- Establish IPsec SA
- Authenticate identities of users or application processes.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Header (1/6)

- An ISAKMP message consists of an ISAKMP header followed by one or more payloads.
- Initiator cookie (8 octets)
- Responder cookie (8 octets)
- Two cookie fields are used to identify an SA

Initiator Cookie				
Responder Cookie				
Next Payload	Major Version	Minor Version	Exchange Type	Flags
Message ID				
Length				

ISAKMP: Header

(2/6)

Initiator Cookie				
Responder Cookie				
Next Payload	Major Version	Minor Version	Exchange Type	Flags
Message ID				
Length				

- Next payload (1 octet)
  - Indicates the type of the first payload in the message (value 0 means the last).
- Major Version (4 bits)
  - Indicates the major version of the ISAKMP protocol in use.
- Minor version (4 bits)
  - Indicates the minor version
- Exchange type (1 octet)
  - Indicates the message and payload ordering in the ISAKMP exchanges.
- Flags (1 octet)
  - Indicates specific options set for the ISAKMP exchange.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Next Payload Types (3/6)

- None (0)
- Security Association (SA) (1)
- Proposal (P) (2)
- Transform (T) (3)
- Key Exchange (KE) (4)
- Identification (ID) (5)
- Certificate (CERT) (6)
- Certificate Request (CR) (7)
- Hash (HASH) (8)
- Signature (SIG) (9)
- Nonce (NONCE) (10)
- Notification (N) (11)
- Delete (D) (12)
- Vendor ID (VID) (13)

- RESERVED (14-127)
- Private USE (128-255)

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Exchange Types (4/6)

- None (0)
- Base (1)
- Identity Protection (2)
- Authentication Only (3)
- Aggressive (4)
- Informational (5)
- ISAKMP Future Use (6-31)
- DOI Specific Use (32-239)
- Private Use (240-255)

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP: Header

(5/6)

Initiator Cookie				
Responder Cookie				
Next Payload	Major Version	Minor Version	Exchange Type	Flags
Message ID				
Length				

- Flags (8 bits)
  - Encryption bit (0 - the least significant bit)
    - 1: all payloads following the header are encrypted
    - 0: payload is not encrypted.
  - Commit bit (1)
    - signal key exchange synchronization.
    - to ensure the encrypted material is not received prior to completion of the SA establishment.
    - can be set at anytime by either party.
    - The value must be reset after the Phase 1 negotiation.
  - Authentication Only bit (2)
    - allow the transmission of information with integrity check but no encryption. (e.g., “emergency mode”).
  - The remaining bits are set to 0 prior to transmission.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP Header: (6/6)

Initiator Cookie				
Responder Cookie				
Next Payload	Major Version	Minor Version	Exchange Type	Flags
Message ID				
Length				

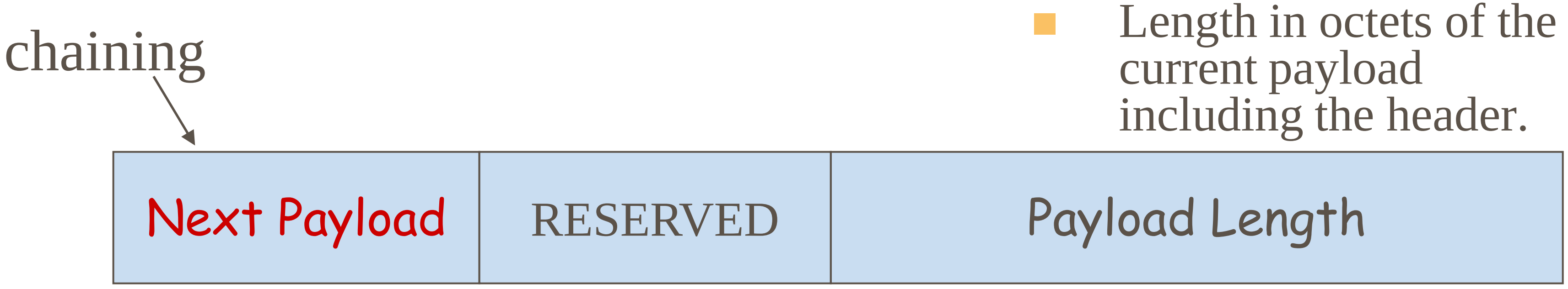
- Message ID (4 octets)
  - Unique message identifier
  - During phase I, the value is set to 0.
  - The value is randomly generated by the initiator for phase II negotiation.
- Length (4 octets)
  - The length of total message (header + payloads) in octets.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP payload: generic payload header (1/2)

- Each ISAKMP payload has a generic payload header plus a number of data attributes.

