IP Security (IPSEC) and Internet Key Exchange (IKE)

Anupama Potluri
Department of Computer and Information Sciences
University of Hyderabad

Overview

- Motivation for IP Security
- IP Security Architecture
- Inbound and Outbound Processing
- Authentication Header (AH)
- Encapsulating Security Payload (ESP)
- Scalability/Privacy Issues
- Internet Key Exchange (IKE)

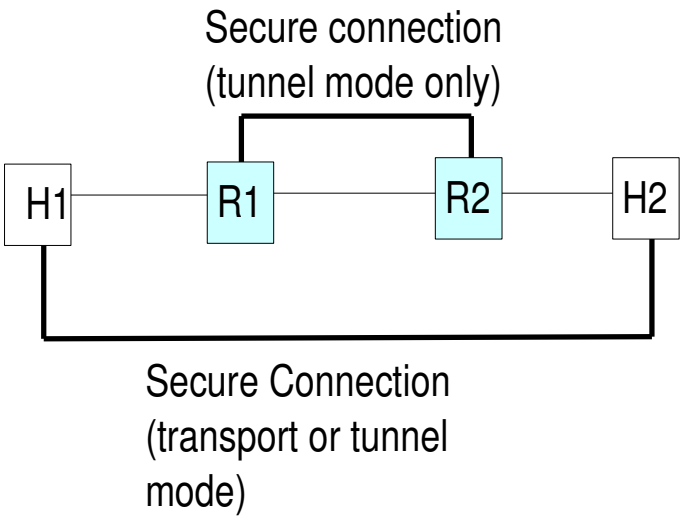
Motivation for IP Security

- Provide security at the network layer including
  - Access Control
  - Data Origin Authentication
  - Connectionless Integrity
  - Confidentiality
- Helps in establishment of Virtual Private Networks (VPNs)

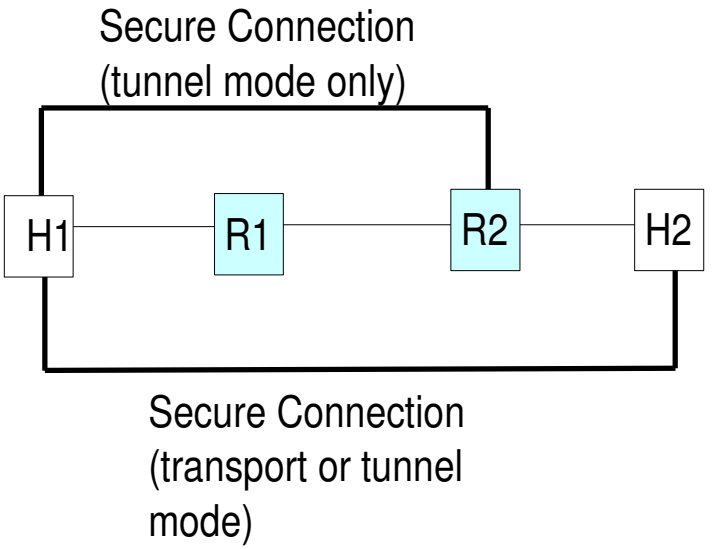
IP Security Architecture

- Components of IPSEC
  - Security Policy Database
  - Security Association Database
  - AH
  - ESP
  - IKE
- Transport and Tunnel Mode

