CIS 6930/4930 Computer and Network Security

Topic 8.2 Internet Key Management

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

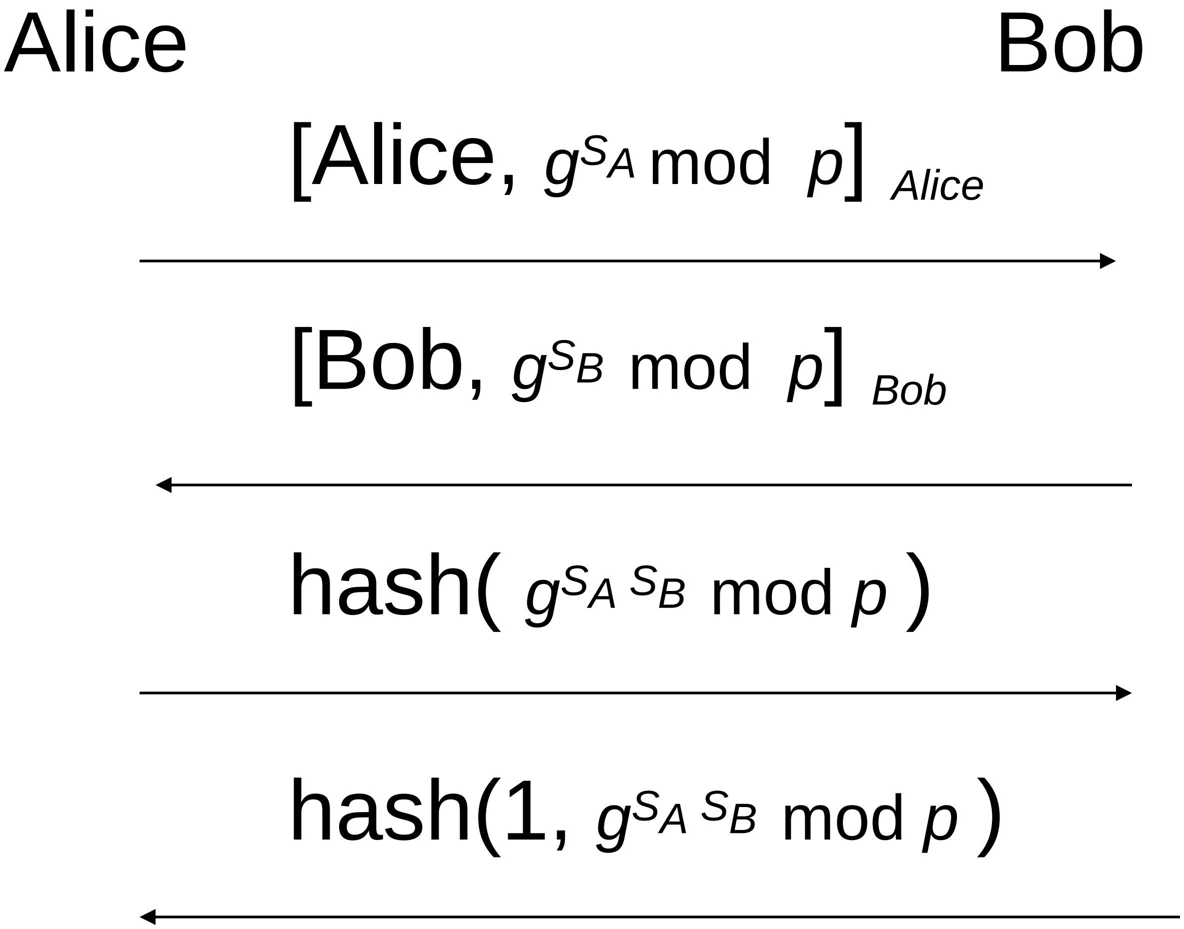
Security Principles

- 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.

Perfect Forward Secrecy Example



Alice

Bob

$[\text{Alice}, g^{s_A} \bmod p]_{Alice}$

$[\text{Bob}, g^{s_B} \bmod p]_{Bob}$

$\text{hash}(g^{s_A s_B} \bmod p)$

$\text{hash}(1, g^{s_A s_B} \bmod p)$

Examples of Non Perfect Forward Secrecy

- Alice sends all messages with Bob’s public key, Bob sends all messages with Alice’s public key
- Kerberos
- Alice chooses session keys, and sends them to Bob, all encrypted with Bob’s public key

Internet Key Management

- Manual key management
  - Mandatory
  - Useful when IPsec developers are debugging
  - Keys exchanged offline (phone, email, etc.)

Internet Key Management

- Automatic key management
  - Two major competing proposals
  - Simple Key Management for Internet Protocols (SKIP)
  - Internet Security Association and Key Management Protocol (ISAKMP) + OAKLEY