- Next Payload (1 octet)
  - Identifier for the payload type of the next payload in the message.
  - If the last, the field is set to 0.
- RSERVED (1 octet)
  - Unused; set to 0.
- Payload length (2 octets)
  - Length in octets of the current payload including the header.



Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP payload: data attribute fields (2/2)

A F	Attribute Type	AF=0 Attribute Length AF=1 Attribute Value
AF=0 Attribute Value AF=1 Not Transmitted		

- Data attribute fields contain information about the attributes for each domain in a DOI document, e.g., IPSEC DOI (IPDOI).
- Attribute Format (AF) (1 bit)
  - 0: Type/Length/Value
  - 1: Type/Value

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

ISAKMP Exchange Types

- Basic exchange
- Identity Protection Exchange
- Authentication Only Exchange
- Aggressive Exchange
- Informational Exchange

Type #1 - Base Exchange (1/4)

- Goal
  - Allow the Key Exchange and Authentication related info to be transmitted in one message.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #1 - Base Exchange (2/4)

Initiator		Responder	
(1) HDR; SA; NONCE	=>		//begin ISAKMP capability exchange
(2)	<=	(2) HDR; SA; NONCE	//basic SA agreed
(3) HDR; KE; IDii; AUTH	=>		// key generated by responder //initiator identity verified by responder

(4)	<=	HDR; KE; IDir; AUTH	// responder identity verified by initiator
			//key generated by initiator
			//SA established

Type #1 - Base Exchange (3/4)

Step (1)

- The SA, Proposal, and Transform payloads are included in the SA payload.
- NONCE
  - a random info used to guarantee liveness and protect against replay attacks.
  - NONCEs provided by both parties are used by the authentication mechanism as a shared proof of participation in the exchange.

Step (2)

- Local security policy dictates the action of the responder if no proposed protection suite is accepted,
  - e.g., the transmission of a Notify payload as part of an Informational Exchange.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #1 - Base Exchange (4/4)

- Notes
  - This can reduce the number of round trips at the expense of **not** providing identity protection.
  - Identities are exchanged before a common shared secret has been established and, therefore, encryption of the identities is not possible.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #2 - Identity Protection Exchange (1/2)

- Goal
  - Separate the Key Exchange info from the Identity and Authentication related info
  - Protect identity exchange under the protection of a previously established common shared secret.
- At the expense of two additional messages.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #2 - Identity Protection Exchange (2/2)

Initiator		Responder	
(1) HDR; SA	=>		//begin ISAKMP capability exchange
(2)	<=	(2) HDR; SA	//basic SA agreed
(3) HDR; KE; NONCE	=>		// key generated by initiator and responder
(4)	<=	HDR; KE; NONCE	
(5) HDR*; IDii; AUTH	=>		//initiator identity verified by responder
(6)	<=	HDR*; IDir; AUTH	// responder identity verified by initiator
			//SA established

(5) &(6)

- The initiator and responder exchange identification info and the results of the agreed upon authentication function.
- This info is transmitted under the protection of the common shared secret.

Type #3 - Authentication Only Exchange (1/2)

- The goal is to allow only Authentication related info to be transmitted.
- Perform only authentication without the computational expense of computing keys.
- Therefore, none of the transmitted info will be encrypted.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #3 - Authentication Only Exchange (2/2)

Initiator		Responder	
(1) HDR; SA; NONCE	=>		//begin ISAKMP // capability exchange
(2)	<=	(2) HDR; SA; NONCE IDir; AUTH	//basic SA agreed // responder identity verified by initiator
(3) HDR; IDii; AUTH	=>		//initiator identity verified by responder //SA established

Type #4 - Aggressive Exchange (1/2)

- The goal is to allow SA, KE and AUTH related payloads to be transmitted in one message.
- To reduce the number of round trips at the expense of **not** providing identity protection.
- The SA is created in one single exchange.

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #4 - Aggressive Exchange (2/2)

Initiator		Responder	
(1) HDR; SA; KE; NONCE; IDii	=>		//begin ISAKMP & key // exchange
(2)	<=	(2) HDR; SA; KE; NONCE; IDir; AUTH	//initiator identity verified by responder
(3) HDR*; AUTH	=>		// responder identity verified by initiator //SA established

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

Type #5 - Informational Exchange

- The goal is to allow one-way transmittal of info that can be used for security association management.

Initiator		Responder
(1) HDR*; N/D // Notify or Delete Payload	=>	// error notification or deletion

Copyright 2010 Yeali S. Sun. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means without the prior written permission of the author.

The end. ☺