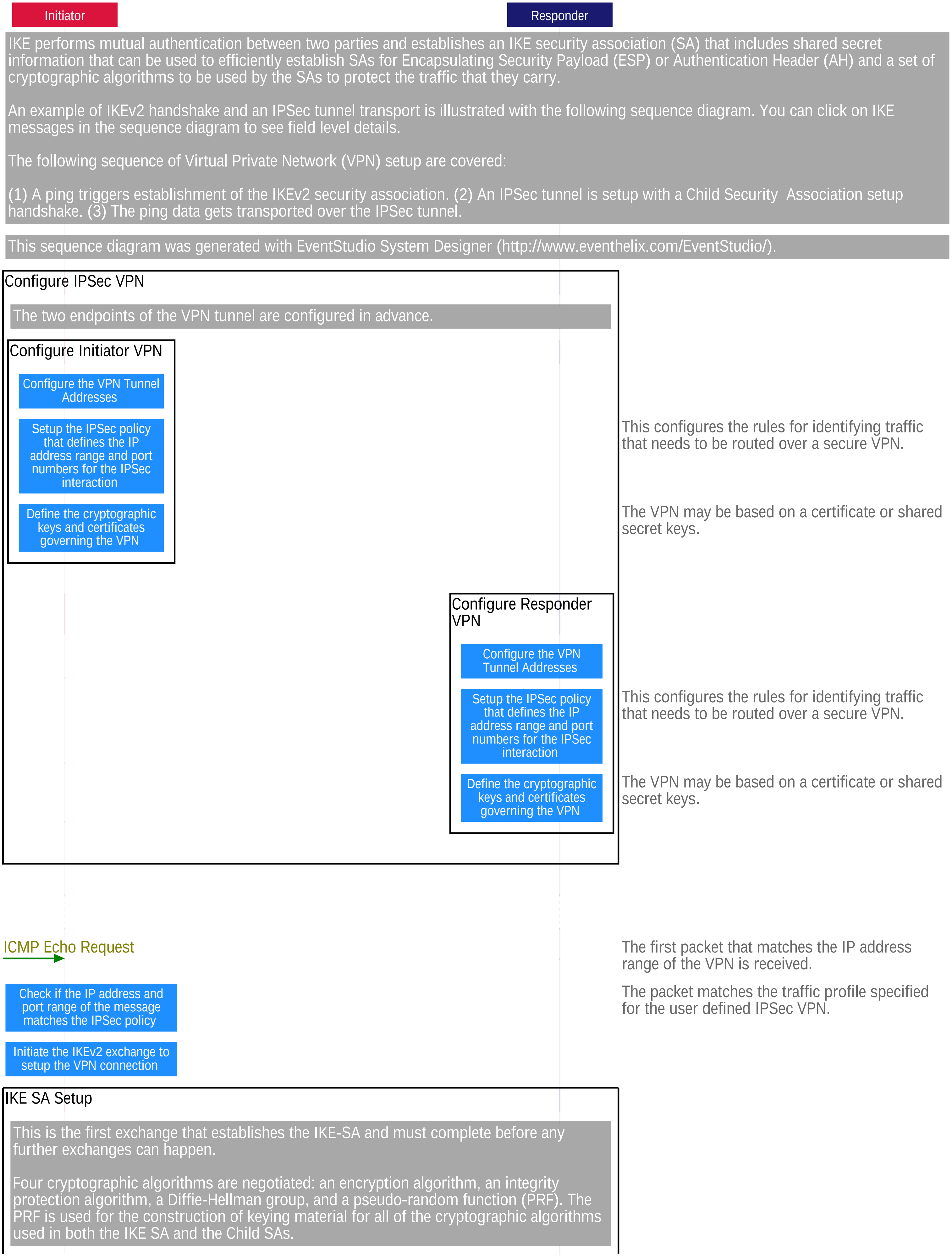


Initiator

Responder

IKE performs mutual authentication between two parties and establishes an IKE security association (SA) that includes shared secret information that can be used to efficiently establish SAs for Encapsulating Security Payload (ESP) or Authentication Header (AH) and a set of cryptographic algorithms to be used by the SAs to protect the traffic that they carry.

