Security Architecture for IP (IPsec)

Security Association (SA), AH-Protocol, ESP-Protocol
Operation-Modes, Internet Key Exchange Protocol (IKE)

Agenda

- **<u>Overview</u>**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **Internet Key Exchange Protocol** **(IKEv1, RFC 2409)**
- **IPsec / IKEv1 Problems**

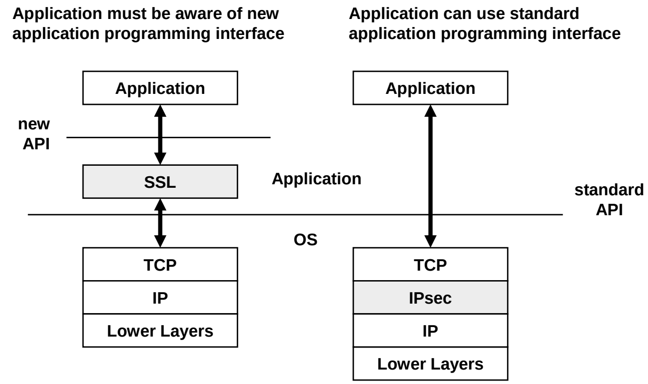
Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IP Security Discussion Raise with IPv6

- **End-to-end security**
  - will become more and more important when Internet goes to the commercial world
  - e.g. shopping in the Internet (save transmission of credit card numbers, electronic money, identity of the sender in case of an order via Internet, integrity of electronic bills etc.)
- **Question was**
  - if the next generation IP protocol (IPv6) should provide end-to-end security as integral part of itself

Which Layer for Security?



IPv6 Security Aspects

- **After heated discussions IESG decided**
  - basic building blocks (without non-repudiation) of network security should be part of IPv6 functionality
  - a vendor of an IPv6 implementation must include support of these basic building blocks in order to be standard-compliant
    - does not mean that the use of authentication and encryption blocks is required; only support must be guaranteed
  - IPv6 security follows the general IPsec recommendations
    - RCF 2401 (Former RFC 1825) Security Architecture for IP (IPv4 and IPv6)
  - difference of security aspects between IPv4 and IPv6
    - security in IPv6 is an integral part of it
    - security in IPv4 must is an add on

End-to-End Security

- **Basic building blocks for end-to-end security**
  - authentication and integrity
    - authentication provides identity of sender
    - integrity ensures that senders message was not changed on the way through the network
  - confidentiality or privacy
    - message cannot be read by others than authorized receiver
  - non-repudiation (!!! not covered by IPsec !!!)
    - the sender cannot later repudiate the contents of the message
- **Techniques used for these building blocks**
  - HMAC and Digital Signature (CA, RSA)
  - encryption (DES, 3DES, IDEA, AES)
  - key distribution techniques (DH, CA, IKE, KDC)

Security Architecture for IP

- **The goal of the IPsec architecture**
  - provision of various security services for traffic at the IP layer in both IPv4 and IPv6 environments
  - in a standardized and universal way
  - „Security Framework“
- **Before IPsec**
  - existing solutions were mostly on the application layer (SSL, S/MIME, ssh, …)
  - existing solutions on the network layer were all propriety
    - e.g. it was complicated, time demanding and expensive to establish multi-application or multi-vendor virtual private networks (VPNs)

Elements of IPsec 1

- **Security Architecture for IP**
    - originally defined in RFC 2401
    - obsoleted by RFC 4301 since Dec 2005
  - describes how to provide a set of security services for traffic at the IP layer
  - describes the requirements for systems that implement IPsec, the fundamental elements of such systems, and how the elements fit together and fit into the IP environment
  - it also describes the security services offered by the IPsec protocols, and how these services can be employed in the IP environment.
  - Security Associations (SA)
    - what they are and how they work, how they are managed and their associated processing
  - Security Policy Database (SPD)
  - Security Association Database (SAD)

Elements of IPsec 2

- **Security Protocols (for traffic security)**
  - Authentication Header (AH)
    - defined in RFC 2402
    - obsoleted by RFC 4302, 4305 since Dec 2005
  - Encapsulating Security Payload (ESP)
    - defined in RFC 2406
    - obsoleted by RFC 4303, 4305 since Dec 2005
  - Cryptographic Algorithm Implementation Requirements for ESP and AH
    - RFC 4305 -> mandatory algorithm are defined
  - Secret-key algorithms are used so far because of performance reasons
    - HMAC-SHA1, HMAC-MD5, AES-XCBC-MAC, DES-CBC, 3DES-CBC, AES-CBC, AES-CTR
    - defined in a lot separate RFC's (see 4305, 2403, 2404, 2405, 2451, 3566, 3602, 3686)

Elements of IPsec 3

- **Management of Security Associations and Keys**
  - manual for static and small environments
  - automatic for scalable environments by ISAKMP
  - ISAKMP (Internet Security Association and Key Management Protocol)
    - defined in RFC 2408
    - obsoleted by RFC 4306 since Dec 2005
  - Internet Key Exchange (IKE) for ISAKMP
    - defined in RFC 2409 (aka IKEv1)
    - obsoleted by RFC 4306 since Dec 2005 -> **IKEv2** !!!
  - Domain of Interpretation (DOI)
    - defined in RFC 2407
    - obsoleted by RFC 4306 since Dec 2005

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

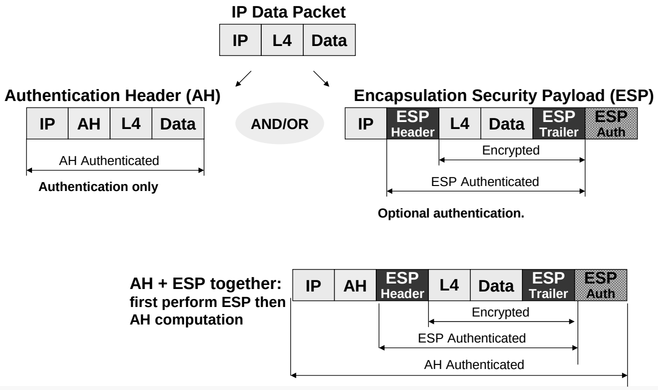
What IPsec does?

- **IPsec enables a system**
  - to select required security protocols, determine the algorithm's to use for the services, and put in place any cryptographic keys required to provide the requested services
- **IPsec can be used**
  - to protect one or more "paths" between a pair of hosts, between a pair of security gateways (VPN), or between a security gateway and a host
  - security gateway could be for example, a router or a firewall implementing IPsec
    - VPN concentrator is another name for such a device if several SA pairs are terminated at the same point
    - VPN is by far the most usual application of IPsec

What IPsec does?