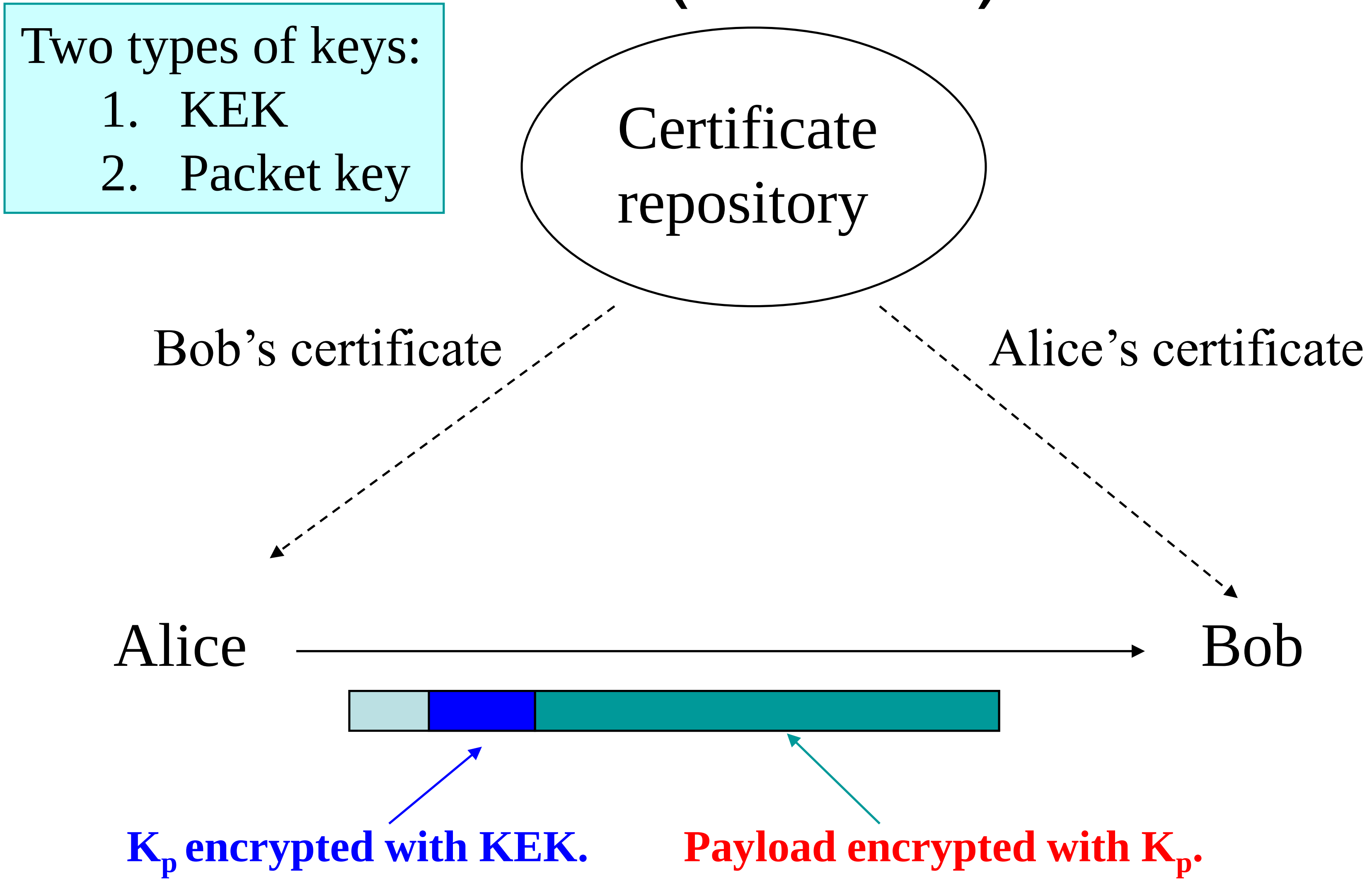
Automatic Key Management

- 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

- Idea
  - 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

K_p encrypted with KEK.

Payload encrypted with K_p.

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.

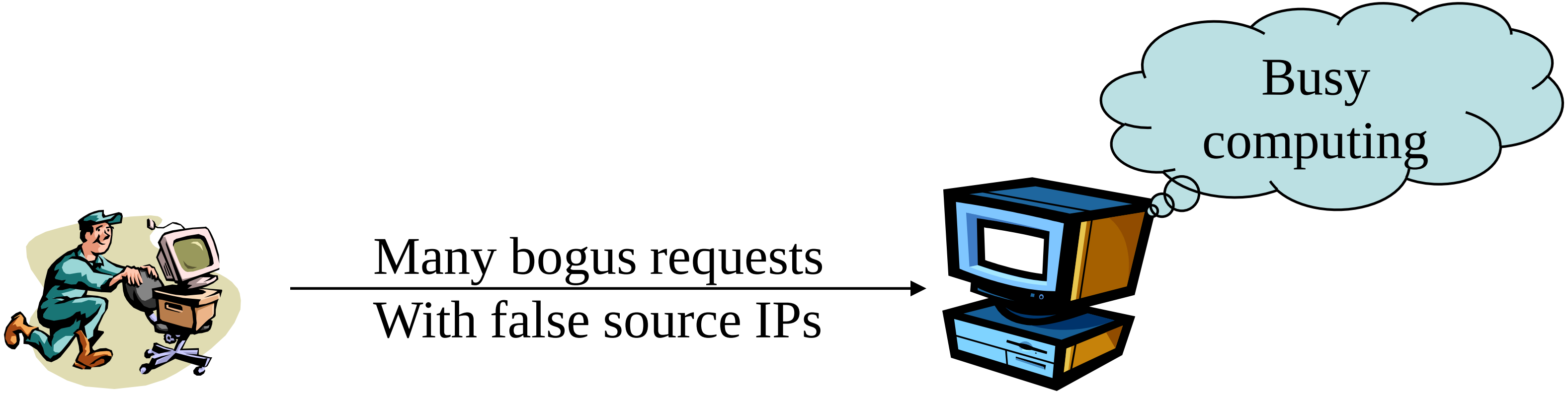
SKIP (Cont’d)

- Limitations
  - No Perfect Forward Secrecy
  - 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



Many bogus requests

With false source IPs

Busy computing

- 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)

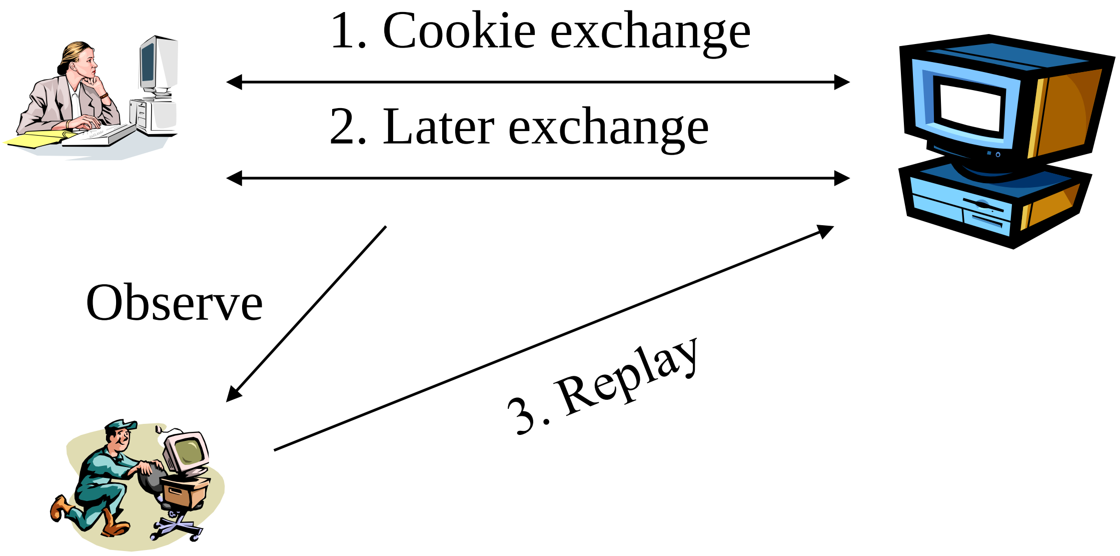
- Cookie
  - Each side sends a pseudo-random number, the cookie, in the initial message, which the other side acknowledges.
  - The acknowledgement must be repeated in the following messages.
  - Do not begin D-H calculation until getting acknowledgement for the other side.