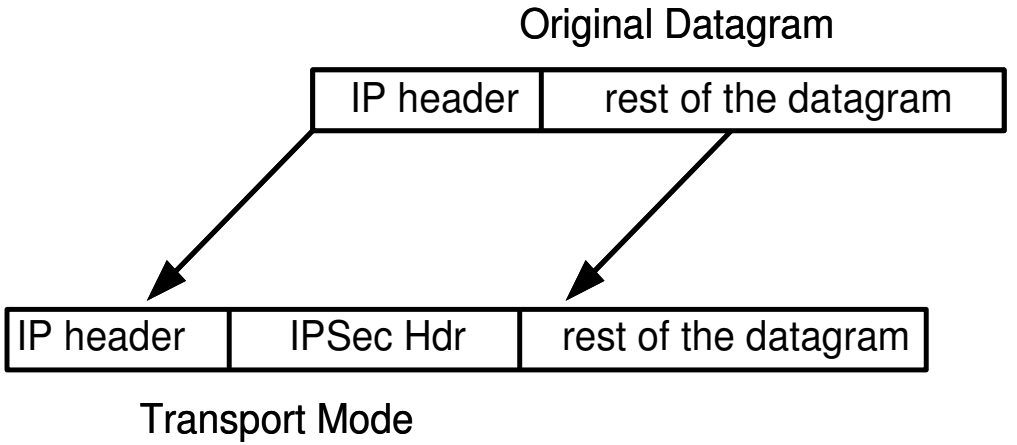
IPSEC Arch: Scenario 1



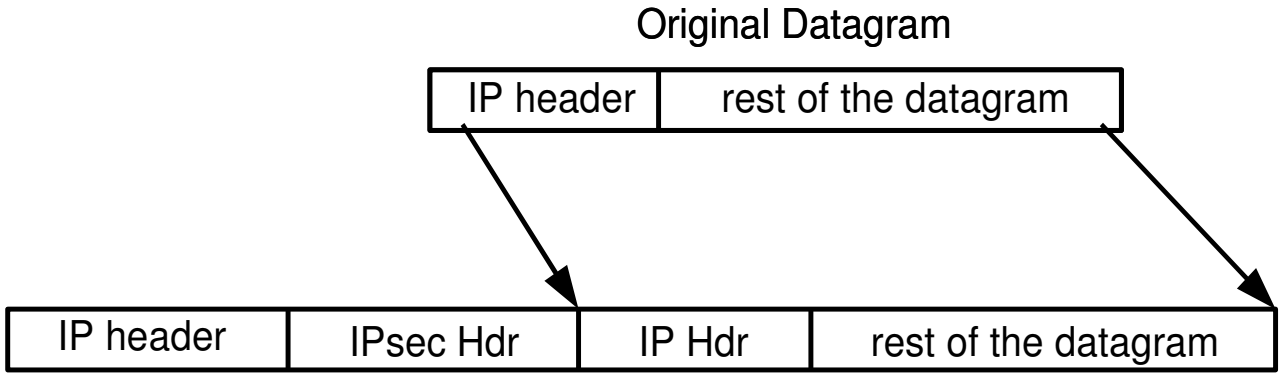
IPSEC Arch: Scenario 2



Transport Mode



Tunnel Mode


Original Datagram
IP header
rest of the datagram
IP header
IPsec Hdr
IP Hdr
rest of the datagram

Security Policy Database

- Defines the security policies of the enterprise
- Example entries:

Source	Dest	Protocol	Src Port	Dest Port	Policy	Sec.Serv.
A	B	TCP	*	80	Pass	None
C	B	TCP	*	22	Apply	ESP
*	B	TCP	*	80	Apply	AH

Security Association Database

- A Security Association is an instantiation of a security policy that is dynamically created and deleted.
- A single security policy can have many SAs since the policy can have a wildcard for any selector but a separate SA for each individual connection
- An SA is identified uniquely by the Destination Address, SPI and Security Protocol (AH or ESP)

Outbound Processing

1. Match the packet's selector fields against the outbound policies in the SPD to locate the first appropriate policy, which will point to zero or more SA bundles in the SAD.
2. Match the packet's selector fields against those in the SA bundles found in (1) to locate the first SA bundle that matches. If no SAs were found or none match, create an appropriate SA bundle and link the SPD entry to the SAD entry. If no key management entity is found, drop the packet.
3. Use the SA bundle found/created in (2) to do the required IPsec processing, e.g., authenticate and encrypt.