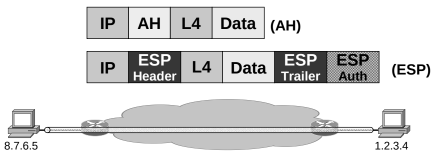
- **The set of security services that IPsec can provide includes**
  - access control
    - prevents unauthorized use of a resource
    - the resource to which access is being controlled is
      - for a host -> computing cycles or data
      - for a security gateway -> network behind the gateway or bandwidth on that network
  - connectionless integrity
    - detects modification of individual IP datagram's
  - data origin authentication
  - rejection of replayed packets (optional)
    - detects arrival of duplicate IP datagram's within a constrained window
  - confidentiality (encryption)
  - all these services are provided at the IP layer
    - hence they can be used by any higher layer protocol e.g., TCP, UDP, ICMP, BGP, etc.

IPsec Headers



IPsec Modes 1

Transport Mode
- Only one IP header.
- Used for end-to-end sessions
- Does not hide communication statistics because of network header (IP addresses of the end systems) is sent in clear text



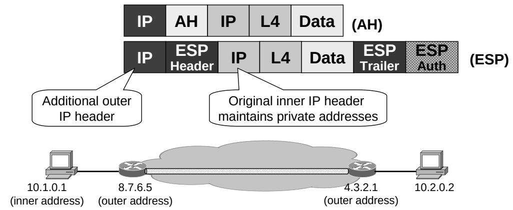
Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IPsec Modes 2

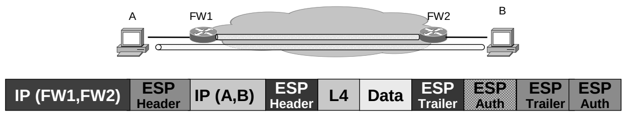
Tunnel Mode
for Site-to-Site VPN

- Whole original IP packet is IPsec-encapsulated.
- Used for VPNs.
- Does hide traffic patterns!

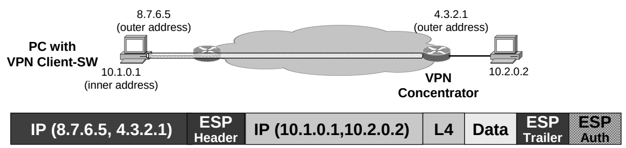


IPsec Modes 3

Tunnel Mode and
Transport Mode (ESP only)



Tunnel Mode for
Client-to-Site VPN



IPsec in Praxis

- **"IPsec used anywhere"**
  - Firewall, Router, Hosts
  - VPN
    - Site-to-Site
    - Remote-to-Site
    - Client-to-Site
  - Scalable solutions available
  - Easy to implement
  - Defined for end-to-end security but not frequently used between end systems
- **Encryption performance**
  - Original standards: DES and Triple-DES
  - Today migration to AES already done (more efficient, longer keys)
  - HW versus SW encryption power
    - e.g. crypto engines on router for higher performance

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **Internet Key Exchange Protocol** **(IKEv1, RFC 2409)**
- **IPsec / IKEv1 Problems**

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

AH Security Service (RFC 2402, 4302)

- **AH provides**
  - IP datagram sender authentication by HMAC or MAC
  - IP datagram integrity assurance by HMAC or MAC
  - replay detection and protection via sequence number (optional)
- **AH does not provide**
  - non-repudiation because of usage of secret-keys (shared keys) for HMAC or MAC
    - note: a real Digital Signature needs usage of public-key technique by signing a message with the private-key
  - confidentiality (encryption)
  - authentication for IP fragments
    - therefore IP fragments must be assembled before authentication is checked (better avoid it by MTU path discovery)

IPv4 and AH

0	4	8	16	31
Vers.=4	HLEN	ToS or DSCP	Total Length	
Fragment Identifier			Flags	Fragment Offset
TTL		protocol = 51	Header Checksum	
Source Address				
Destination Address				
IP Options				Pad
First 32 bits of AH				
....				
Last 32 bits of AH				
Payload				
....				

Authentication Header (AH)

	0	8	16	24 31
0	**Next Header**	**Length**	**Reserved**	
4	**Security Parameters Index (SPI)**			
8	**Sequence Number**			
10	**Authentication Data (variable number of 32-bit words)**			

note: AH was originally defined as extension header for IPv6 and later same structure was also used for IPv4

Authentication Header (AH)

- **Next Header (8 bits)**
  - indicates the next header following the AH header
  - same values allowed as protocol field in IPv4 header
    - IP in IP (4), TCP (6), UDP (17), ICMP (1), OSPF (89), etc
    - next header value of immediately preceding header = 51 (AH)
- **Length**
  - length of AH header
    - number of 32-bit words
- **Security Parameter Index**
  - a 32-bit number identifying (together with IP destination address) the security association for this IP datagram
  - SPI value 0 is reserved for local implementation specific use and must not be sent on the wire

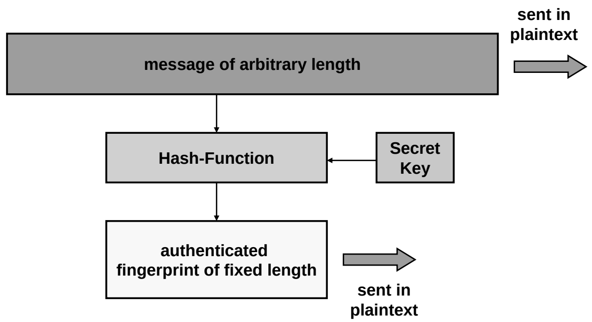
Authentication Header (AH)

- **Sequence number:**
  - monotonically increasing counter value (mandatory and always present)
  - defined in RFC 2085
  - prevention against replay attacks enabled by default
  - mandatory for transmitter but the receiver need not act upon it
  - every new SA resets this number to zero (thus first packet = 1), no cycling: after sending the 2^{32nd} packet, a new SA must be established
  - RFC 4302 allows usage of 64 bit sequence numbers

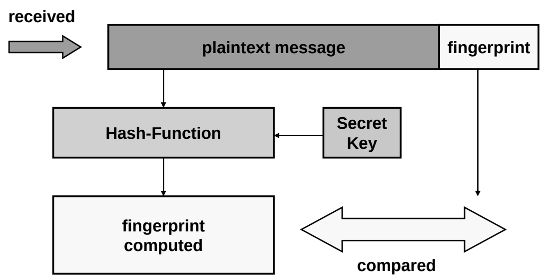
Authentication Header (AH)

- **Authentication Data:**
  - contains Integrity Check Value (ICV)
    - all fields behind AH header plus predictable field of IP header before AH (that means TTL, checksum, ToS/DSCP fields are regarded to be zero for ICV calculation)
  - the algorithm for authentication is free and must be negotiated
  - mandatory default calculation of the authentication data must be supported
    - HMAC with keyed-MD5 (RFC 2403), 128 bit secret-key
    - HMAC with keyed-SHA-1 (RFC 2404), 160 bit secret-key
  - alternative
    - DES-CBC based MAC
  - non-repudiation (IP datagram signing) is not supported!