Requirements for cookie generation

- An attacker cannot reuse cookies.
- Impossible to predict
  - 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



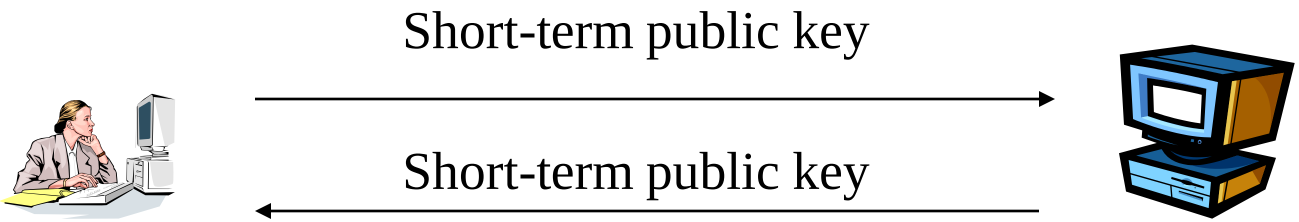
Man-In-The-Middle Attack

- 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

Ephemeral Diffie-Hellman



Short-term public key

Short-term public key

- Session key is computed on the basis of short-term DH public keys.
- Exchange of these short-term public keys requires authentication and integrity.
  - Digital signatures.
  - Keyed message digests.
- 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
  - Internet 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, authentication, etc.
  - 32 bit message ID
  - Variable number of payloads
    - Each has a generic header with
      - Payload boundaries
      - Next payload type (possible none)

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 Exchange Types

- 0 none
- 1 base
- 2 identity protection
- 3 authentication only
- 4 aggressive
- 5 informational

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 messages, 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

8	H	hash
9	SIG	signature
10	NONCE	nonce
11	N	notification
12	D	delete

IKE Overview

- IKE = ISAKMP + part of OAKLEY
  - 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

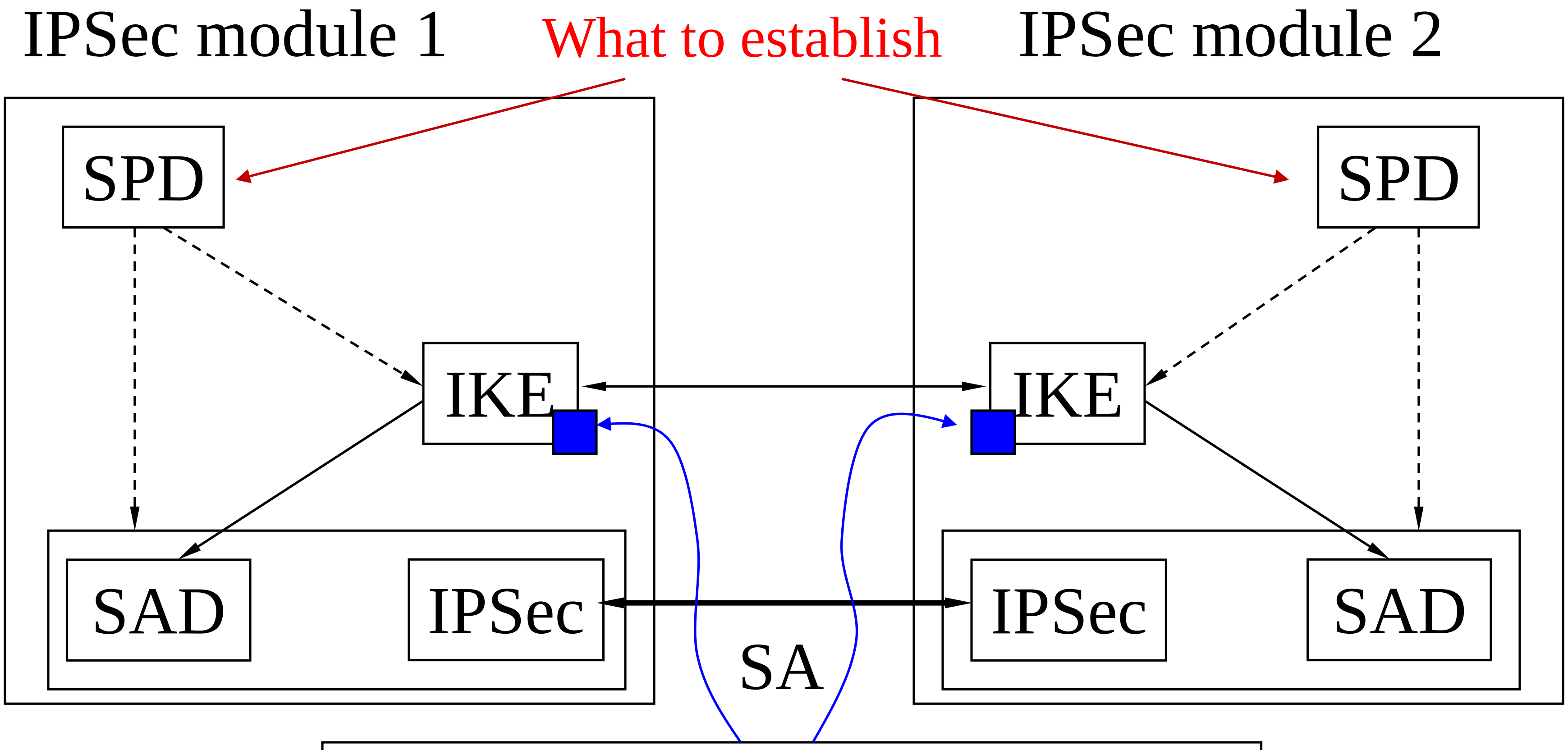
IKE Overview (Cont’d)

- Request-response protocol
  - Initiator
  - Responder
- Two phases
  - Phase 1: Establish an IKE (ISAKMP) SA
  - Phase 2: Use the IKE SA to establish IPsec SAs

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

IPSEC Architecture



IKE policies (How to establish the IPsec SAs):
1. Encryption algorithm; 2. Hash algorithm;
3. D-H group; 4. Authentication method.

A Clarification About PFS

- In RFC 2409:
  - Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will only permit access to 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.

