Lecture Introduction into Cyber Security

Internet Protocol Security (IPsec) (Part 2)

Asya Mitseva, M.Sc.
Prof. Dr.-Ing. Andriy Panchenko

Chair of IT Security
Brandenburg University of Technology Cottbus-Senftenberg

10 January 2019

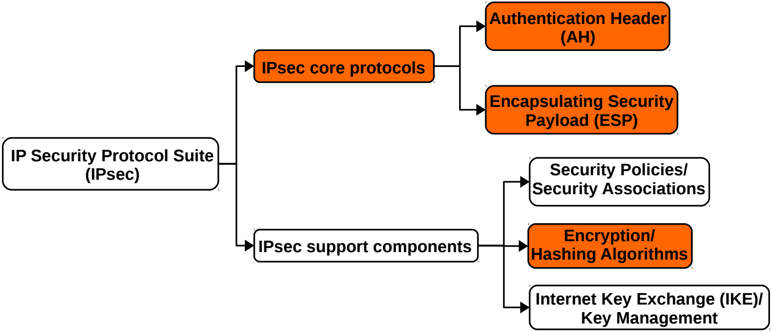
Brandenburgische Technische Universität Cottbus - Senftenberg

UNIVERSITÉ DU LUXEMBOURG

Erasmus+

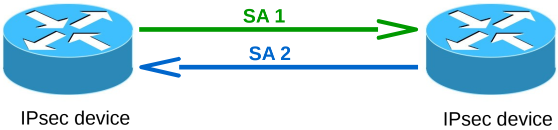
Recap: IPsec Overview

- Consist of two core protocols and set of supporting components
  - *Authentication Header (AH):* provides integrity & origin authentication
  - *Encapsulating Security Payload (ESP):* provides integrity, origin authentication, and encryption services
  - AH/ESP rely on pre-shared session keys & predefined crypto algorithms
  - *Supporting components:* specify mechanisms used for encryption and set up session keys for AH and ESP



Security Associations (SAs) (1/3)

- One-way logical connection between the sender and the receiver
- Determine the security services to the traffic on that connection
- Manually configured or negotiated through Internet Key Exchange
- Two SAs are required for bi-directional communication



- One SA can implement either AH or ESP *but not* both
- Combined use of AH and ESP requires *security association bundle*
  - Sequence of SAs through which traffic should be processed
- Sender stores several SAs for different receivers, types of traffic, etc.

Security Associations (SAs) (2/3)

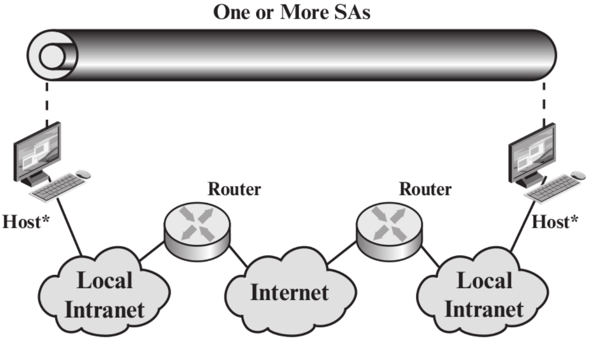
- SAs are stored in *Security Association database (SAD)*
- In SAD, SA is uniquely identified by
  - *Security Parameter Index (SPI)*: Carried in AH/ESP headers to enable receiver to select SA used to protect the packet
  - *IP Destination Address*: IP address of the receiver of the SA
  - *Security Protocol Identifier*: Shows if the association is AH or ESP SA
- IPsec host searches for the longest set of SA identifiers in SAD
- SAD contains parameters associated with each established SA
- SA is defined by the following parameters in SAD entry
  - *Security Parameter Index (SPI)*: Used to uniquely identify SA utilized, selected by the receiver
  - *Sequence Number Counter*: Used to generate the Sequence Number field in AH/ESP headers

Security Associations (SAs) (3/3)