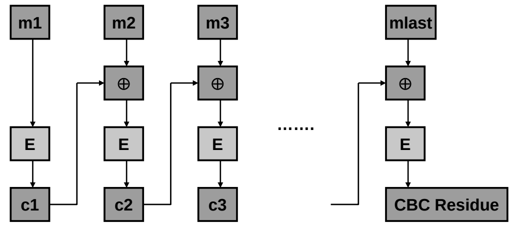
Refresher: HMAC in Action (Sender) 1



Refresher: HMAC in Action (Receiver) 2

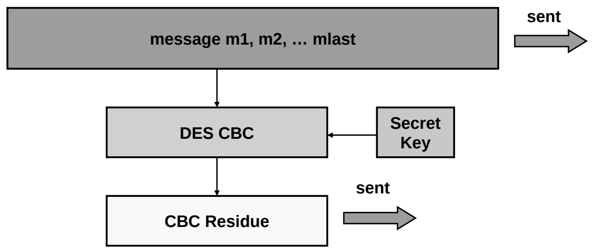


Refresher: MAC CBC Residue 1

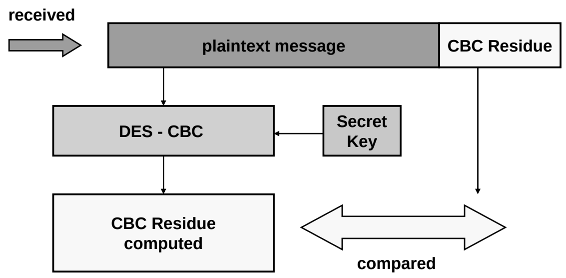


Encryption at sender to create CBC residue
Same done at the receiver to check

Refresher: MAC CBC Residue 2



Refresher: MAC CBC Residue 3



Agenda

- **Overview**
- **AH Protocol**
- **<u>ESP Protocol</u>**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
- **IPsec / IKEv1 Problems**

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

ESP Security Service (RFC 2406, 4303)

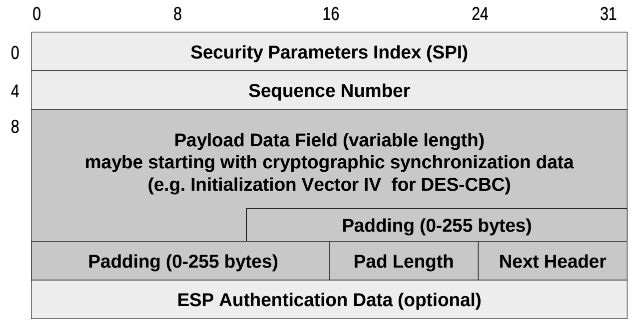
- **ESP provides**
  – confidentiality (encryption of payload with secret-key algorithm)
  – replay detection and protection via sequence number (optional)
  – IP datagram sender authentication by HMAC (optional)
  – IP datagram integrity assurance by HMAC (optional)
- **ESP does not provide**
  – key distribution
  – encryption of IP fragments
    - therefore IP fragments must be assembled before decryption

IPv4 and ESP

0 4 8 16 31

Vers.=4	HLEN	ToS	Total Length	
Fragment Identifier			Flags	Fragment Offset
TTL		<u>protocol = 50</u>	Header Checksum	
Source Address				
Destination Address				
IP Options				Pad
ESP Header with ESP Parameters Encrypted Data ESP Trailer				

Encapsulating Security Payload (ESP)



note: ESP was originally defined as extension header for IPv6 and later same structure was used also for IPv4

ESP Header, Payload

- **SPI and Sequence Number**
  - used for same functions as in the AH header
    - defining SA and prevention of replay attack
  - this are the only fields of ESP transmitted in cleartext
  - RFC 4303 allows usage of 64 bit sequence numbers
- **Payload Field of ESP is encrypted**
  - actual format depends on encryption method
    - e.g. location of Initialization Vector (IV) for DES-CBC
    - note: in such a case every IP datagram must contain an IV because IP datagram's may arrive out of sequence

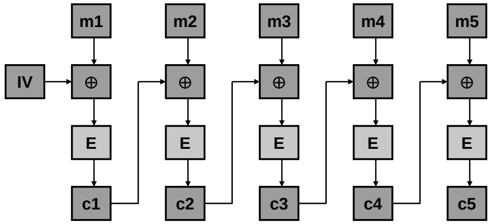
ESP Trailer

- **Padding Field**
  - is used to fill the plaintext to the size required by the encryption algorithm (e.g. the block size of a block cipher)
  - is used to align 4 byte boundaries
- **Pad Length**
  - pointer to end of data
- **Next Header**
  - identifies the type of data contained in the Payload Data Field, e.g., an extension header in IPv6 or an upper layer protocol identifier
    - same values allowed as protocol field in IPv4 header

ESP Encryption Methods

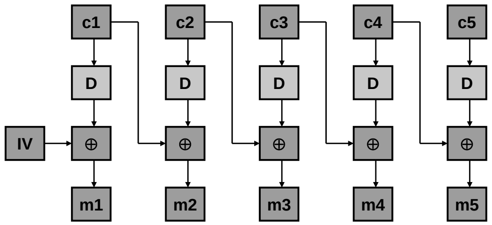
- **Mandatory default transformation of the data**
  - DES-CBC (Data Encryption Standard - Cipher Block Chaining)
    - parameter field contains Initialization Vector (IV) field
- **Triple-DES, Blowfish, IDEA, RC5 and AES as alternative**
  - see RFC 2451
- **An ESP “Null” algorithm must be supported**
  - see RFC 2401
  - see RFC 2410 where it is praised for ease of implementation, great speed and simplicity ;-)
- **Optional authentication**
  - HMAC with keyed-MD5 or HMAC with keyed-SHA-1

Refresher DES Mode - CBC Encryption 1



Encryption with CBC

Refresher DES Mode - CBC Decryption 2



Decryption with CBC

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association (RFC 2401)**
- **Internet Key Exchange Protocol (IKEv1, RFC 2409)**
- **IPsec / IKEv1 Problems**

Requirements

- **Authentication and encryption techniques require that sender and receiver agree on**
  - a key or keys
  - an authentication or encryption algorithm
  - other parameters e.g. lifetime of a key
- **Set of agreements forms**
  - a security association between sender and receiver
- **If a packet is received**
  - it can be verified or decrypted if the receiver can link it with the context of a security association
  - Security Parameter Index (SPI) field of AH or ESP headers is used as such a link
    - value negotiated as part of the key-exchange procedure

Security Association (SA)

- **SA is a fundamental concept of IPsec**
  - prerequisite for any IPsec services
- **SA define the policy used between the peers for certain traffic flows**
  - Traffic (Peers Identity, Access List)
  - Header (AH or ESP)
  - Algorithms (For authentication and encryption)
  - Keys

"For telnet sessions with host A use 3DES with keys K1, K2, and K3 for payload encryption, SHA with key K4 for authentication..."

