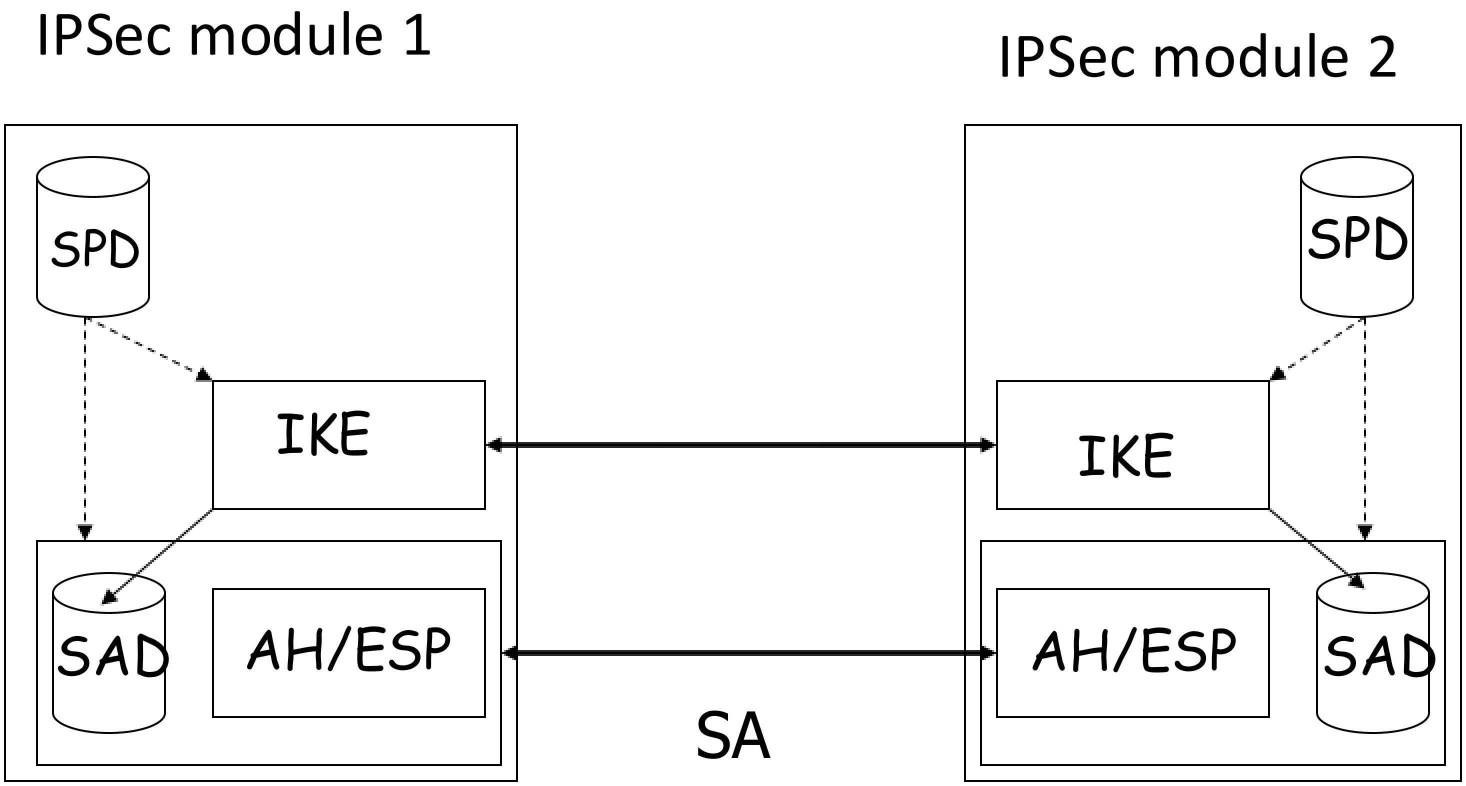
IKEv2: IPSec Key Management Protocol

IP Security Architecture

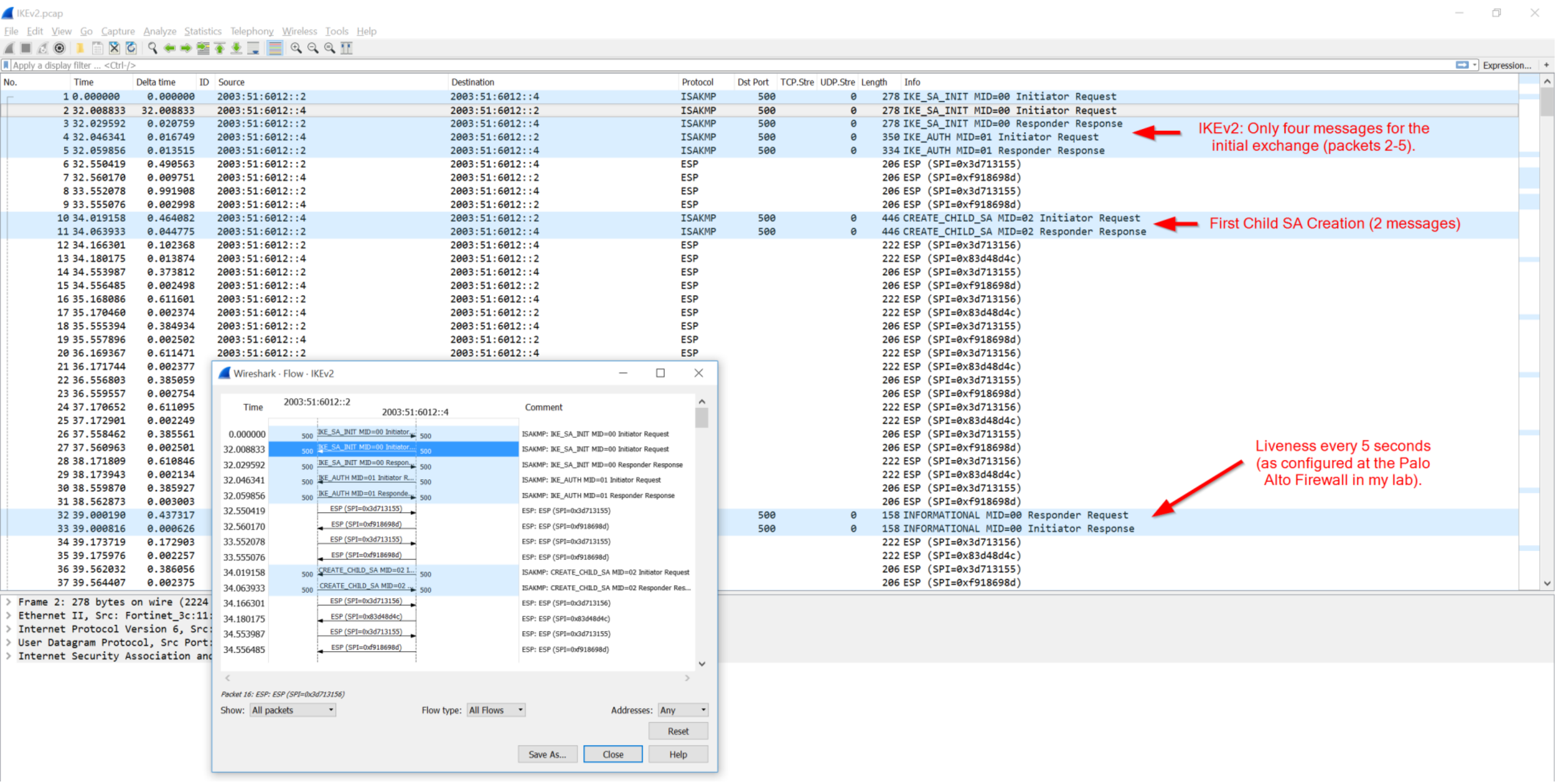


SAD: Security Association Database

SPD: Security Policy Database

IKE: Internet Key Exchange

Wireshark capture



No.	Time	Delta time	ID	Source	Destination	Protocol	Dst Port	TCP.Stre	UDP.Stre	Length	Info
1	0.000000	0.000000		2003:51:6012::2	2003:51:6012::4	ISAKMP	500		0	278	IKE_SA_INIT MID=00 Initiator Request
2	32.008833	32.008833		2003:51:6012::4	2003:51:6012::2	ISAKMP	500		0	278	IKE_SA_INIT MID=00 Initiator Request
3	32.029592	0.020759		2003:51:6012::2	2003:51:6012::4	ISAKMP	500		0	278	IKE_SA_INIT MID=00 Responder Response
4	32.046341	0.016749		2003:51:6012::4	2003:51:6012::2	ISAKMP	500		0	350	IKE_AUTH MID=01 Initiator Request
5	32.059856	0.013515		2003:51:6012::2	2003:51:6012::4	ISAKMP	500		0	334	IKE_AUTH MID=01 Responder Response
6	32.550419	0.490563		2003:51:6012::2	2003:51:6012::4	ESP				206	ESP (SPI=0x3d713155)
7	32.560170	0.009751		2003:51:6012::4	2003:51:6012::2	ESP				206	ESP (SPI=0xf918698d)
8	33.552078	0.991908		2003:51:6012::2	2003:51:6012::4	ESP				206	ESP (SPI=0x3d713155)