- SA is defined by the following parameters in SAD entry
  - *Sequence Counter Overflow*: Indicate if sequence number overflow should generate auditable event
  - *Anti-Replay Window*: Determine initial slot and size of anti-replay window for this SA
  - *AH Information*: Authentication algorithm, keys, keys lifetime, etc. used for AH
  - *ESP Information*: Authentication and encryption algorithms, keys, keys lifetime, etc. used for ESP
  - *Lifetime of SA*: Time interval or byte count after which SA must be replaced with new SA or terminated
  - *IPsec Protocol Mode*: Tunnel, transport or wildcard
  - *Path MTU*: Max size of packet that can be transmitted without fragmentation

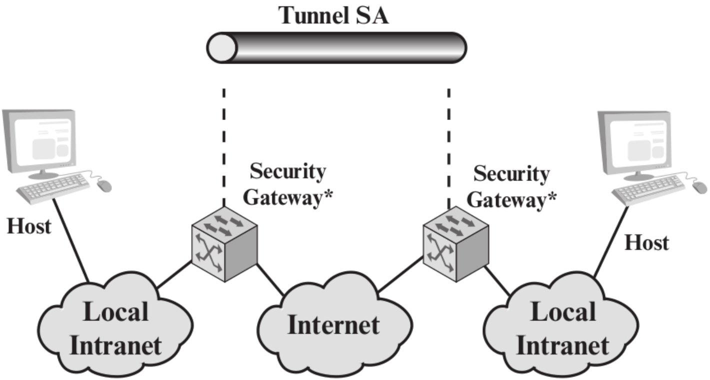
Combinations of Security Associations (1/4)

- **Case 1:** IPsec security between hosts
  - AH in transport mode
  - ESP in transport mode
  - ESP followed by AH in transport mode, i.e., ESP SA inside AH SA
  - Anyone of the use cases above inside AH or ESP in tunnel mode

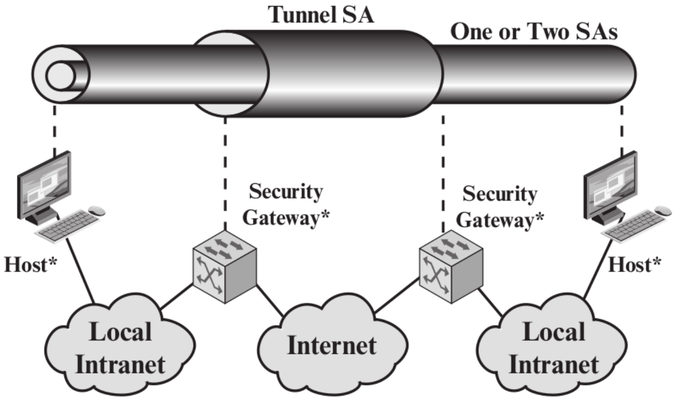


Combinations of Security Associations (2/4)

- **Case 2:** IPsec security between gateways, no hosts support IPsec
  - AH, ESP, or ESP with authentication in tunnel mode
  - Simple *Virtual Private Network (VPN)*

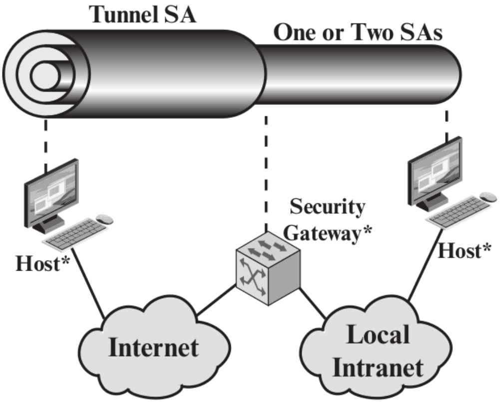


- **Case 3:** Combination of case 1 and case 2
  - ESP gateway-to-gateway tunnel with traffic flow confidentiality enabled
  - Hosts support IPsec services in transport mode



Combinations of Security Associations (4/4)