Security Association (SA)

- **What is a SA in context of IPsec?**
  - it is a cryptographically protected "simplex" connection between two peers affording security services
  - note:
    - for a bidirectional connection between two peers two SA's are necessary, one for each direction
    - for every method (AH or ESP) a separate SA is necessary
- **SA is uniquely identified by the triple**
  - Security Parameter Index (SPI)
  - IP Destination Address of peer
  - Security Protocol Identifier (AH or ESP)
  - note: these are components of the received IP datagram

Institute of Computer Technology - Vienna University of Technology

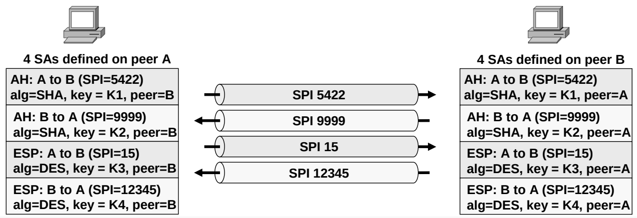
L97 - IPsec

Security Association (SAD)

- **Associated with each end of an SA are**
  - cryptographic parameters
    - algorithms
    - keys, lifetime of keys
  - sequence number counter
  - sequence counter overflow
  - anti-replay window
  - lifetime of the SA
  - mode of the SA (transport or tunnel)
- **These are maintained in binary format in the memory of a peer's computing engine**
  - as the Security Association Database (SAD)

Security Associations Example

- SAs are unidirectional !
  - Thus multiple SAs are typically established between two peers
  - One SA per direction and so called transform (AH or ESP method)
  - Security policies can be totally asymmetric !
  - Identified by the Security Parameter Index (SPI) and peer's IP address
    - SPI is a 32-bit value



Inbound/Outbound Traffic

- **Security Policy Database (SPD)**
  - contains policy for handling of IP datagram's to be sent (outbound) or received (inbound) on an interface
    - given category of IP datagram should be dropped
    - given category of IP datagram should be forwarded without IPsec
    - given category of IP datagram should be forwarded with IPsec
  - categories are identified by selectors (traffic selectors)
    - e.g. Access Control List (ACL)
  - entries for outbound traffic
    - pointer to SAD entry
  - entries for inbound traffic
    - note: for inbound datagram's corresponding SAD entry is found based on IP destination address, SPI and Service Profile Identifier of packet

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Introduction

- **IPsec works if SAs are set up between peers**
  - the algorithms are determined, the session key are established, and so on
- **SA management and key exchange techniques are necessary**
  - either manual distribution
  - or automatic on demand distribution
- **Automatic solution for IPsec**
  - IKE (Internet Key Exchange, RFC 2409)
    - protocol for doing mutual authentication in a secure way and establishing a shared secret in order to create IPsec SAs
    - took many years to come out of IETF

History 1

- **Original contenders for such a protocol**
  - Photuris (RFC 2522)
    - signed anonymous DH
    - stateless cookies
  - SKIP (Simple Key-Management for Internet Protocols)
    - http://skip.incog.com/inet-95.ps
    - perfect forward secrecy
    - avoid long term DH values
- **While fighting**
  - ISAKMP (Internet Security Association and Key Management Protocol, RFC 2408) emerged
    - it is not a protocol
    - it is a framework in which message fields could be exchanged in order to create such a key exchange protocol

History 2

- **IETF decides**
  - that solution for IPsec must be compliant to ISAKMP
- **Photuris and SKIP**
  - were not ISAKMP compliant and hence died
- **Further protocols were developed**
  - Oakley (RFC 2412)
  - SKEME (Secure Key Exchange MEchanism)
  - both were ISAKMP compliant
- **Then IKE appeared**
  - Internet Key Exchange (RFC2409)
  - which combines Oakley and SKEME using ISAKMP syntax

History 3

- **But IKE was incomplete**
  - hence another document was created
  - “The Internet IP Security Domain of Interpretation (DOI) for ISAKMP” (RFC2407)
  - DOI specifies a particular use of ISAKMP
    - like a profile defining which parameters could be chosen
- **In order to implement IKE**
  - you must know RFCs 2407 (DOI), 2408 (ISAKMP) and 2409 (IKE)
  - very confusing, complex, inconsistent and unreadable
    - note IETF will come out with an consolidated RFC for replacement
  - although the world did manage to have interoperable implementations

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IKE General Aspects

- **IKE complexity to guard against a number of attacks:**
  - base for key determination is DH number exchange
    - DH must be protected against man-in-the-middle-attack and therefore a form of an authenticated DH exchange is needed
  - denial of service attack
    - cookies: the messages are constructed with unique cookies that can be used to quickly identify and reject invalid messages without the need to execute processor-intensive cryptographic operations
  - man-in-the-middle attack:
    - protection is provided against the common attacks, such as deletion of messages, modification of messages, reflecting messages back to the sender, replaying of old messages, and redirection of messages to unintended recipients

Refresher: Diffie-Hellman Key Exchange

Alice
Private Value S_A
Public Value P_A

Bob
Private Value S_B
Public Value P_B

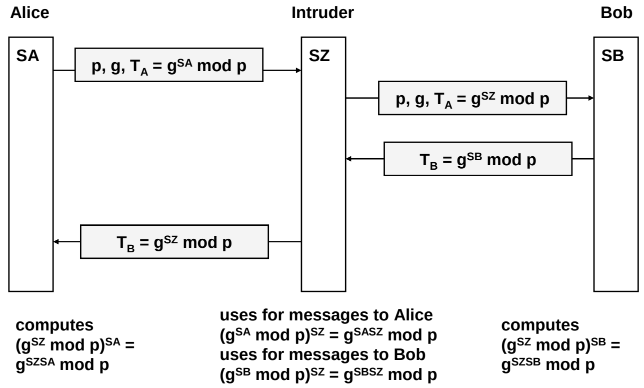
$P_A = g^{S_A} \bmod p$ →

← $P_B = g^{S_B} \bmod p$

Shared secret :

$$P_B^{S_A} \bmod p = g^{S_A S_B} \bmod p = P_A^{S_B} \bmod p$$

Refresher: DH Man-in-the-middle Attack



IKE General Aspects

- **IKE complexity to guard against a number of attacks (cont.):**
  - perfect forward secrecy (PFS):
    - compromise of past keys provides no useful clues for breaking any other key, whether it occurred before or after the compromised key; each refreshed key will be derived without any dependence on predecessor keys
- **Transport of IKE messages**
  - runs on top of UDP
  - port number 500 on both sides
  - starts with ISAKMP header followed by payloads
    - header fields and payload types defined by ISAKMP
    - protocol procedures defined by IKE

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IKE General Aspects

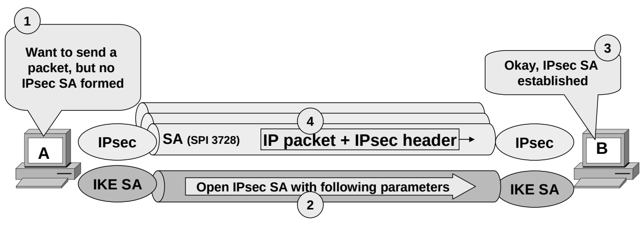
IKE messages

IP	UDP	ISAKMP Header	Payloads	Payloads

- **IKE phase 1**
  - establishing a secure channel -> IKE SA
- **IKE phase 2**
  - establishing requested IPsec SAs on demand
  - protected by IKE SA of phase 1

IKE General Aspects

- **Establishes an authenticated and encrypted tunnel**
  - IKE SA main mode (bidirectional), UDP port 500
- **Creates unidirectional IPsec SAs on demand**
  - Also keys are exchanged



IKE Phases 1

- **Phase 1**
  - does mutual authentication and establishes session keys (initial DH keying material) between two entities
  - authentication is based on identities such as names and secrets such as public-key pairs or pre-shared secrets
  - exchanges are known as ISAKMP SA or IKE SA
  - main mode versus aggressive mode
- **Phase 2**
  - multiple phase 2 SAs between the two entities can be established (e.g. an ESP SA or AH SA)
  - based on session keys established in phase 1 (initial DH keying material)
  - quick mode

IKE Phases 2

- **Phase 1 modes**
  - main mode provides integrity protection
  - aggressive mode uses fewer rounds
- **SA of phase 1 (= IKE SA)**
  - is used to do further negotiations
- **In phase 2**
  - establishment of SA for data communication
  - protected by SA of phase 1 (IKE SA)
- **Perfect Forward Secrecy (PFS)**
  - new DH exchange performed for each new phase 2 SA
  - without PFS DH values of phase 1 are used

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IKE Phase 1 - Modes

- **Main mode**
  - do mutual authentication and establishment of session keys in six messages
  - additional functionality
    - hide endpoint identifiers (usernames) from eavesdropper
    - more flexibility in negotiating of cryptographic parameters
- **Aggressive mode**
  - do mutual authentication and establishment of session keys in three messages

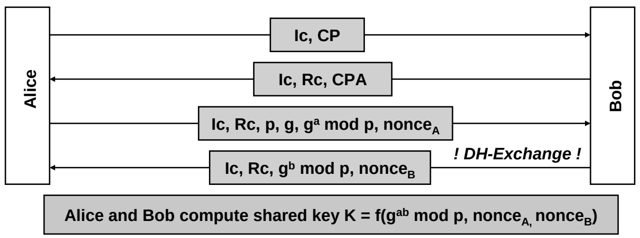
IKE Phase 1 - Keys

- **Key types used for authentication**
  - public signature key
  - public key encryption (original specification)
  - public key encryption (revised specification)
  - pre-shared secret key
  - 8 possibilities for doing phase 1 !!!
    - main and aggressive mode for each of the four
- **Session keys for IKE SA**
  - integrity key
  - encryption key
  - both are secret-keys used for symmetric algorithms

Agenda

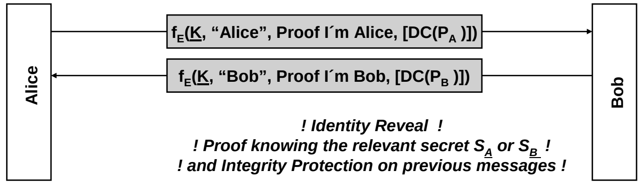
- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **Internet Key Exchange Protocol** **(IKEv1, RFC 2409)**
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Main-Mode Public Signature Keys 1



Ic … Initiator Cookie, Rc … Responder Cookie -> IKE connection Identifier
CP … crypto proposal (encryption, hash, authentication key method, DH group)
CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH values
a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once

Main-Mode Public Signature Keys 2



DS_A= Proof I´am Alice = f_E(S_A , H(Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))
DS_B = Proof I´am Bob = f_E(S_B , H(Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))

DC(P_A) = P_A +f_E(S_{Cert} , P_A) … Digital Certificate of Alice´s Public Key P_A, optional
DC(P_B) = P_B + f_E(S_{Cert} , P_B) … Digital Certificate of Bob´s Public Key P_B , optional
S_{cert} … Private Key of Certificate Authority
S_A … Private Key Alice, S_B … Private Key Bob

Proof ok if f_D(P_A , DS_A) = own Hash of (Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values)
Proof ok if f_D(P_B , DS_B) = own Hash of (Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values)

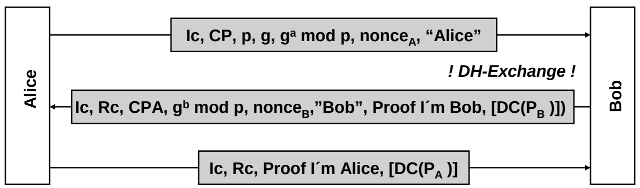
Public Signature Keys - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (nonces, g^{ab} mod p)
    - prf … pseudo random function like a hash function
      - note: SKEYID depends on authentication keys used
  - SKEYID is the seed for all other keys
    - IKE SA and IPsec SAs !!!
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - integrity key SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - encryption key SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Aggressive Mode Public Signature Keys