An example of IKEv2 handshake and an IPSec tunnel transport is illustrated with the following sequence diagram. You can click on IKE messages in the sequence diagram to see field level details.

The following sequence of Virtual Private Network (VPN) setup are covered:

(1) A ping triggers establishment of the IKEv2 security association. (2) An IPSec tunnel is setup with a Child Security Association setup handshake. (3) The ping data gets transported over the IPSec tunnel.

This sequence diagram was generated with EventStudio System Designer (http://www.eventhelix.com/EventStudio/).

Configure IPSec VPN

The two endpoints of the VPN tunnel are configured in advance.

Configure Initiator VPN

Configure the VPN Tunnel Addresses

Setup the IPSec policy that defines the IP address range and port numbers for the IPSec interaction

This configures the rules for identifying traffic that needs to be routed over a secure VPN.

Define the cryptographic keys and certificates governing the VPN

The VPN may be based on a certificate or shared secret keys.

Configure Responder VPN

Configure the VPN Tunnel Addresses

Setup the IPSec policy that defines the IP address range and port numbers for the IPSec interaction

This configures the rules for identifying traffic that needs to be routed over a secure VPN.

Define the cryptographic keys and certificates governing the VPN

The VPN may be based on a certificate or shared secret keys.

ICMP Echo Request

The first packet that matches the IP address range of the VPN is received.

Check if the IP address and port range of the message matches the IPSec policy

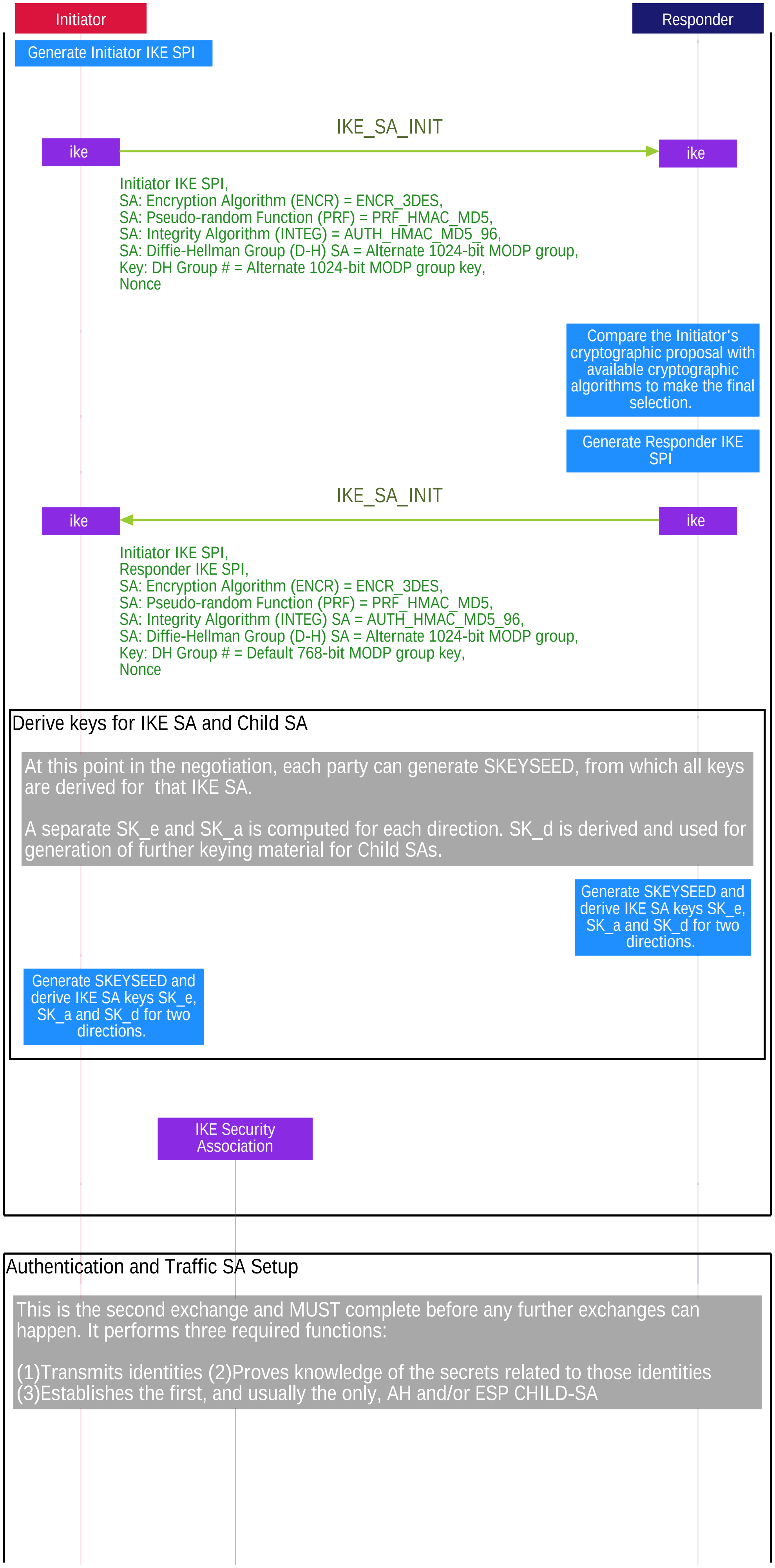
The packet matches the traffic profile specified for the user defined IPSec VPN.