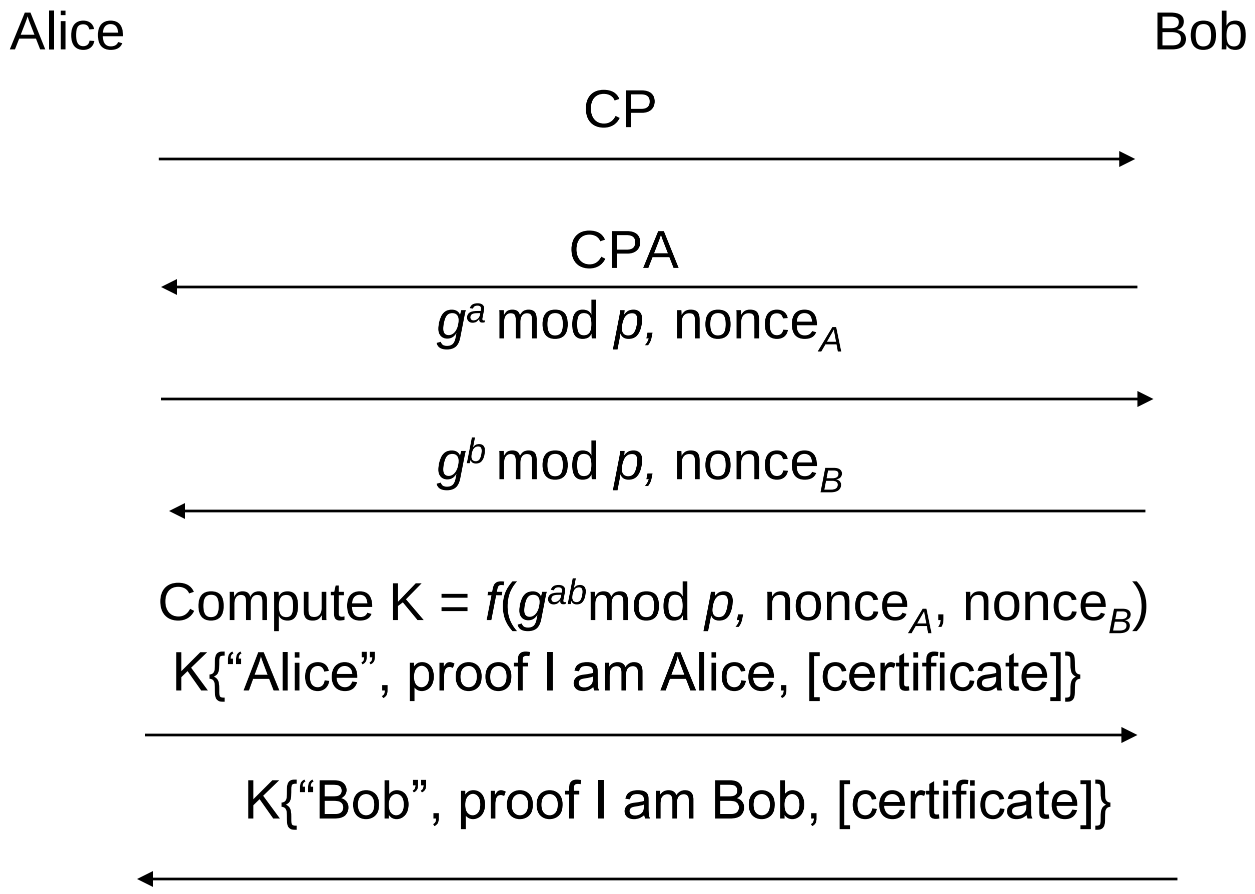
IKE Phase 1

- Negotiating cryptographic parameters
  - Specifies suites of acceptable algorithms:
    - {(3DES, MD5, RSA public key encryption, DH),
    - (AES, SHA-1, pre-shared key, elliptic curve), …}
  - Specifies a MUST be implemented set of algorithms:
    - Encryption=DES, hash=MD5/SHA-1, authentication=pre-shared key/DH
  - The lifetime of the SA can also be negotiated

IKE Phase 1

- Four authentication methods
  - Authentication with public signature key
  - Authentication with public key encryption
  - Authentication with public key encryption, revised
  - Authentication with a pre-shared key

IKE Phase 1:
Public Signature Keys, Main Mode



Alice

Bob

CP

CPA

g^a mod p, nonce$_A$

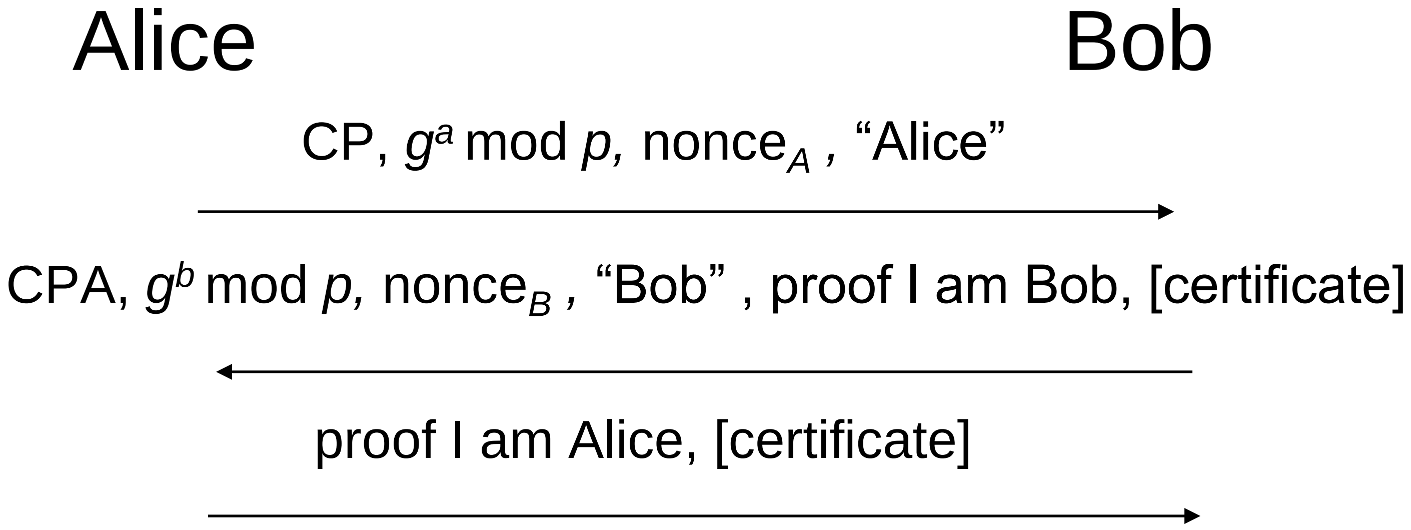
g^b mod p, nonce$_B$

Compute K = f(g^{ab}mod p, nonce$_A$, nonce$_B$)