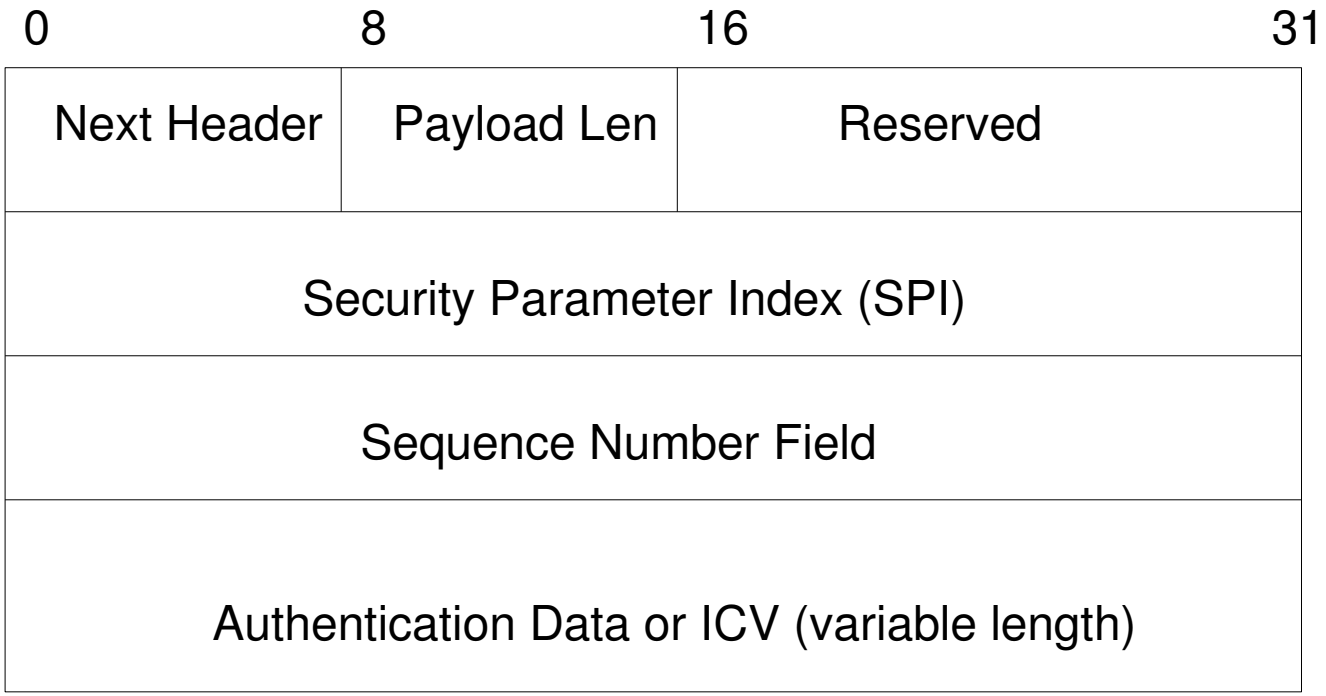
Inbound Processing

1. Use the packet's destination address (outer IP header), IPsec protocol, and SPI to look up the SA in the SAD. If the SA lookup fails, drop the packet and log/report the error.
2. Use the SA found in (1) to do the IPsec processing, e.g., authenticate and decrypt.
3. Find an incoming policy in the SPD that matches the packet.
4. Check whether the required IPsec processing has been applied.

Authentication Header

- Security Services provided are:
  - Data Origin Authentication
  - Connectionless Integrity
  - Anti-Replay

Authentication Header Format



Outbound Processing: Calculating the ICV

- The AH ICV is a one-way hash computed using SHA-1 or MD5 over:
  - IP header fields that are either immutable in transit or that are predictable in value upon arrival at the endpoint for the AH SA
  - the AH header (Next Header, Payload Len, Reserved, SPI, Sequence Number, and the Authentication Data (which is set to zero for this computation), and explicit padding bytes (if any))
  - the upper level protocol data, which is assumed to be immutable in transit

Mutable, Immutable and Predictable Fields

- **Immutable**
  - Version, Internet Header Length, Total Length, Identification, Protocol (This should be the value for AH.), Source Address, Destination Address (without loose or strict source routing)
- **Mutable but predictable**
  - Destination Address (with loose or strict source routing)
- **Mutable** (zeroed prior to ICV calculation)
  - Type of Service (TOS), Flags, Fragment Offset, Time to Live (TTL), Header Checksum