9	33.555076	0.002998		2003:51:6012::4	2003:51:6012::2	ESP				206	ESP (SPI=0xf918698d)
10	34.019158	0.464082		2003:51:6012::4	2003:51:6012::2	ISAKMP	500		0	446	CREATE_CHILD_SA MID=02 Initiator Request
11	34.063933	0.044775		2003:51:6012::2	2003:51:6012::4	ISAKMP	500		0	446	CREATE_CHILD_SA MID=02 Responder Response
12	34.166301	0.102368		2003:51:6012::2	2003:51:6012::4	ESP				222	ESP (SPI=0x3d713156)
13	34.180175	0.013874		2003:51:6012::4	2003:51:6012::2	ESP				222	ESP (SPI=0x83d48d4c)
14	34.553987	0.373812		2003:51:6012::2	2003:51:6012::4	ESP				206	ESP (SPI=0x3d713155)
15	34.556485	0.002498		2003:51:6012::4	2003:51:6012::2	ESP				206	ESP (SPI=0xf918698d)
16	35.168086	0.611601		2003:51:6012::2	2003:51:6012::4	ESP				222	ESP (SPI=0x3d713156)
17	35.170460	0.002374		2003:51:6012::4	2003:51:6012::2	ESP				222	ESP (SPI=0x83d48d4c)
18	35.555394	0.384934		2003:51:6012::2	2003:51:6012::4	ESP				206	ESP (SPI=0x3d713155)
19	35.557896	0.002502		2003:51:6012::4	2003:51:6012::2	ESP				206	ESP (SPI=0xf918698d)
20	36.169367	0.611471		2003:51:6012::2	2003:51:6012::4	ESP				222	ESP (SPI=0x3d713156)
21	36.171744	0.002377								222	ESP (SPI=0x83d48d4c)
22	36.556803	0.385059								206	ESP (SPI=0x3d713155)