K{“Alice”, proof I am Alice, [certificate]}

K{“Bob”, proof I am Bob, [certificate]}

IKE Phase 1: Public Signature Keys, aggressive Mode



Alice

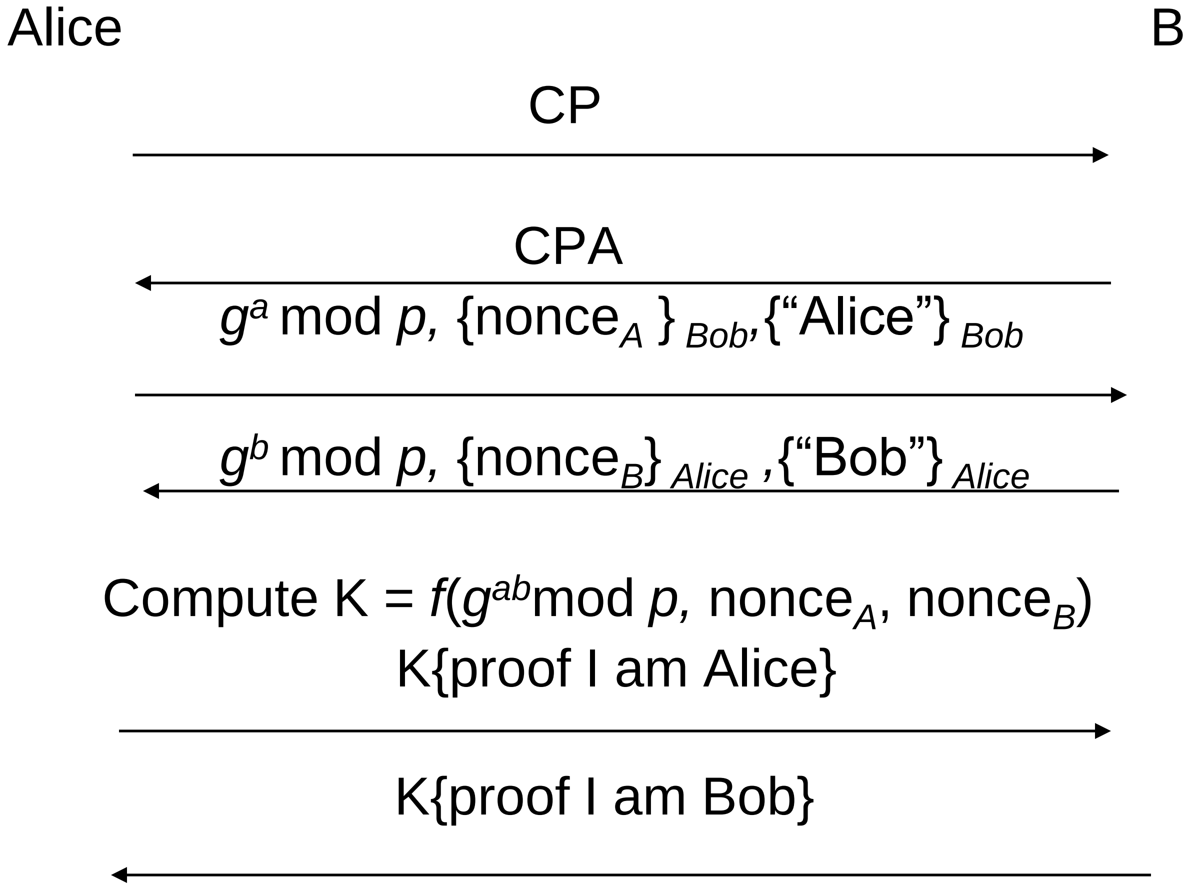
Bob

CP, g^a mod p, nonce$_A$, “Alice”

CPA, g^b mod p, nonce$_B$, “Bob” , proof I am Bob, [certificate]

proof I am Alice, [certificate]

IKE Phase 1:
Public Encryption Keys, Main Mode



Alice

Bob

CP

CPA

g^a mod p, {nonce$_A$ }$_{Bob}$,{“Alice”}$_{Bob}$

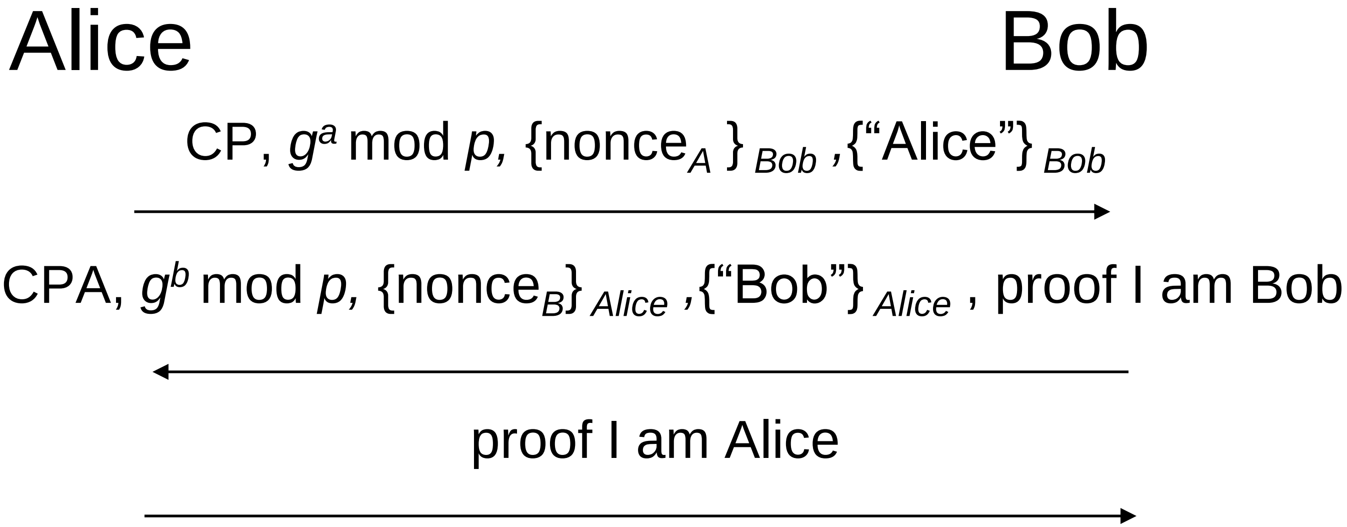
g^b mod p, {nonce$_B$}$_{Alice}$,{“Bob”}$_{Alice}$

Compute K = f(g^{ab}mod p, nonce$_A$, nonce$_B$)

K{proof I am Alice}

K{proof I am Bob}

IKE Phase 1: Public Encryption Keys, aggressive Mode



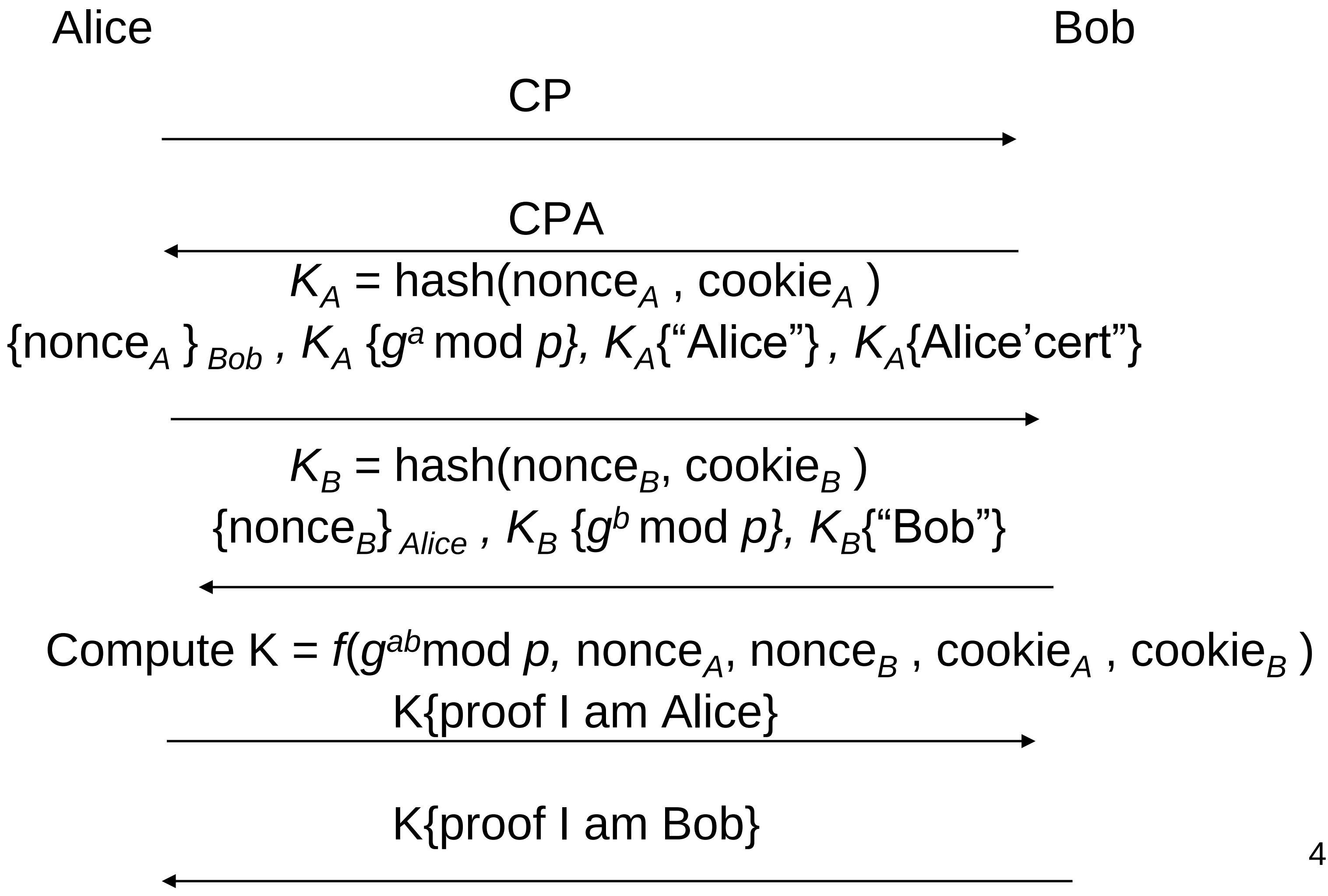
Alice Bob

CP, g^a mod p, $\{\text{nonce}_A\}_{Bob}$,{“Alice”}$_{Bob}$

CPA, g^b mod p, $\{\text{nonce}_B\}_{Alice}$,{“Bob”}$_{Alice}$, proof I am Bob

proof I am Alice

IKE Phase 1:
Public Encryption Keys(revised), Main Mode



Alice

Bob

CP

CPA

K_A = hash(nonce$_A$, cookie$_A$)

{nonce$_A$ }$_{Bob}$, K_A {g^a mod p}, K_A{“Alice”} , K_A{Alice’cert”}

K_B = hash(nonce$_B$, cookie$_B$)