Initiate the IKEv2 exchange to setup the VPN connection

IKE SA Setup

This is the first exchange that establishes the IKE-SA and must complete before any further exchanges can happen.

Four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudo-random function (PRF). The PRF is used for the construction of keying material for all of the cryptographic algorithms used in both the IKE SA and the Child SAs.



Initiator

Responder

Generate Initiator IKE SPI

IKE SPI (aka cookie) is an 8 type pseudo random number generated as md5{src ip, dest ip, random #, time}

IKE_SA_INIT

ike

ike

Initiator IKE SPI,
SA: Encryption Algorithm (ENCR) = ENCR_3DES,
SA: Pseudo-random Function (PRF) = PRF_HMAC_MD5,
SA: Integrity Algorithm (INTEG) = AUTH_HMAC_MD5_96,
SA: Diffie-Hellman Group (D-H) SA = Alternate 1024-bit MODP group,
Key: DH Group # = Alternate 1024-bit MODP group key,
Nonce

The initiator sends the initial cryptographic proposal for the IKE SA. This includes sending the supported encryption algorithm (ENCR), pseudo random algorithm (PRF) and integrity algorithm (INTEG). The Diffie-Hellman (DH) group are also included. The DH public key is also included in the initial exchange.

Compare the Initiator's cryptographic proposal with available cryptographic algorithms to make the final selection.

The Responder selects the IKE SA proposal.

Generate Responder IKE SPI

Generate the 8 byte IKE SPI (cookie).

IKE_SA_INIT

ike

ike

The Responder replies back to the Initiator with the selected cryptographic proposal.

Initiator IKE SPI,
Responder IKE SPI,
SA: Encryption Algorithm (ENCR) = ENCR_3DES,
SA: Pseudo-random Function (PRF) = PRF_HMAC_MD5,
SA: Integrity Algorithm (INTEG) SA = AUTH_HMAC_MD5_96,
SA: Diffie-Hellman Group (D-H) SA = Alternate 1024-bit MODP group,
Key: DH Group # = Default 768-bit MODP group key,
Nonce

Derive keys for IKE SA and Child SA

At this point in the negotiation, each party can generate SKEYSEED, from which all keys are derived for that IKE SA.

A separate SK_e and SK_a is computed for each direction. SK_d is derived and used for generation of further keying material for Child SAs.

Generate SKEYSEED and derive IKE SA keys SK_e, SK_a and SK_d for two directions.

Generate SKEYSEED and derive IKE SA keys SK_e, SK_a and SK_d for two directions.