Ic … Initiator Cookie, Rc … Responder Cookie
CP … crypto proposal, CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH values, a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once
DS_A= Proof I´am Alice = f_E(S_A , H(Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))
DS_B = Proof I´am Bob = f_E(S_B , H(Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))
$DC(P_A) = P_A + f_E(S_{Cert}, P_A)$ … Digital Certificate of Alice´s Public Key P_A
$DC(P_B) = P_B + f_E(S_{Cert}, P_B)$ … Digital Certificate of Bob´s Public Key P_B
S_A … Private Key Alice, S_B … Private Key Bob
Proof ok if $f_D(P_A , DS_A)$ = own Hash of (Alice, nonceA, nonceB, Ic, Rc, CP, public DH values)
Proof ok if $f_D(P_B , DS_B)$ = own Hash of (Bob, nonceA, nonceB, Ic, Rc, CP, public DH values)

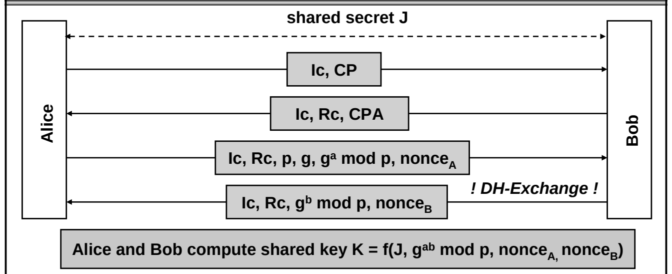
Public Signature Keys - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (nonces, g^{ab} mod p)
    - prf … pseudo random function like a hash function
      - note: SKEYID depends on authentication keys used
  - SKEYID is the seed for all other keys
    - IKE SA and IPsec SAs !!!
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - integrity key SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - encryption key SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

Agenda

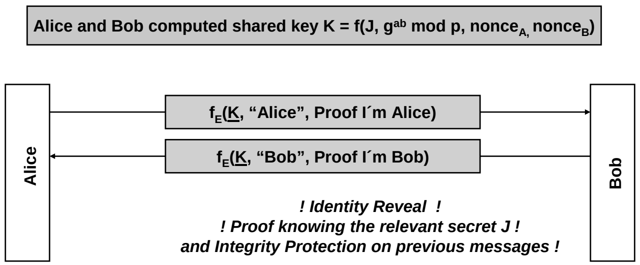
- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Main-Mode Pre-shared Secret-Key 1



Ic … Initiator Cookie, Rc … Responder Cookie -> IKE connection Identifier
CP … crypto proposal (encryption, hash, authentication key method, DH group)
CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH values
a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only <u>once</u>

Main-Mode Pre-shared Secret-Key 2



Proof I´am Alice = f_E(<u>J</u> , H(Alice, nonceA, nonceB, Ic, Rc, CP, public DH values))
Proof I´am Bob = f_E(<u>J</u> , H(Bob, nonceA, nonceB, Ic, Rc, CP, public DH values))

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

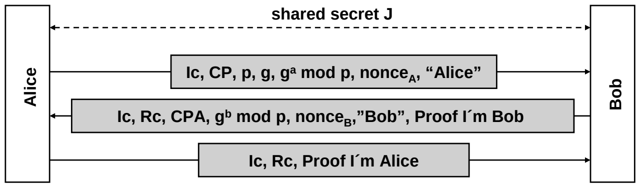
Pre-shared Secret-Key - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (J, nonces)
    - prf … pseudo random function like a hash function
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - <u>integrity key</u> SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - <u>encryption key</u> SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **<u>Internet Key Exchange Protocol</u>** **<u>(IKEv1, RFC 2409)</u>**
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - <u>Pre-shared Keys Aggressive-Mode</u>
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Aggressive Mode Pre-shared Secret-Key



Ic … Initiator Cookie, Rc … Responder Cookie
CP … crypto proposal, CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH value, a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once

Proof I´am Alice = f_E(J , H(Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH value))
Proof I´am Bob = f_E(J , H(Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH value))

Pre-shared Secret-Key - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (J, nonces)
    - prf … pseudo random function like a hash function
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - integrity key SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - encryption key SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

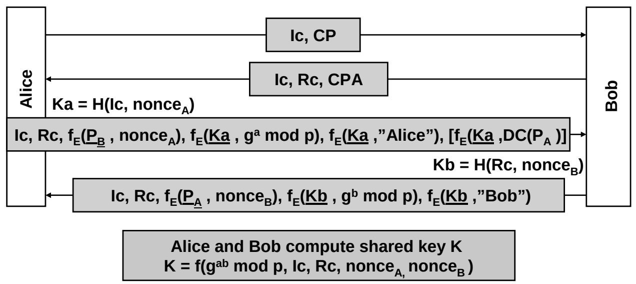
Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

Agenda

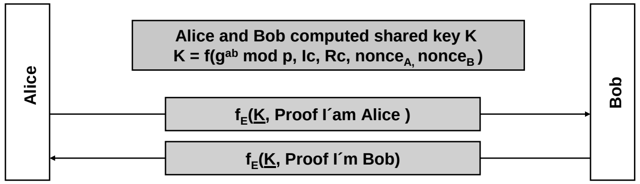
- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Main-Mode Public Encryption Keys (revised) 1



Ic … Initiator Cookie, Rc … Responder Cookie -> IKE connection Identifier
CP … crypto proposal, CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH value, a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once
P_A … Public Key Alice, P_B … Public Key Bob

Main-Mode Public Encryption Keys (revised) 2



Proof I´am Alice = H(Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))
Proof I´am Bob = H(Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values))

Proof ok if f_D(K , Proof I´m Alice) =
own Hash of (Alice, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values)

Proof ok if f_D(K , Proof I´m Bob) =
own Hash of (Bob, $nonce_A$, $nonce_B$, Ic, Rc, CP, public DH values)

Public Encryption Keys - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (hash(nonces), cookies)
    - prf … pseudo random function like a hash function
    - note: SKEYID depends on authentication keys used
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - integrity key SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - encryption key SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

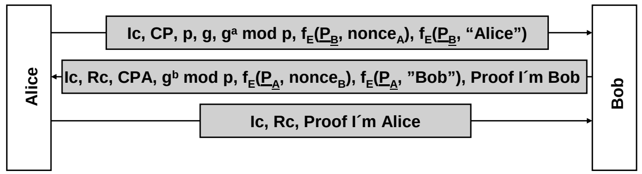
Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **Internet Key Exchange Protocol** **(IKEv1, RFC 2409)**
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

Aggressive Mode Public Encryption Keys (original)



Ic … Initiator Cookie, Rc … Responder Cookie
CP … crypto proposal, CPA … crypto proposal accepted
g^a mod p, g^b mod p … public DH values, a, b … private DH values of Alice, Bob
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once