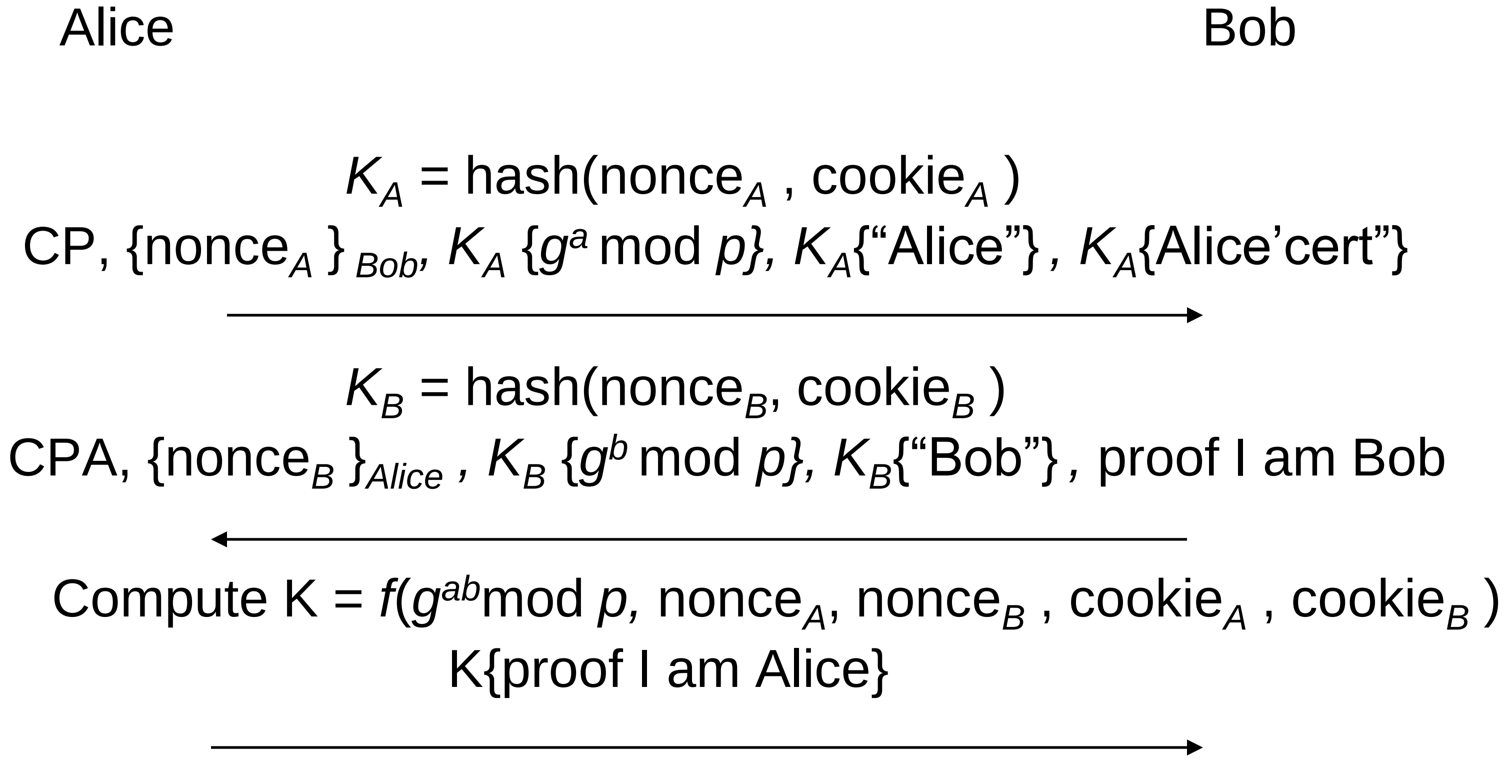
{nonce$_B$}$_{Alice}$, K_B {g^b mod p}, K_B{“Bob”}

Compute K = f(g^{ab}mod p, nonce$_A$, nonce$_B$, cookie$_A$, cookie$_B$)

K{proof I am Alice}

K{proof I am Bob}

IKE Phase 1: Public Encryption Keys(revised), Aggessive Mode



Alice

Bob

K_A = hash(nonce$_A$, cookie$_A$)

CP, {nonce$_A$ }$_{Bob}$, K_A {g^a mod p}, K_A{“Alice”} , K_A{Alice’cert”}

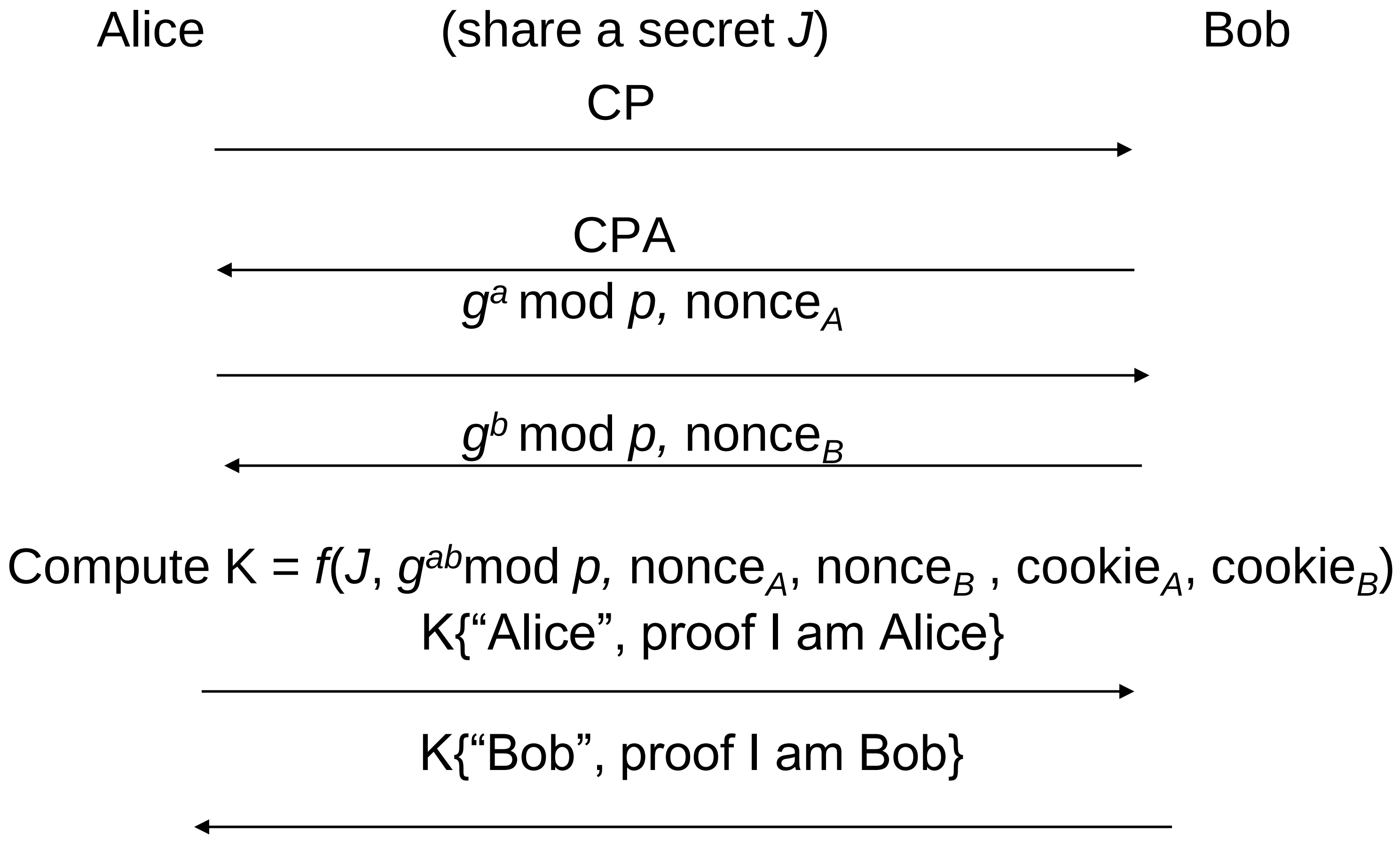
K_B = hash(nonce$_B$, cookie$_B$)

CPA, {nonce$_B$ }$_{Alice}$, K_B {g^b mod p}, K_B{“Bob”} , proof I am Bob

Compute K = f(g^{ab}mod p, nonce$_A$, nonce$_B$, cookie$_A$, cookie$_B$)

K{proof I am Alice}

IKE Phase 1: Pre-shared Secret, Main Mode



Alice (share a secret *J*) Bob

CP

CPA

$g^a \bmod p$, nonce$_A$

$g^b \bmod p$, nonce$_B$

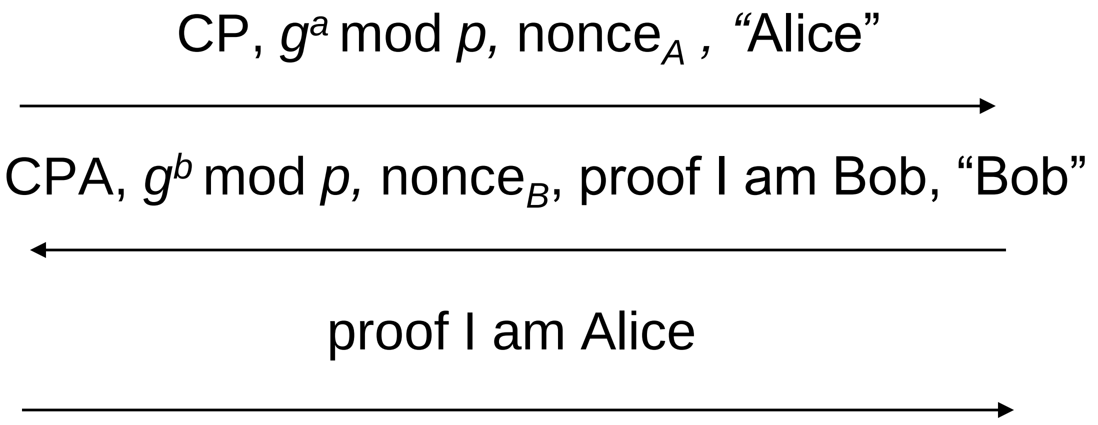
Compute K = $f(J, g^{ab} \bmod p$, nonce$_A$, nonce$_B$, cookie$_A$, cookie$_B)$

K{“Alice”, proof I am Alice}

K{“Bob”, proof I am Bob}

IKE Phase 1:
Pre-Shared secret, aggressive Mode

Alice (share a secret *J*) Bob



CP, g^a mod *p,* nonce$_A$, "Alice"

CPA, g^b mod *p,* nonce$_B$, proof I am Bob, "Bob"

proof I am Alice

IKE Phase 1: Establish a Shared Key

- 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: I’s (or R’s) nonce

IKE Phase 1: Establish a Shared Key (Cont'd)

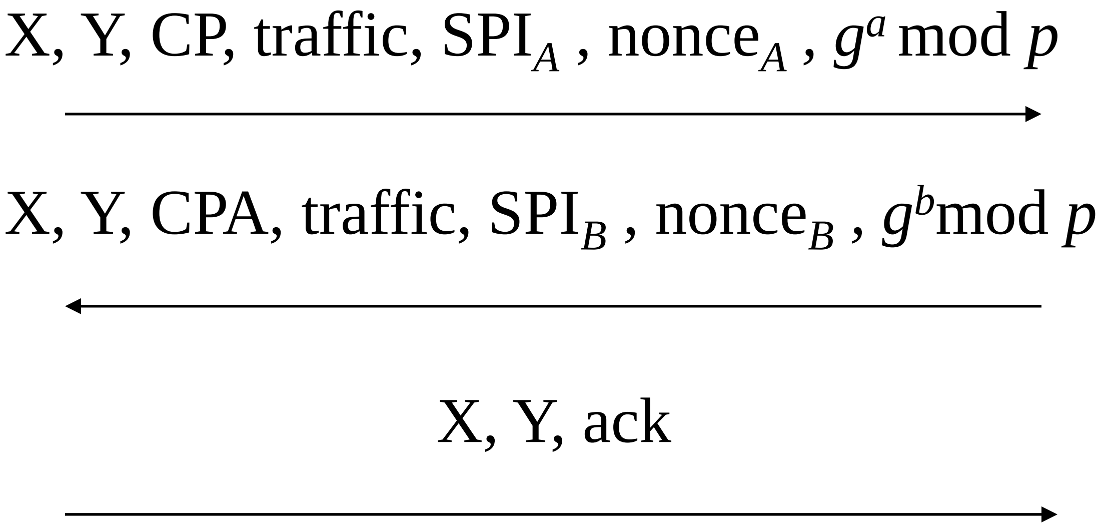
- Three groups of keys
  - Derived key for non-ISAKMP negotiations
    - SKEYID_d = prf(SKEYID, g^{xy} | CKY-I | CKY-R | 0)
  - Authentication key
    - SKEYID_a = prf(SKEYID, SKEYID_d | g^{xy} | CKY-I | CKY-R | 1)
  - Encryption key
    - SKEYID_e = prf(SKEYID, SKEYID_a | g^{xy} | CKY-I | CKY-R | 2)

IKE Phase 2 -- Quick Mode

- Negotiates parameters for the phase-2 SA
- Information exchanged with quick mode must be protected by the phase-1 SA
- Essentially a SA negotiation and an exchange of nonces
- Used to derive keying materials for IPsec SAs

IKE Phase 2 -- Quick Mode (Cont’d)

- 3-messages protocol



X, Y, CP, traffic, SPI_A , nonce_A , $g^a \bmod p$

X, Y, CPA, traffic, SPI_B , nonce_B , $g^b \bmod p$

X, Y, ack

IKE Phase 2 -- Quick Mode (Cont'd)

- All messages are encrypted using SKEYID_e, and integrity protected using SKEYID_a (except X, Y)
- Parameters:
  - X: pair of cookies generated during phase 1
  - Y: 32-bit number unique to this phase 2 session chosen by the initiator
  - DH is optional and could be used to provide PFS

Conclusion

- Perfect forward secrecy (PFS)
- SKIP
    - long term shared keys, no PFS
- Oakley
    - – a refinement of the basic Diffi-Hellman key exchange protocol.
- ISAKMP
    - Internet security association and key management protocol

– IKE

- Two phases, main and aggressive modes