23	36.559557	0.002754								206	ESP (SPI=0xf918698d)
24	37.170652	0.611095								222	ESP (SPI=0x3d713156)
25	37.172901	0.002249								222	ESP (SPI=0x83d48d4c)
26	37.558462	0.385561								206	ESP (SPI=0x3d713155)
27	37.560963	0.002501								206	ESP (SPI=0xf918698d)
28	38.171809	0.610846								222	ESP (SPI=0x3d713156)
29	38.173943	0.002134								222	ESP (SPI=0x83d48d4c)
30	38.559870	0.385927								206	ESP (SPI=0x3d713155)
31	38.562873	0.003003								206	ESP (SPI=0xf918698d)
32	39.000190	0.437317					500		0	158	INFORMATIONAL MID=00 Responder Request
33	39.000816	0.000626					500		0	158	INFORMATIONAL MID=00 Initiator Response
34	39.173719	0.172903								222	ESP (SPI=0x3d713156)
35	39.175976	0.002257								222	ESP (SPI=0x83d48d4c)
36	39.562032	0.386056								206	ESP (SPI=0x3d713155)
37	39.564407	0.002375								206	ESP (SPI=0xf918698d)

Outline

- Motivations of Automated Key Management
- Key Concepts
  - Diffie-Hellman Key Exchange Protocol
  - Perfect Forward Secrecy
  - Pseudo-Random Function (PRF)
- IKEv2
  - Authentication and Key Generation
  - Cryptographic Algorithm Negotiation
  - Re-keying
- Some Comments on IKEv2

Why Automated Key Management?