P_A … Public Key Alice, P_B … Public Key Bob
Proof I´am Alice = H($nonce_B$, Ic, Rc, public DH values)
Proof I´am Bob = H($nonce_A$, Ic, Rc, public DH values)

Proof ok if Proof I´m Alice = own Hash of ($nonce_B$, Ic, Rc, public DH values)
Proof ok if Proof I´m Bob = own Hash of ($nonce_A$, Ic, Rc, public DH values)

Public Encryption Keys - Session Keys for IKE SA

- **Session Key Preparation**
  - SKEYID = prf (hash(nonces), cookies)
    - prf … pseudo random function like a hash function
    - note: SKEYID depends on authentication keys used
  - SKEYID_d = prf (SKEYID, (g^{ab} mod p, cookies, 0)
    - secret bits to create the other keys
- **IKE SA Session Key Building**
  - integrity key SKEYID_a =
    prf (SKEYID, (SKEYID_d (g^{ab} mod p, cookies, 1))
  - encryption key SKEYID_e =
    prf (SKEYID, (SKEYID_a (g^{ab} mod p, cookies, 2))

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

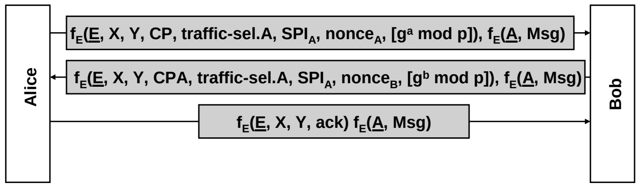
Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

IKE Phase 2 - Quick Mode

- **All messages in phase 2 are**
  - encrypted by encryption key SKEYID_e = E
  - integrity protected by integrity key SKEYID_a = A
  - E and A were built in IKE phase 1
- **Phase 2 exchange**
  - again some new nonces and other information are sent which get shuffled into the SKEYID of phase 1 in order to generate a new pair of encryption and integrity key
    - similar procedure as for IKE-SA but with another start value
  - this generated keys then can be used for a requested IPsec SA (either AH, ESP or ESP/AH)
  - note for next slide:
    - SA is initiated by Alice to open an simplex SA from Alice to Bob
    - Traffic-selector is used by Alice to signal which type of traffic will use this SA towards Bob

IKE Phase 2 - Quick Mode



X = (Ic, Rc), Ic … Initiator Cookie, Rc … Responder Cookie of IKE phase1
Y … distinguisher for initiator, if several setups are performed at the same time
CP … crypto proposal, CPA … crypto proposal accepted
$nonce_A$, $nonce_B$ … quantity which any given user of a protocol uses only once

Msg … is the actual message sent

g^a mod p, g^b mod p … new generated public DH values,
a, b … new generated private DH values of Alice, Bob
(note: this is optional in phase 2; only required if perfect forward secrecy (PFS) is wanted, otherwise DH values of phase 1 are used for generation of key material for IPsec SA)

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** **(RFC 2401)**
- **Internet Key Exchange Protocol** **(IKEv1, RFC 2409)**
  - Introduction
  - Public Signature Keys Main-Mode
  - Public Signature Keys Aggressive-Mode
  - Pre-shared Keys Main-Mode
  - Pre-shared Keys Aggressive-Mode
  - Public Encryption Keys Main-Mode (Revised)
  - Public Encryption Keys Aggressive-Mode (Original)
  - IKE Phase 2
  - ISAKMP / IKE Encoding
- **IPsec / IKEv1 Problems**

IKE Encoding - ISAKMP Message Formats

- **All messages**
  - have a fixed header
    - ISAKMP header
  - and a sequence of so called payloads
- **Fixed header contains**
  - type of next payload
- **Every payload**
  - starts with type of next payload
  - length of this payload
    - last payload element has “type of next payload” set = 0

		fixed	variable	
IP	UDP	ISAKMP Header	Payloads	Payloads

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

Payload Types 1

- **0 = end**
  - no next payload
- **1 = SA (security association)**
  - identifies Domain Of Interpretation (DOI), in our case just simple IP
  - must be followed by type 2 and 3
- **2 = P (proposal)**
  - phase 1: Initiator cookie, Responder cookie
  - phase 2: SPI value
- **3 = T (transform)**
  - cryptographic choices
    - encryption type, hash type, DH group (g, p)

Payload Types 2

- **4 = KE (key exchange)**
  - public DH value
- **5 = ID**
  - phase 1: endpoint identifier (e.g. Alice, Bob)
  - phase 2: traffic descriptor (traffic selector)
- **6 = CERT (certificate)**
- **7 = CR (certificate request)**
- **8 = hash**
- **9 = signature**
- **10 = nonce**
  - random number

Payload Types 3

- **11 = notification**
- **12 = delete**
  - e.g. closing SA (SPI)
- **13 = vendor ID**
- **14 - 127 reserved for future use**
- **128 - 255 reserved for private use**

Fixed Header

number of bytes	
8	initiator´s cookie
8	responder´s cookie
1	next payload
1	version number
1	exchange type
1	flags
4	message ID
4	message length (in bytes)

Fixed Header Fields 1

- **Exchange Type:**
  - 1 = base (not used by IKE)
  - 2 = identity protection = IKE main mode
  - 3 = authentication only (not used by IKE)
  - 4 = aggressive = IKE aggressive mode
  - 5 = informational
    - single message e.g. informing other side about a problem (refusal because of wrong version number)
  - 6 - 31 = reserved values for future ISAKMP exchange types
  - 32 - 239 defined within a particular DOI
  - 240 - 255 for private use

Fixed Header Fields 2

- **Flags:**
  - bit 0: if set then following payloads are encrypted
  - bit 1: commit (confusing naming)
    - ISAKMP -> receiver should wait until sender issues a "I am ready" message (Bob tells Alice to wait for Bob´s Ack)
    - IKE -> receiver is requested to acknowledge this message (Bob tells Alice to send an ACK)
  - bit 2: authentication only, if set then following payloads are not encrypted
    - only set in phase 2 to specify a message is in cleartext
- **Message ID:**
  - used in phase 2 to tie together related packets

Payload Type SA, P and T 1

- **The SA payload for IKE contains the P (proposal) and T (transform) payloads**
- **T must be carried within P, P must be carried within SA**
  - e.g. payloads for 2 Proposals with 4 and 2 Transforms will look

    SA P T T T T P T T
- **P indicates also the protocol to be negotiated**
  - phase 1 IKE
  - phase 2 AH
  - phase 2 ESP
  - IP compression (PCP)