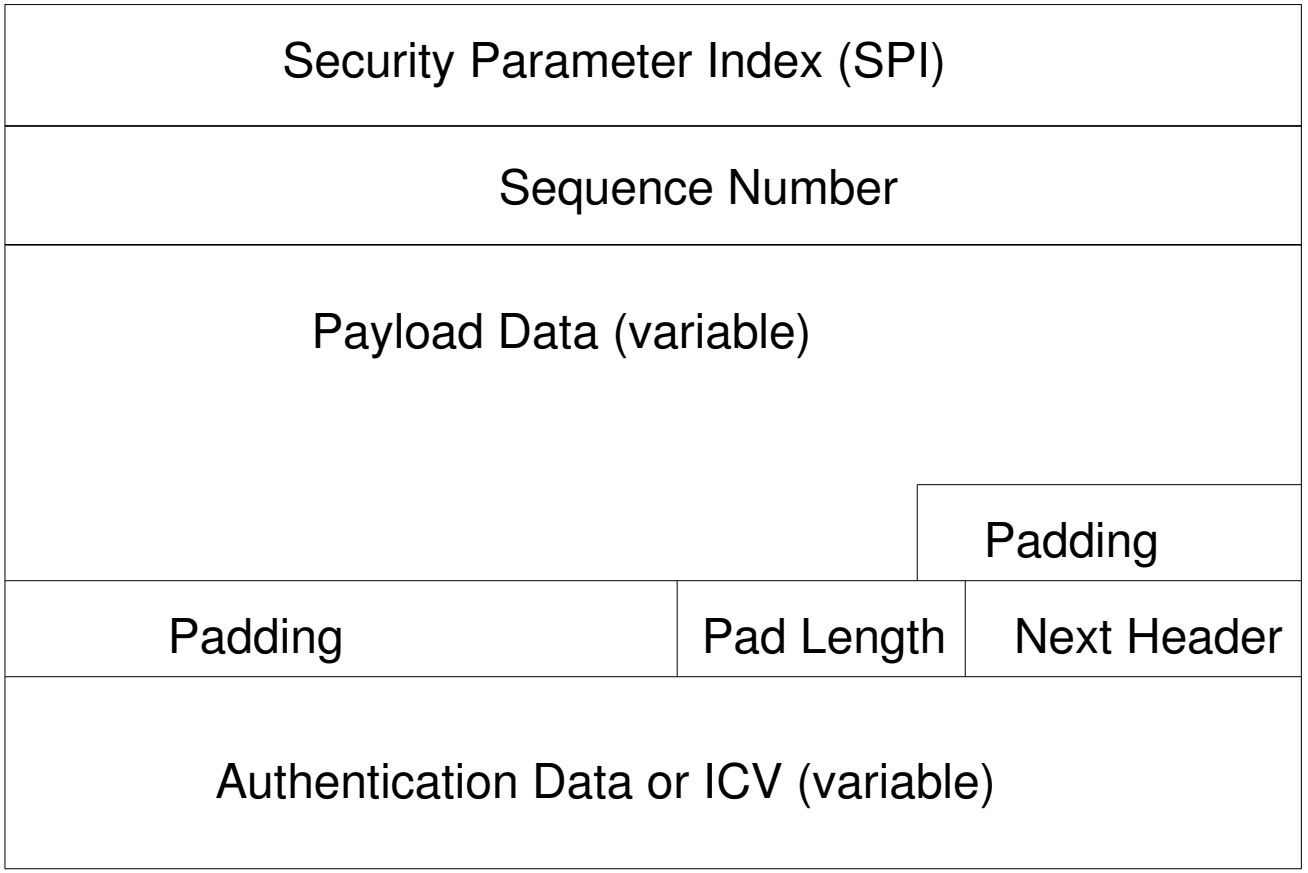
Inbound Processing

- Reassemble the packet, if required
- Security Association Lookup – (SPI, Destination and AH) as the key
- Verify the sequence number
- Verify the ICV

Encapsulating Security Payload

- Security Services provided are
  - Data Origin Authentication
  - Connectionless Integrity
  - Confidentiality
  - Anti-Replay
  - Limited Traffic Flow Confidentiality

ESP Header/Trailer Format



ESP Outbound Processing

1. encapsulates (into the ESP Payload field):
   - for transport mode, just the original upper layer protocol information.
   - for tunnel mode -- the entire original IP datagram.
2. adds any necessary padding.
3. encrypts the result (Payload Data, Padding, Pad Length and Next Header) using the key, encryption algorithm, algorithm mode indicated by the SA.

ESP Inbound Processing

1. decrypts the ESP Payload Data, Padding, Pad Length, and Next Header using the key, encryption algorithm, algorithm mode indicated by the SA.
2. processes any padding as specified in the encryption algorithm specification.
3. reconstructs the original IP datagram from:

   a) for transport mode -- original IP header plus the original upper layer protocol information in the ESPPayload field

   b) for tunnel mode -- tunnel IP header + the entire IP datagram in the ESP Payload field.