- AH & ESP need keys.
- Manual Techniques
  - They are the simplest.
  - They are practical only in a small and static environment.
  - They need the human intervention and can easily lead to mis-configurations.
  - They do not scale well.
  - Static key is not good for security.

Revision: Any problem about DH?

Diffie-Hellman Key Exchange Protocol

User A

Generate random $X_A < p$

calculate $Y_A = \alpha^{X_A} \bmod p$

Calculate $k = (Y_B)^{X_A} \bmod p$

$\xrightarrow{Y_A}$

$\xleftarrow{Y_B}$

User B

Generate random $X_B < p$

Calculate $Y_B = \alpha^{X_B} \bmod p$

Calculate $k = (Y_A)^{X_B} \bmod p$

Diffie-Hellman in Practice

- Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
  - 768-bit modulus and primitive root 2.
  - 1024-bit modulus and primitive root 2.
  - Two “elliptic curve” DH parameters (details omitted here)
  - 1536-bit MODP Group
  - 2048-bit MODP Group
  - 3072-bit MODP Group
  - 4096-bit MODP Group
  - 6144-bit MODP Group
  - 8192-bit MODP Group

Perfect Forward Secrecy (PFS)

- By perfect forward secrecy we mean that the compromise of a single session key will not compromise other session keys.
- To this end, any key should not be derived from any predecessor key.

Pseudo-Random Function (PRF)

- A PRF function takes a variable length key, variable length data and produces a fixed length output.
  - e.g., a keyed hash function h_k obtained from a hash function h with the HMAC method.
- A PRF is used in IKE for generating keying materials and authentication (details will be given later).
- In RFC4307: Recommended PRF
  - PRF_HMAC_SHA1 MUST RFC2104
  - PRF_HMAC_MD5 MAY RFC2104
  - PRF_AES128_CBC SHOULD+ AES-PRF
- Technical details of these PRFs are omitted here.

PRF+

prf+ (K,S) = T1 , T2 , T3 , T4 , ...

where the blocks of strings:

T1 = prf (K, S | 0x01)

T2 = prf (K, T1 | S | 0x02)

T3 = prf (K, T2 | S | 0x03)

T4 = prf (K, T3 | S | 0x04)

...