- **Case 4:** Remote host connects to company's firewall from outside
  - Tunnel mode is required between the remote host and the firewall
  - One or more SAs are used between the remote host and the local host



Security Policy Database (SPD) (1/2)

- IPsec provides flexibility with respect to which IPsec services are applied to which traffic
- SPD determines which traffic is related to specific SA or no SA (if the traffic is allowed to bypass IPsec)
- SPD entry is defined by a set of IP and upper-layer protocol field values, known as *selectors*
- SPD entry is determined by the following selectors
  - *Remote IP Address*: One or more IP addresses of the receiver(s)
  - *Local IP Address*: One or more IP addresses of the sender(s)
  - *Next Layer Protocol*: The protocol operating over IP
  - *Name*: User identifier from the operating system
  - *Local and Remote Ports*: One or more sender and receiver ports

Security Policy Database (SPD) (2/2)

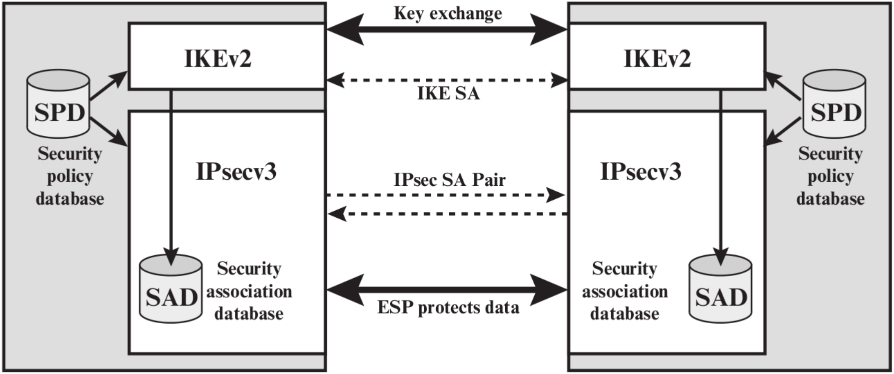
- Selectors are used to filter outgoing/incoming traffic & map it into SA

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

IP Traffic Processing

- **Alice sends IP packet to Bob**
  - Look up SPD to check if the packet should be protected with IPsec
  - SPD provides pointer to the corresponding SA entry in SAD
  - SA gives information about SPI, protocol, crypto algorithms, keys, etc.
  - Include the SPI in the encapsulated packet
- **Bob receives the packet from Alice**
  - Lookup the corresponding SA in SPD based on {SPI, destination IP address, security protocol identifier}
  - Based on SA, find crypto algorithms, keys, etc. in SAD
  - Decrypt the packet and check if it matches selectors in SPD

IPsec Architecture



Key Management of IPsec

- **Assure the determination and distribution of secret keys**
- **Support two types of key management**
  - *Manual:* System administrator manually configure his system with its own keys and the keys of other systems
  - *Automated:* Enable on-demand creation of keys for SAs, suitable for large distributed systems with evolving configuration
- **Automated IPsec key management protocols**
  - *Internet Key Exchange version 1 (IKEv1):* Consists of two protocols
    - *Oakley Key Determination Protocol:* Key exchange protocol based on Diffie-Hellman algorithm
    - *Internet Security Association & Key Management Protocol (ISAKMP):* Framework for Internet key management
  - *Internet Key Exchange version 2 (IKEv2)*

IKEv2: Overview

- **IKEv2 is refinement of Diffie-Hellman key exchange algorithm**
- **IKEv2 considers the following Diffie-Hellman (DH) weaknesses**
  - Does not provide any data about identities of parties
  - Man-in-the-middle attack possible
  - Vulnerable to *clogging attacks*
    - The opponent requests huge number of keys by using many spoofed IP addresses
    - The victim spends considerable resources to do useless computations
- **Security features provided by IKEv2**
  - Use of *groups* to specify global parameters for Diffie-Hellman
  - Use of *cookies* to thwart clogging attacks
  - Use of *nonces* to protect against replay attacks
  - *Authenticate Diffie-Hellman* exchange to thwart man in the middle