IPSEC - Summary

- IPSEC consists of
  - a security policy database that determines the type of **security applied** for that traffic in that enterprise
  - a **security association** that is a **specific instantiation** of a security policy
  - Based on the security needed, the AH and ESP protocols can be applied **either alone or in combination**
  - For Security Associations between **Security Gateways**, only **Tunnel Mode** is allowed

IPSEC Summary

- **Fragmentation** happens **after IPSEC** is applied at the source and **Reassembly before IPSEC** is applied at the destination
- All **Mutable fields are zeroed** before IPSEC is applied
- If the **integrity of the IP header** needs to be protected, **AH** is used
- If **Confidentiality** is needed, **ESP** is used.

Scalability/Privacy Issues

- AH and ESP use symmetric key cryptography using **shared keys**.
- Sharing of keys can be **manual** and stored in the systems.
- This is **non-scalable** as every pair of systems/users must have a unique key
- Privacy can be **compromised** if the system is compromised

Internet Key Exchange (IKE)

- IKE is a dynamic key exchange protocol that provides
  - authentication and confidentiality for the material exchanged to generate keys.
  - Perfect Forward Secrecy for Identities and Keys
- Has two phases:
  - Phase 1 : To negotiate the keys used to authenticate/protect the IKE exchange itself – **Main Mode** or **Aggressive Mode**
  - Phase 2 : To negotiate the keys used in IPSEC or any other security protocol - **Quick Mode**

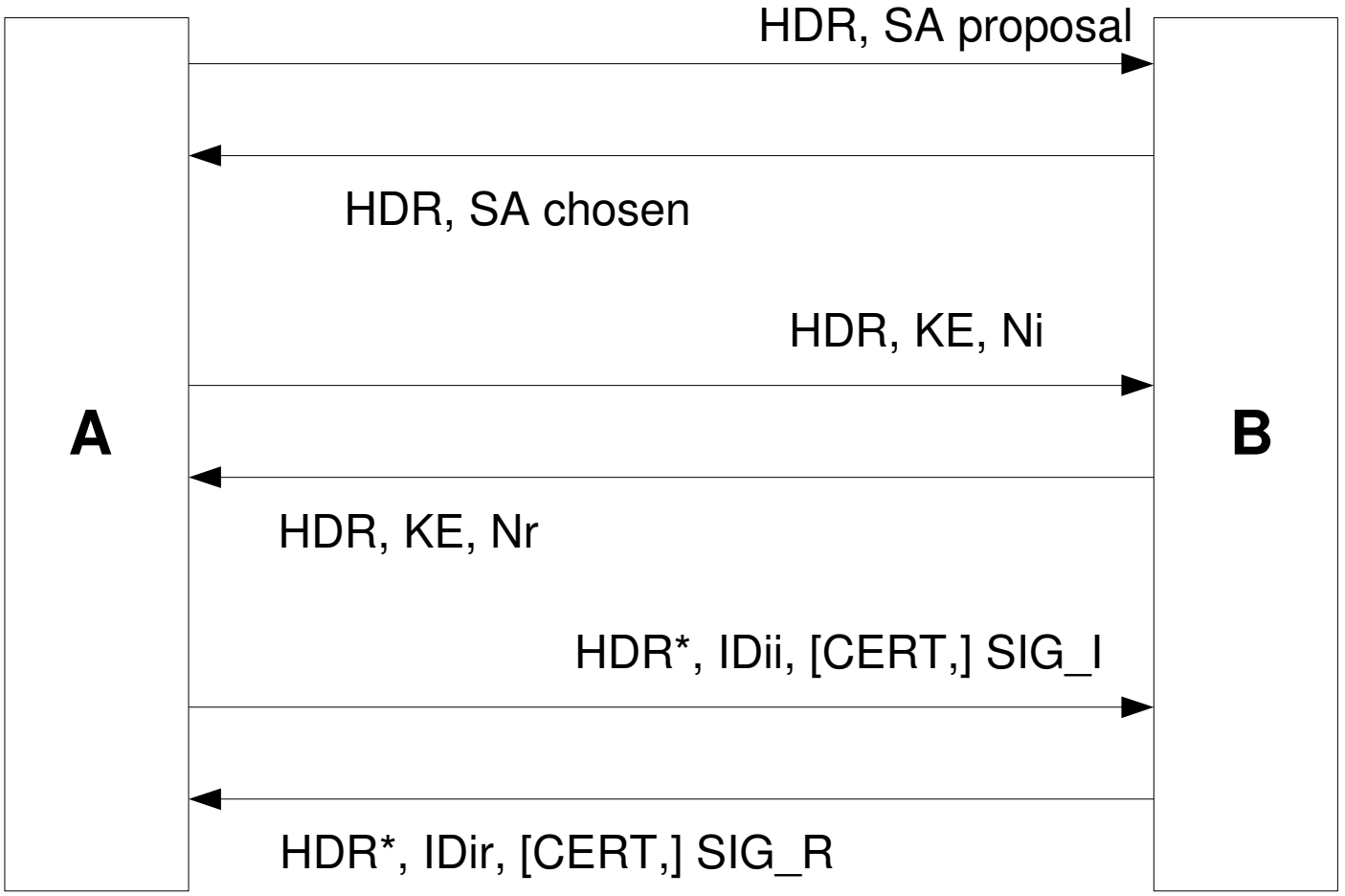
IKE - Payloads

- Fixed Header, HDR
- **Security Association Payload (SA)** – contains security proposals and transforms
- **Key Exchange Payload (KE)** – contains the Diffie-Hellman public keys
- **Nonce Payload (Ni and Nr)** – random numbers as protection against replay attacks
- **Identification Payload (IDii, IDir)** – Identity of the peers in the key exchange

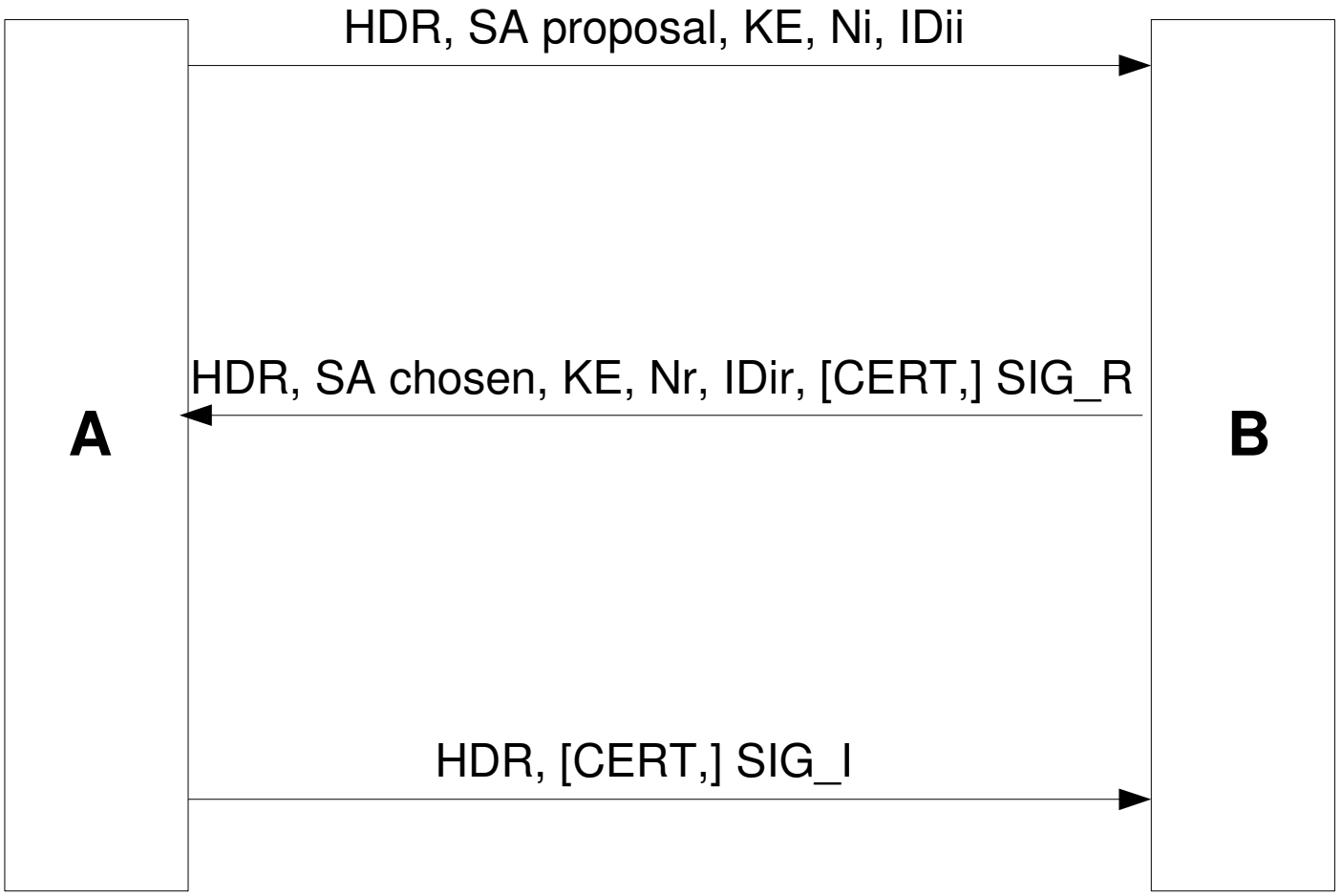
IKE Authentication Payloads

- **Signature Payload (SIG_I, SIG_R)** – contains digital signatures
- **Certificate Payload (CERT)** – contains the certificate mapping the identity to the public key signed by the Certificate Authority (CA)
- **Hash Payload (HASH_x)** – contains the one-way hash value using SHA-1 or MD5 as per the specification

IKE Main Mode – Auth. with Signatures



IKE Aggressive Mode – Auth. with Signatures



Generation of Keying Material

- The KE payload carries the Diffie-Hellman public keys ***g^x*** and ***g^y***. These are then used to generate ***g^xy*** which is used for generating keying material
- The generation is as follows:

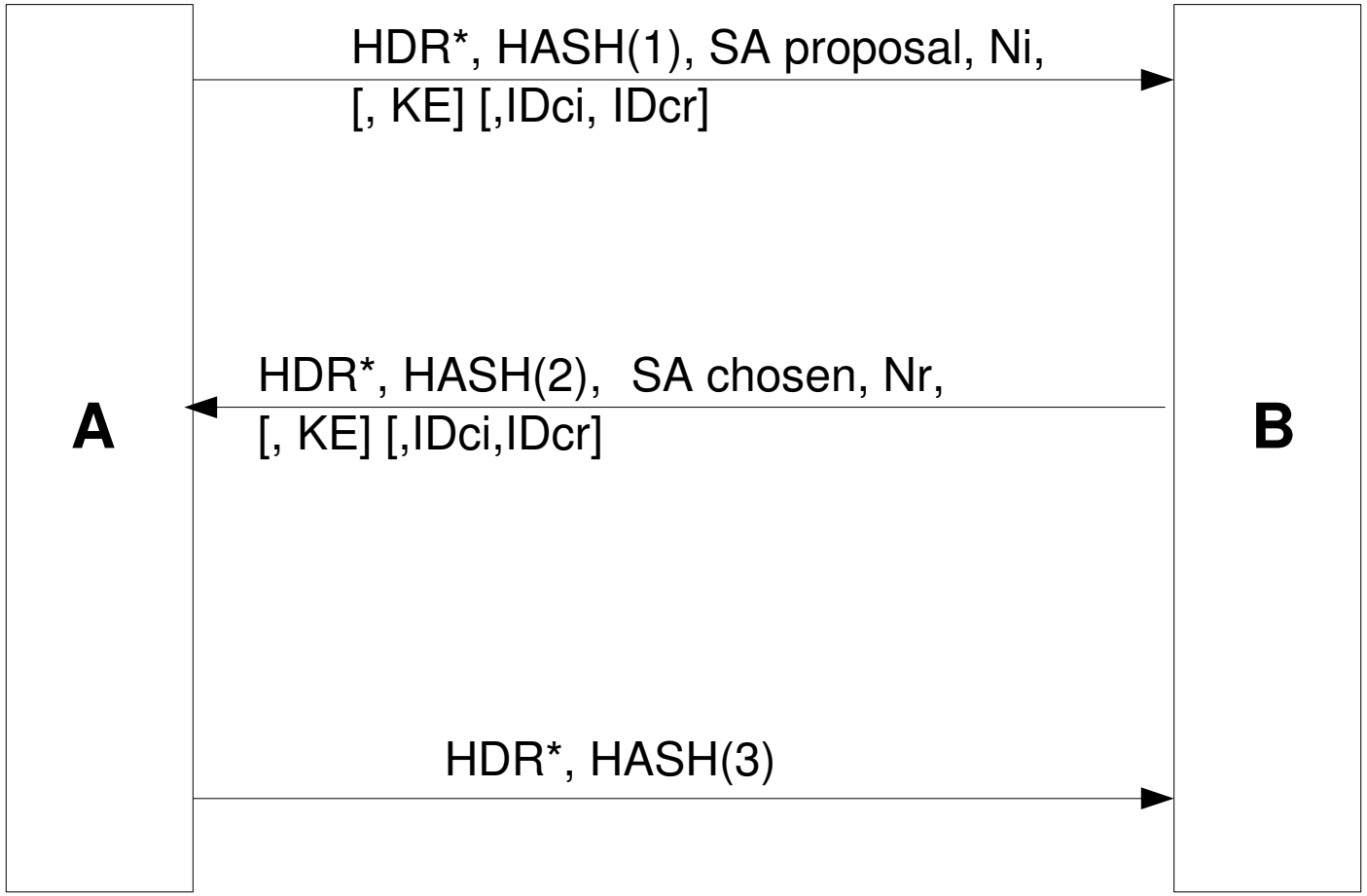
```
SKEYID = prf(Ni_b | Nr_b, g^xy)

SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)

SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I |
  CKY-R | 1)

SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I |
  CKY-R | 2)
```

Generation of Hash Values

- The **signatures** in the previous slides are calculated by **encrypting a hash** of the message using **the private key.** The hash values

```
HASH_I = prf(SKEYID, g^x | g^y | CKY-I | CKY-R
  | SAi_b | IDii_b )
HASH_R = prf(SKEYID, g^y | g^x | CKY-R | CKY-I
  | SAi_b | IDir_b )
```

IKE Quick Mode



SA, Proposal and Transform Payloads

NH = SA	Reserved	Payload Length	
Domain of Interpretation (DOI)			
Situation			
NH = Proposal	Reserved	Payload Length	
Proposal 1	PROTO_AH	SPI size =4	# Trans. = 1
SPI			
NH=Transform	Reserved	Payload Length	
Transform 1	AH_SHA	Reserved	
Attributes in TLV format (variable in length)			

Hash Calculation in Quick Mode

- The hash values seen in Quick Mode are calculated as follows:

```
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)
```

- Keying material for IPSEC SA is generated as follows:

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

Perfect Forward Secrecy

- **Perfect Forward Secrecy (PFS)** is defined as follows:
  - Compromise of a single key allows access to only data protected by a single key.
- This is achieved by ensuring that the key used to protect transmission of data **is not used** to generate keying material for future communication.

IKE - PFS

- IKE achieves
  - **PFS for keys** by having an additional Diffie-Hellman exchange as part of Quick Mode exchange and deleting a IPSEC SA after the session is done or a timeout occurs.
  - **PFS for Identities** by using Main Mode to protect identities and deleting an ISAKMP SA negotiated once the quick mode negotiation is completed.
  - **PFS for both Identities and Keys** by combining the two.

IKE Summary

- IKE is a key exchange protocol that allows for the **keying material to be exchanged with authentication** (and confidentiality, if required).
- It allows for **Perfect Forward Secrecy of Identities and Keys**
- It consists of two phases – **Phase 1** for the exchange of keying material to **protect phase 2** exchange and **Phase 2** for exchange of **keying material of IPSEC**

References

- **RFC 2401:** Security Architecture for the Internet Protocol by Stephen Kent, BBN Corporation and Ron Atkinson, @Home Network, Nov. 1998.
- **RFC 2402:** IP Authentication Header (AH) by Stephen Kent, BBN Corporation and Ron Atkinson, @Home Network, Nov. 1998.
- **RFC 2406:** IP Encapsulating Security Payload (ESP) by Stephen Kent, BBN Corporation and Ron Atkinson, @Home Network, Nov. 1998.
- **RFC 2409:** Internet Key Exchange (IKE) by Daniel Harkins and Dave Carrel, Cisco Systems Inc., Nov. 1998.
- **Network Security: Private Communication in a PUBLIC World** by Charlie Kaufman, Radia Perlman and Mike Speciner, Pearson Education, 2002.

Questions?