where

| means concatenation

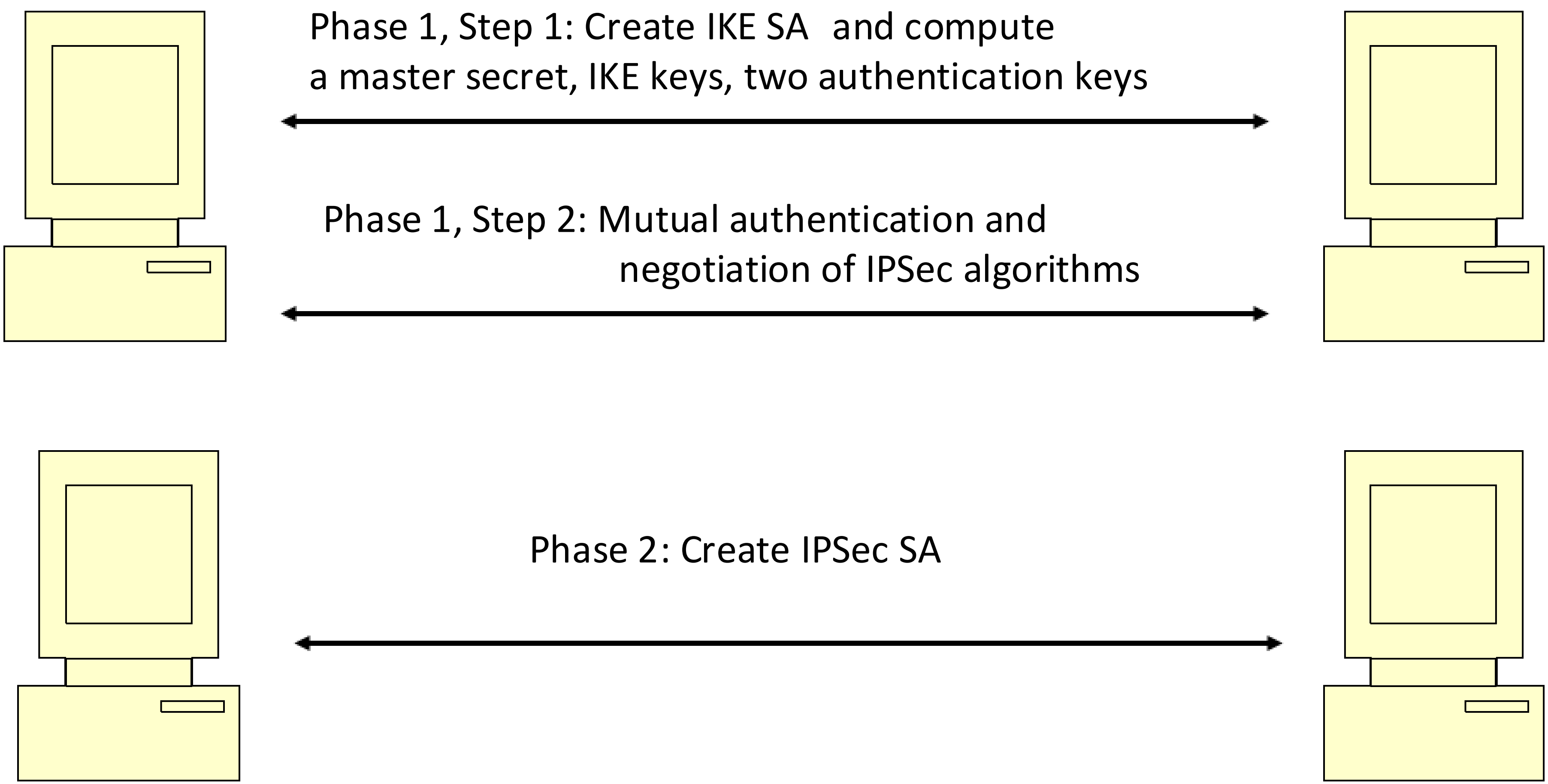
0x01 etc. are constants

A number of Ti's are computed iteratively as needed

Outline

- Motivations of Key Management
- Key Concepts
  - Diffie-Hellman Key Exchange Protocol
  - Perfect Forward Secrecy
- IKEv2
  - Authentication and Key Generation
  - Cryptographic Algorithm Negotiation
  - Re-keying
- Some Comments on IKEv2

IKEv2 Outline



IKEv2 Protocol

Phase 1, Step 1: IKE_SA_INIT

- Negotiate IKE algorithms (Ciphers, Hash algorithms, DH group)
- Compute four secret keys for IKE
- Compute master secret k_d for computing IPSec keys in Phase 2.
- Compute two mutual authentication keys for Step 2 of Phase 1 below.

Phase 1, Step 2: IKE_AUTH

- Mutual authentications (two choices)
- Negotiation of IPsec algorithms (piggybacked here)

Phase 2: CREATE_CHILD_SA

- Setup IPSec security associations

Phase 1.1: IKE_SA_INIT (1)

Initiator — **Responder**

HDR, SAi1, KEi, Ni →

← HDR, SAr1, KEr, Nr, [CERTREQ]

- HDR (IKE header)
  - Version number
  - SPIi: A value chosen by the initiator to identify this IKE security association.
  - ……
- SAi1
  - Supported crypto algorithms of initiator for the IKE_SA (DH group, encrpt, authen algor for protecting the messages in Phase 1.2 and Phase 2, prf)
- KEx
  - Diffie-Hellman values

- Nx
  - Nonce of Init./Responder
  - Used for authentication & computing secret keys
- SAr1
  - Expressed choice based on SAi1
- [CERTREQ]
  - Optional request that decides a mutual authentication method