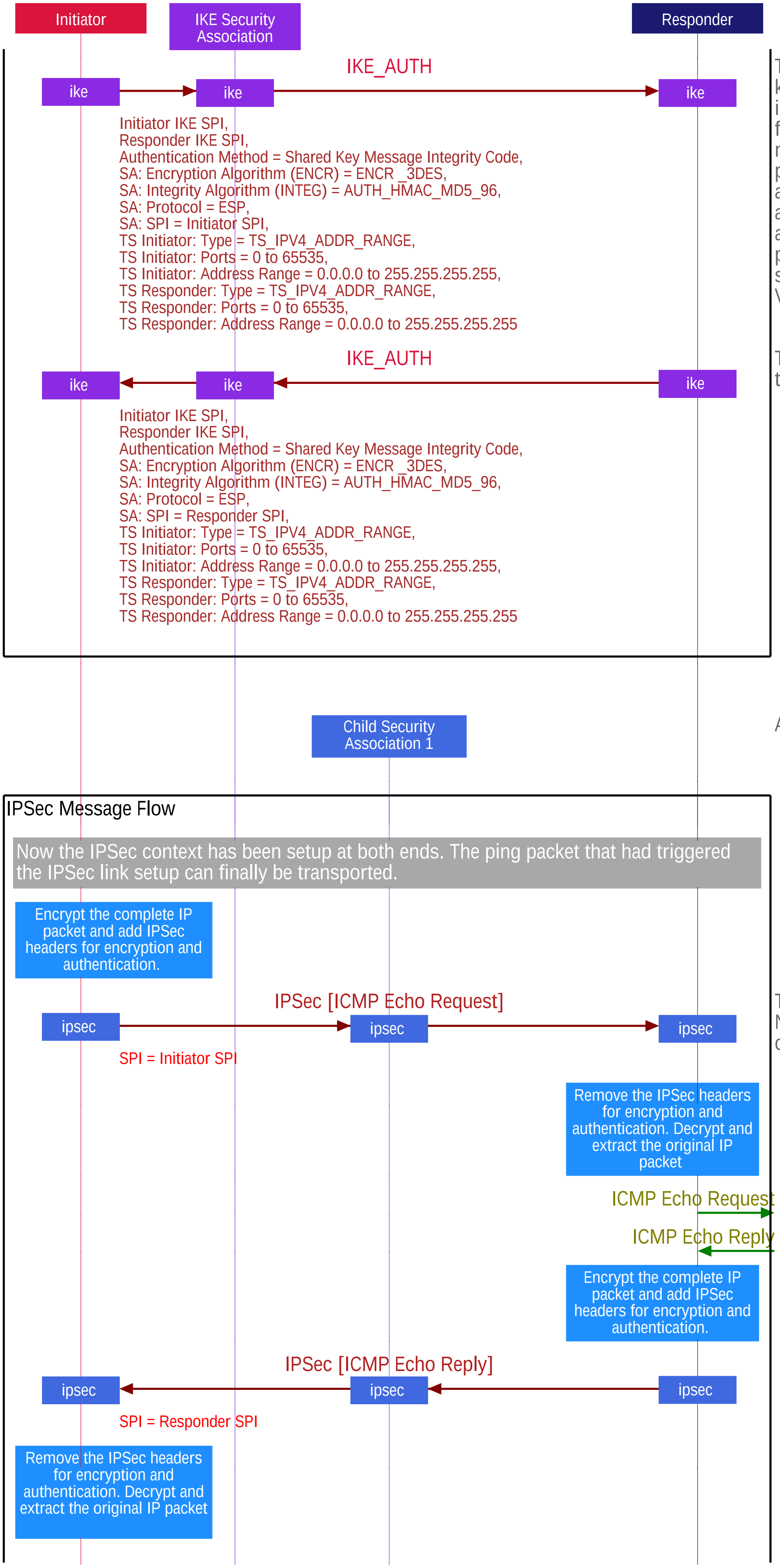
IKE Security Association

At this point, an IKE security association is active between the Initiator and the Responder. All IKE messages will be transferred using this association.

Authentication and Traffic SA Setup

This is the second exchange and MUST complete before any further exchanges can happen. It performs three required functions:

(1)Transmits identities (2)Proves knowledge of the secrets related to those identities (3)Establishes the first, and usually the only, AH and/or ESP CHILD-SA



Initiator

IKE Security Association

Responder

IKE_AUTH

ike → ike → ike

Initiator IKE SPI,
Responder IKE SPI,
Authentication Method = Shared Key Message Integrity Code,
SA: Encryption Algorithm (ENCR) = ENCR _3DES,
SA: Integrity Algorithm (INTEG) = AUTH_HMAC_MD5_96,
SA: Protocol = ESP,
SA: SPI = Initiator SPI,
TS Initiator: Type = TS_IPV4_ADDR_RANGE,
TS Initiator: Ports = 0 to 65535,
TS Initiator: Address Range = 0.0.0.0 to 255.255.255.255,
TS Responder: Type = TS_IPV4_ADDR_RANGE,
TS Responder: Ports = 0 to 65535,
TS Responder: Address Range = 0.0.0.0 to 255.255.255.255

The initiator asserts its identity, proves knowledge of the secret corresponding to the identity and integrity protects the contents of the first message using the AUTH payload. The message also proposes the cryptographic parameters for IPSec Child SA. The encryption algorithm and the integrity protection algorithm are proposed. A SPI (Security Parameter Index) is associated with the selected IPSec cryptographic parameters. The traffic selectors in the message specify the IP addresses and ports for the IPSec VPN connection.

IKE_AUTH

ike ← ike ← ike

Initiator IKE SPI,
Responder IKE SPI,
Authentication Method = Shared Key Message Integrity Code,
SA: Encryption Algorithm (ENCR) = ENCR _3DES,
SA: Integrity Algorithm (INTEG) = AUTH_HMAC_MD5_96,
SA: Protocol = ESP,
SA: SPI = Responder SPI,
TS Initiator: Type = TS_IPV4_ADDR_RANGE,
TS Initiator: Ports = 0 to 65535,
TS Initiator: Address Range = 0.0.0.0 to 255.255.255.255,
TS Responder: Type = TS_IPV4_ADDR_RANGE,
TS Responder: Ports = 0 to 65535,
TS Responder: Address Range = 0.0.0.0 to 255.255.255.255

The reply to the IKE_AUTH message completes the IPSec ESP cryptographic handshake.

Child Security Association 1

At this point, the IPSec Child SA has been setup.

IPSec Message Flow

Now the IPSec context has been setup at both ends. The ping packet that had triggered the IPSec link setup can finally be transported.

Encrypt the complete IP packet and add IPSec headers for encryption and authentication.

IPSec [ICMP Echo Request]

ipsec → ipsec → ipsec

SPI = Initiator SPI

This packet is encrypted and integrity protected. No eavesdropper can decipher the packets contents or modify the packet without detection.

Remove the IPSec headers for encryption and authentication. Decrypt and extract the original IP packet

ICMP Echo Request

ICMP Echo Reply

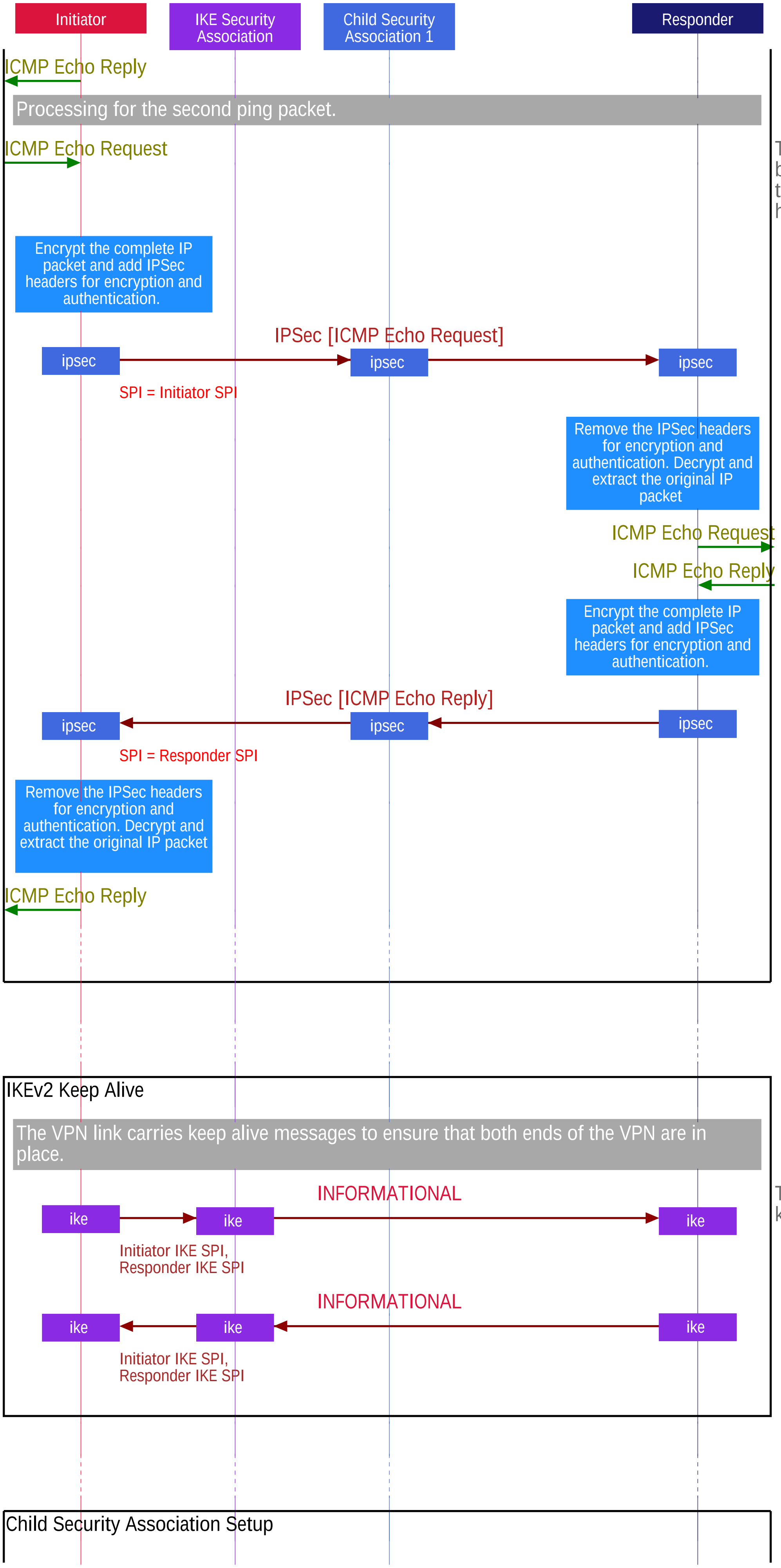
Encrypt the complete IP packet and add IPSec headers for encryption and authentication.

IPSec [ICMP Echo Reply]

ipsec ← ipsec ← ipsec

SPI = Responder SPI

Remove the IPSec headers for encryption and authentication. Decrypt and extract the original IP packet



Initiator | IKE Security Association | Child Security Association 1 | Responder

ICMP Echo Reply

Processing for the second ping packet.

ICMP Echo Request

The IPSec link is already active, so the packet can be directly encrypted and sent. (Remember that the first ping packet had triggered the IKE handshake and IPSec link establishment.

Encrypt the complete IP packet and add IPSec headers for encryption and authentication.

IPSec [ICMP Echo Request]

ipsec → ipsec → ipsec

SPI = Initiator SPI

Remove the IPSec headers for encryption and authentication. Decrypt and extract the original IP packet

ICMP Echo Request

ICMP Echo Reply

Encrypt the complete IP packet and add IPSec headers for encryption and authentication.

IPSec [ICMP Echo Reply]

ipsec ← ipsec ← ipsec

SPI = Responder SPI

Remove the IPSec headers for encryption and authentication. Decrypt and extract the original IP packet

ICMP Echo Reply

IKEv2 Keep Alive

The VPN link carries keep alive messages to ensure that both ends of the VPN are in place.

INFORMATIONAL

ike → ike → ike

Initiator IKE SPI, Responder IKE SPI

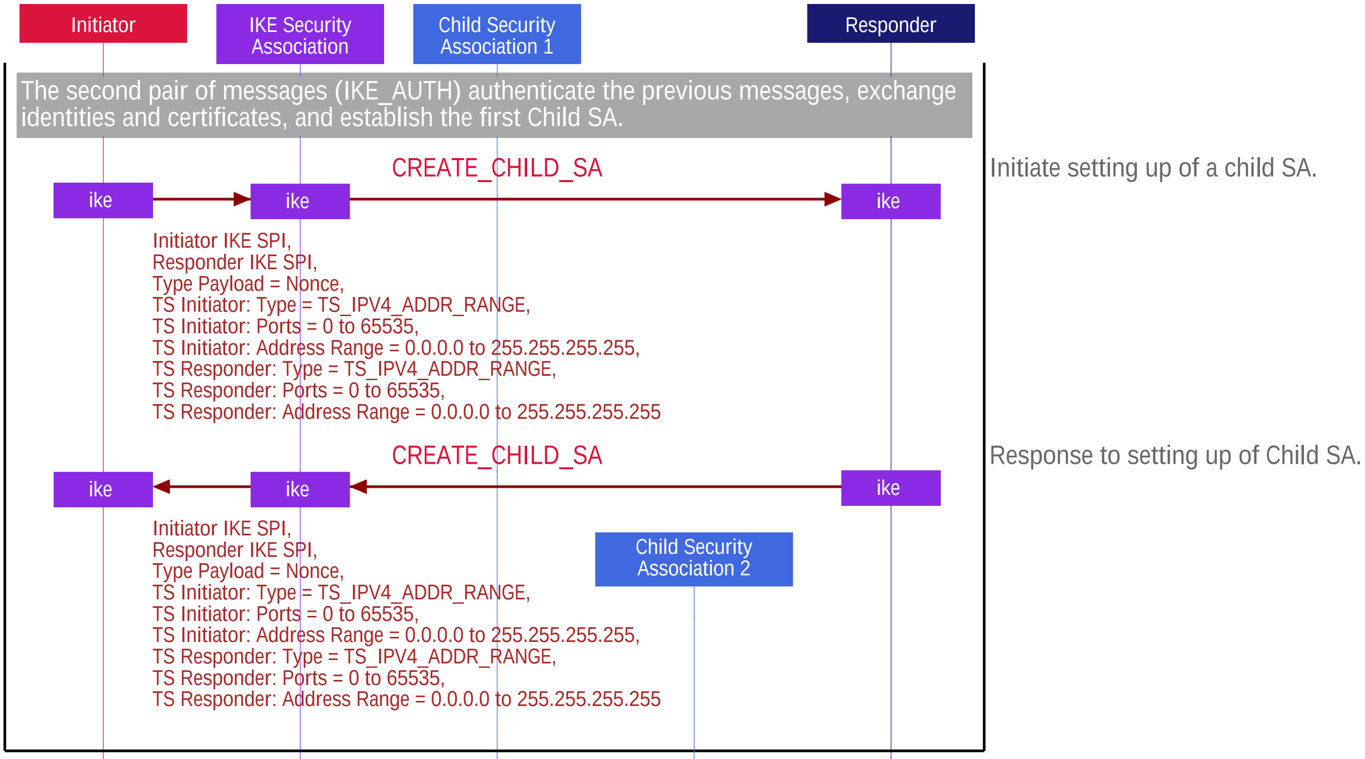
The INFORMATIONAL IKE message is used for keep alive.

INFORMATIONAL

ike ← ike ← ike

Initiator IKE SPI, Responder IKE SPI

Child Security Association Setup



The second pair of messages (IKE_AUTH) authenticate the previous messages, exchange identities and certificates, and establish the first Child SA.

CREATE_CHILD_SA

Initiate setting up of a child SA.

Initiator IKE SPI,
Responder IKE SPI,
Type Payload = Nonce,
TS Initiator: Type = TS_IPV4_ADDR_RANGE,
TS Initiator: Ports = 0 to 65535,
TS Initiator: Address Range = 0.0.0.0 to 255.255.255.255,
TS Responder: Type = TS_IPV4_ADDR_RANGE,
TS Responder: Ports = 0 to 65535,
TS Responder: Address Range = 0.0.0.0 to 255.255.255.255

CREATE_CHILD_SA

Response to setting up of Child SA.

Initiator IKE SPI,
Responder IKE SPI,
Type Payload = Nonce,
TS Initiator: Type = TS_IPV4_ADDR_RANGE,
TS Initiator: Ports = 0 to 65535,
TS Initiator: Address Range = 0.0.0.0 to 255.255.255.255,
TS Responder: Type = TS_IPV4_ADDR_RANGE,
TS Responder: Ports = 0 to 65535,
TS Responder: Address Range = 0.0.0.0 to 255.255.255.255

This diagram was generated with EventStudio System Designer (http://www.eventhelix.com/EventStudio/).

Explore more TCP/IP sequence diagrams at: http://www.eventhelix.com/RealtimeMantra/Networking/