IKEv2 Security Features (1/2)

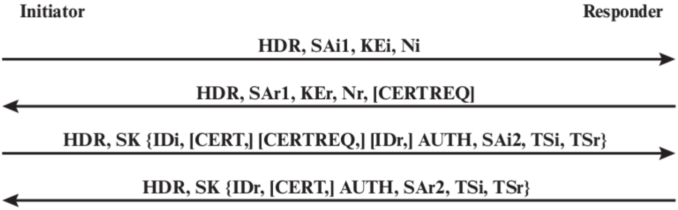
- **Groups specifying global parameters for Diffie-Hellman**
  - Include definition of two global parameters and identity of the algorithm
- **Use of cookies to thwart clogging attacks**
  - Receiver stores sender's state in unforgeable cookie
  - The cookie is sent to the sender
  - If the IP address of the sender is not spoofed, it will obtain the cookie and respond by putting the corresponding cookie in message
  - Cookie is regenerated by the receiver and compared with the cookie returned by the sender
- **Use of nonces to protect against replay attacks**
  - Locally generated pseudo-random numbers
  - Transmitted encrypted during certain portions of the exchange

IKEv2 Security Features (2/2)

- **Authentication of Diffie-Hellman exchange**
  - Use of *digital signatures*
    - Signing mutually obtainable hash
    - Each peer encrypts the hash with its private key
    - The hash is generated over parameters such as user IDs, nonces
  - Use of *public-key encryption*
    - Peers encrypt parameters, such as user IDs, nonces, with its private key
  - Use of *symmetric-key encryption*
    - Peers encrypt parameters, such as user IDs, nonces, with symmetric key
    - The key is derived out-of-band

IKEv2: Key Exchange (1/4)

- **Initial exchange**
  - Peers exchange data about security parameters and DH values
  - Set up *IKE SA* defining security parameters for subsequent IKE message exchanges
  - Set up *initial SA* for regular IPsec communication, known as *child SA*



HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie–Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

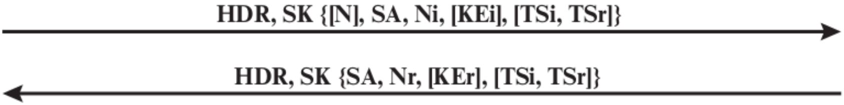
IKEv2: Key Exchange (2/4)

- **Initial exchange**

1. The initiator informs the responder about set of supported crypto algorithms (SA_i1), its Diffie-Hellman value (KE_i), and its nonce (N_i).
2. The responder informs the receiver about its choice of crypto algorithms from SA_i1 (SA_r1), its Diffie-Hellman value (KE_r), and its nonce (N_r). It also requests proof of initiator's identity ($CERTREQ$).
3. The initiator and the responder can compute shared but unauthenticated shared key by using the shared nonces, KE_i, and KE_r.
4. The initiator asserts its identity (ID_i), sends its certificate ($CERT$), requests responder's certificate ($CERTREQ$), and specifies to which of the responder's identity (ID_r) it wants to talk. It also announces set of supported crypto algorithms (SA_i2) for IPsec SA and which portions of traffic will be protected by the SA (TS_i, TS_r). The message is encrypted and authenticated.
5. The responder asserts its identity (ID_r), sends its certificate ($CERT$), and completes the IPsec SA negotiation.