Payload Type SA, P and T 2

- **Usage of P**
  - in phase 1 -> only one proposal -> phase 1 IKE
  - in phase 2 -> could be several proposals
    - e.g. AH only, ESP only, AH+ESP, or any of these plus IP compression
- **T indicates a complete suite of cryptographic algorithms/parameters**
  - in phase 1 you need 4 (5) algorithms/parameters
    - authentication type
    - hash type
    - encryption type
    - DH group (g/p or elliptic curve)
    - optional lifetime of IKE SA (default is 8 hours)

Payload Type SA, P and T 3

- **Usage of T**
  - Transform ID contains the number for a method defined within DOI
  - numbers defined in RFC 2407
  - for IKE
    - Key-IKE (Oakley)
  - for AH
    - MD5, SHA, DES (CBC residue)
  - for ESP
    - DES, 3DES
    - RC4, RC5
    - IDEA, 3IDEA
    - CAST, Blowfish
    - Null

Transforms Overview

AH	ESP Encryption	ESP Auth.	PCP
AH-MD5	ESP-NULL	ESP-MD5	PCP-LZS
AH-SHA	ESP-DES	ESP-SHA	
...	ESP-3DES	...	
	ESP-IDEA		
	...		

Payload Compression Protocol (PCP)

- **Problem: encrypted datagram's cannot be compressed efficiently**
  - encryption introduces randomness
- **PCP reduces IP datagram size *before* encryption**
  - hence must be a component of IPsec
- **Increases the overall communication performance**
  - RFC 2393

Agenda

- **Overview**
- **AH Protocol**
- **ESP Protocol**
- **Security Association** (RFC 2401)
- **Internet Key Exchange Protocol** (IKEv1, RFC 2409)
- **<u>IPsec / IKEv1 Problems</u>**

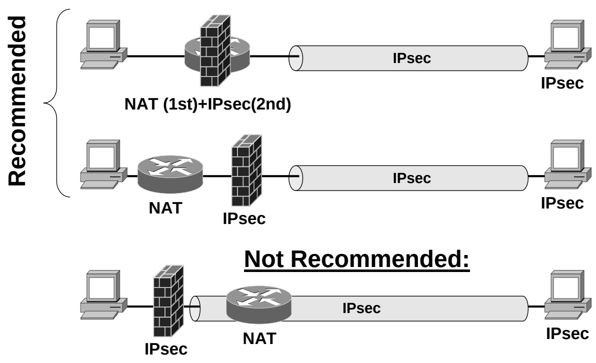
NAT and IPsec AH, ESP

- **AH hash includes the whole IP header**
  - Cannot work together with NAT
- **ESP**
  - Transport mode: authentication excludes IP but not TCP/UDP header for hash calculation!
    - but TCP checksum includes the Pseudo-IP header
    - therefore turn off TCP checksum verification in the receiver
  - Tunnel mode: Outer IP header is neither encrypted nor authenticated – no problems with NAT
  - note:
    - N(P)AT (NAT with port address translation) will modify TCP port numbers
    - if TCP + Payload is ESP encrypted that is not possible
    - propriety Cisco solution -> encapsulate ESP in UDP or TCP
- **See RFC 3715 NAT Traversal for ongoing work**

NAT and IPsec IKE

- **Internet Key Exchange (IKE)**
  - Problem if exchanged keys or certificates are bound to gateway's IP address
  - avoid it by using other identifier of the endpoint e.g. User-ID or FQDN
- **Expiration of Security Association (SA)**
  - Re-key request is sent to the initial UDP port 500
  - Problems with multiple security gateways behind a N(P)AT device
- **See RFC 4306 (IKEv2) for ongoing work**

Apply NAT before IPsec !



Problems with IPsec / IKEv1 1

- **IPsec for Site-to-Site VPN**
  - Often uses pre-shared secrets for authentication of IKE peers
  - Why?
    - certificates means maintaining a PKI (Public Key Infrastructure)
      - at least a private CA (Certification Authority) server is needed
    - VPN router/concentrator can often be physically protected
- **IPsec for Client-to-Site VPN**
  - Different situation
    - Mobile PCs calling from insecure places
    - Pres-hared secret may be compromised hence configuration and maintenance overhead if number of clients is high
  - Therefore combination of IPsec, well-known RAS Authentication Techniques (PPP with EAP, RFC 3748) and X-AUTH
    - Client dials-in, authenticates itself at a authentication server (VPN concentrator) and then the necessary IPsec configuration is pushed from the VPN concentrator to the client
      - sometimes even enhanced with activation of a host based FW function at the client side of IPsec
      - Client gets an IP address from the VPN concentrator and all client traffic may be forced to go exclusively to the VPN concentrator
    - solved with X-AUTH exchange as add-on to IKEv1
      - X-AUTH exchange is an inherent optional part of IKEv2
  - IPsec Tunnel mode is used

Institute of Computer Technology - Vienna University of Technology

L97 - IPsec

Problems with IPsec / IKEv1 2

- **IPsec for End-to-End VPN (Client-to-Client VPN in transport mode**
  - Some inherent problems when using certificates caused by the fact that certificates normally deals with names but IPsec (especially the SPD) knows only about IP addresses
  - Binding of certificates to IP addresses not possible or not wanted
- **See the following documents for more information about some basic problems of IPsec and IKEv1:**
  - **"A Cryptographic Evaluation of IPsec"**
    - **Niels Ferguson and Bruce Schneier**
    - **-> http://www.schneier.com/paper-ipsec.pdf**
  - **"Experiences with Host-to-Host IPsec"**
    - **Tuomas Aura, Michael Roe, and Anish Mohammed**
    - **http://research.microsoft.com/users/tuomaura/Publications/aura-roe-mohammed-protocols05.pdf**
  - **"Key Exchange in IPSec: Analysis of IKE"**
    - **Charlie Kaufman, Radia Perlman**
    - **IEEE Internet Computing Volume 4 Issue 6 (2000) page50-56**
    - http://ieeexplore.ieee.org/iel5/4236/19367/00895016.pdf?isnumber=19367&prod=JNL&arnumber=895016&arSt=50&ared=56&arAuthor=Perlman%2C+R.%3B+Kaufman%2C+C.
  - **"Analysis of the IPsec Key Exchange"**
    - **Charlie Kaufman, Radia Perlman**
    - http://krypt1.cs.uni-sb.de/teaching/seminars/ws2001/literature/PerKau2001.pdf
- **This leads to an adaptation of the original IPsec RFCs (IPsec Architecture, AH and ESP Headers) and to an completely rework of the IKE protocol -> IKEv2**