Phase 1.1: IKE_SA_INIT (2)

- After exchanging two messages, each party can generate SKEYSEED based on the values in KEi and KEr by DH
  - SKEYSEED=prf(Ni | Nr, g^(s_is_r)) [Remark: s_i the secret of I]

    Nonces add the freshness to the key materials.
  - {SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr} =

    prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)

    The prefix of output of the function prf+ is cut into pieces as different keys
- SK_d is the master secret that will be used to compute IPSec SA keys later in Phase 2.
- Messages in Phase 1.2 and Phase 2 will be integrity protected and encrypted by SK_ai, SK_ei, SK_ar, SK_er, respect.
- SK_pi and SK_pr are pre-shared secret keys for authentication in Phase 1.2 (technical details of this authentication method are given later).

Phase 1.2: IKE_AUTH (1)

Initiator **Responder**

HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTHi, SAi2, TSi, TSr} →

← HDR, SK {IDr, [CERT,] AUTHr, SAr2, TSi, TSr}

- {…}
  - indicated payloads are encrypted and integrity protected using that direction's SK_e & SK_a and the IKE encryption and auth algorithms
- IDi, IDr
  - For authentication by preshared secrets (SK_pi, SK_pr) (details given later)
- AUTHx
  - Preshared secrets (SK_pi, SK_pr) (details given later)
  - Digital signature (details given later)
- SAi2/SAr2 piggybacked here
  - For CREATE_CHILD_SA
  - They contain only algorithms
- TSx
  - Traffic Selector Info (IP Add + Port)
  - It defines which traffic to be protected by SAi2, SAr2
  - It contains protocol, port range, address range
  - TSi = (0, 0-65535,192.0.2.202-192.0.2.202)
  - TSr = (0, 0-65535,192.0.2.0-192.0.2.255)

The Whole Picture of Phase 1

Initiator → **Responder**

HDR, SAi1, KEi, Ni →

← HDR, SAr1, KEr, Nr, [CERTREQ]

HDR, SK{IDi, [CERT,] [CERTREQ,] [IDr,] AUTHi, SAi2, TSi, TSr} →

← HDR, SK{IDr, [CERT,] AUTHr, SAr2,TSi, TSr}

Remark 1: [CERTREQ] means authentication with digital certificate.
Remark 2: “SK{}” means encryption using the keys sk_{ei} and sk_{er} .
Remark 2: SAi2 and SAr2 are negotiations of IPSec SA algorithms, piggybacked in this authentication step.

Mutual Authent. by AUTH (2)

- Digital Signature Based
  - Requires individual [CERT] exist in both messages
  - [CERTREQ] indicates to use certificate authentication
  - Initiator signs the 1st message appended by Nr and prf(SK_pi, IDi)
  - responder signs the 2nd message appended by Ni and prf(SK_pr, IDr)
- Pre-shared Key (SK_pi, SK_pr)
  - Authenticators AUTHx use the negotiated prf function
  - AUTHx = prf(prf(Shared Secret, "Key Pad for IKEv2"), <InitiatorSignedOctets> or <ResponderSignedOctets>)
  - "InitiatorSignedOctets" involves: 1st message in Phase 1.1, Nr, IDi, prf(SK_pi, IDi)
  - "ResponderSignedOctets" is similar.

CHILD_SA Negotiations in IKE_AUTH

- Establishment of CHILD_SA is piggybacked in IKE_AUTH
- Initiator proposes SAi2 in message 3
- Responder answers SAr2 in message 4
- Traffic protected by the SA is also negotiated through traffic selectors (TSi, TSr)

Phase 2: CREATE_CHILD_SA

Initiator | Responder

HDR, SK { [N], [SAi], Ni, [KEi], [TSi, TSr] } →

← HDR, SK { [SAr], Nr, [KEr], [TSi, TSr] }

- [N]: Indication negotiation of new IPSec SA
- [KEx]
  - Diffie-Hellman value, different from those in Phase 1.1
  - Used only when PFS is required. In this case, they will be used in computing new IPSec keys
- [TSx]
  - Traffic Selector Negotiations for new IPSec SA
  - Used only when [N] is used
- If [N] is not used, this is the 1st IPSec SA creation under this IKE SA
- The protection SK{} here is by the IKE SA negotiated before.
- Ni and Nr should be different from those in Phase 1.1. They and SK_d are used to compute IPSec secretr keys.