IKEv2: Key Exchange (3/4)

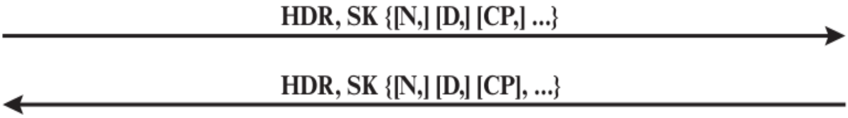
- **Child SA exchange**
  - Used to create new child SAs and to rekey both IKE SAs and child SAs
  - [*N*] indicates which SA is being rekeyed
  - SA is rekeyed by creating a new SA and then deleting the old one



HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie–Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

- **Informational exchange**
  - Used to exchange management information, IKE error messages, other notifications

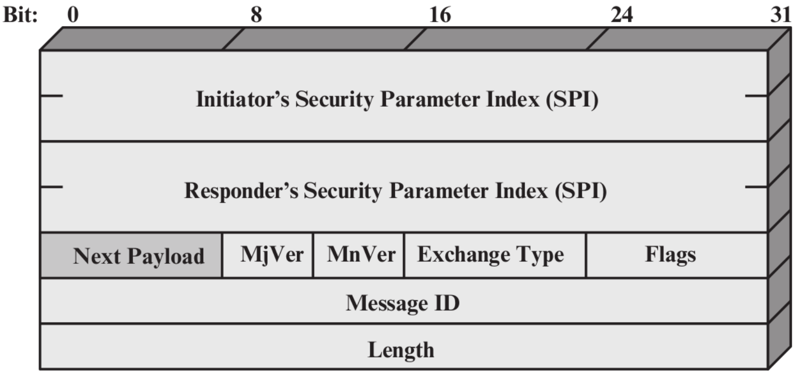


HDR = IKE header
SAx1 = offered and chosen algorithms, DH group
KEx = Diffie–Hellman public key
Nx= nonces
CERTREQ = Certificate request
IDx = identity
CERT = certificate

SK {...} = MAC and encrypt
AUTH = Authentication
SAx2 = algorithms, parameters for IPsec SA
TSx = traffic selectors for IPsec SA
N = Notify
D = Delete
CP = Configuration

IKE Packet

- Consist of one IKE header and one or more payloads
- Carried in UDP (User Datagram Protocol) datagram
- IKE header format



IKE Packet Header Format

- *Initiator SPI*: Identify unique IKE SA chosen by the initiator
- *Responder SPI*: Identify unique IKE SA chosen by the responder
- *Next Payload*: Identify the type of the first payload in the message
- *Major Version*: Define major version of IKE in use
- *Minor Version*: Define minor version of IKE in use
- *Exchange Type*: Identify the type of exchange
- *Flags*: Identify specific options set for this IKE exchange
- *Message ID*: Used to control retransmissions of lost packets and match requests to responses
- *Length*: Length of the total message

IKE Payload Types

- **SA Payload:** Used to begin establishment of SA
  - Consists of *proposal*, *transform*, and *attributes*
- **Key Exchange Payload:** Set up session key, different key exchange techniques are supported
- **Authentication Payload:** Contain data for message authentication purposes
- **Notify payload:** Contain either error or status information associated with given SA
- **Delete payload:** Indicate one or more SAs deleted by the sender and no longer valid
- **Traffic Selector Payload:** Peers can identify which packet flows are IPsec protected
- **Etc.**

IPsec: Conclusion

- IPsec provides transparent security for users of IP
- IPsec consists of two core protocols and set of supporting elements
  - AH assures integrity protection and origin authentication
  - ESP assures integrity, origin authentication, and payload encryption
  - SAD and SPD define which IPsec security services should be applied to which traffic
  - IKE provides automated establishment and management of key material
- IPsec provides *host-to-host* security
- IPsec does *not* provide *user-to-user* or *application-to-application* security

References

- William Stallings, *Cryptography and Network Security*, Chapter 20
- Charlie Kaufman et al., *Network Security: Private Communication in a Public World*, Chapter 16, 17
- RFC 4301, https://tools.ietf.org/html/rfc4301
- RFC 4302, https://tools.ietf.org/html/rfc4302
- RFC 4303, https://tools.ietf.org/html/rfc4303
- RFC 4835, https://tools.ietf.org/html/rfc4835
- RFC 2410, https://tools.ietf.org/html/rfc2410
- RFC 6071, https://tools.ietf.org/html/rfc6071