- An established IKE SA may be used to create many IPSec SAs and may be used for a long time.
- A set of IPSec algorithms was already negotiated in Phase 1.2.

  However, if a new IPSec SA should be created, then [N] is used to indicate this. At the same time, new [KEi] and [TSi, TSr] (different from those in Phase 1.2) may be negotiated.
- The Ni and Nr here are different from those in Phase 1.1, and will be used to compute IPSec secret keys.

Finally, Keys for AH or ESP

- After CREATE_CHILD_SA, the key(s) for AH or ESP will be generated!
- KEYMAT = prf+(SK_d, Ni | Nr)
  - Ni and Nr are the new nonces in Phase 2
  - They are independent of the two nonces in Phase 1
  - KEYMAT is cut into pieces as AH and/or ESP keys
- For stronger PFS
  - KEYMAT = prf+(SK_d, g^(s_i s_r) (new) | Ni | Nr) ,
  - Where g^s_i and g^s_r are the new DH values in Phase 2, SK_d is the old one Phase 1, Ni and Nr are new ones in Phase 2.
  - KEYMAT is cut into pieces as AH and/or ESP keys

1010110101010100010101101010111010101010101010100010010101010100011010101110101010100010110

GOBIERNO DE ESPAÑA MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

Outline

- Motivations of Key Management
- Key Concepts
  - Diffie-Hellman Key Exchange Protocol
  - Perfect Forward Secrecy
  - Pseudo-Random Function (PRF)
- IKEv2
  - Authentication and Key Generation
  - Cryptographic Algorithm Negotiation
  - Re-keying
- Improvements over IKE (v1)
- Some Comments on IKEv2

Cryptographic Algorithm Negotiation

- "SA" payload consists of one or more proposals:
  - IPSec protocols: IKE, ESP, AH
  - Cryptographic algorithms associated with each protocol
  - A prf function may be included
- The responder answers this choice based on the proposals proposed by the Initiator

Re-keying

- Secret keys of IKE, ESP and AH should be only used in a limited amount of time.
- After SA lifetime expires, re-keying must be done.
- Either side thinks that an SA has been used for enough time, it negotiates a new SA.
- After the new SA is setup, delete the old one.

Outline

- Motivations of Key Management
- Key Concepts
  - Diffie-Hellman Key Exchange Protocol
  - Perfect Forward Secrecy
- IKEv2
  - Authentication and Key Generation
  - Cryptographic Algorithm Negotiation
  - Re-keying
- Comments about IKEv2

Some Comments on IKEv2

- It's debatable to keep the Phase I & II architecture
- Still over-flexible in terms of
  - Optional choice of PFS in CREATE_CHILD_SA
- A revised version of IKEv2 was leased in 2014 and is available in: https://tools.ietf.org/html/rfc7296
  - It is now a standard.
- A "minimal" version of March 2016 can be found in:

https://datatracker.ietf.org/doc/rfc7815/

GOBIERNO DE ESPAÑA MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

References

- Bellovin, S., "COMS W4180 Session 11 IP Sec", http://www.cs.columbia.edu/~smb/classes/f06/l10.pdf
- Lee, Kyesang., "Internet Key Exchange version 2 (IKEv2) protocol", http://seclab.cs.ucdavis.edu/-seminars/IKEv2.ppt
- Paterson, K., "A Cryptographic Tour of the IPsec Standards", http://eprint.iacr.org/2006/097.pdf
- Perlman, R., Kaufman, C., "Key exchange in IPSec: analysis of IKE", IEEE Internet Computing, Vol. 4 Issue: 6, Nov.-Dec. 2000, pp. 50 -56.
- Harkins, Kaufman, Kivinen, Kent, Perlman, "Design Rationale for IKEv2", www3.ietf.org/proceedings/02jul/I-D/draft-ietf-ipsec-ikev2-rationale-00.txt

GOBIERNO DE ESPAÑA MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL