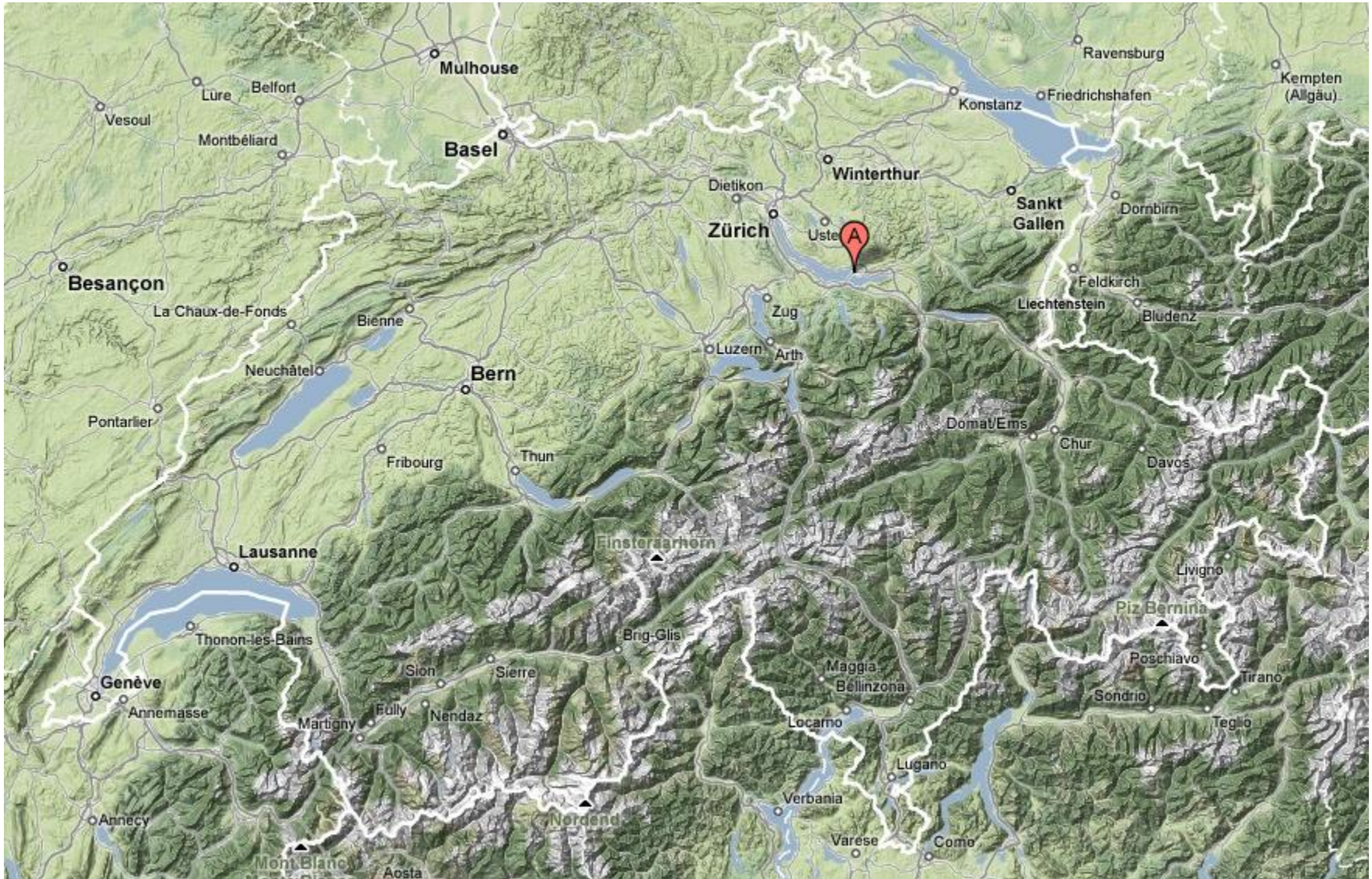


# IKEv2-based VPNs using strongSwan

Prof. Dr. Andreas Steffen

[andreas.steffen@strongswan.org](mailto:andreas.steffen@strongswan.org)

# Where the heck is Rapperswil?



- University of Applied Sciences with about 1000 students
- Faculty of Information Technology (300-400 students)
- Bachelor Course (3 years), Master Course (+1.5 years)



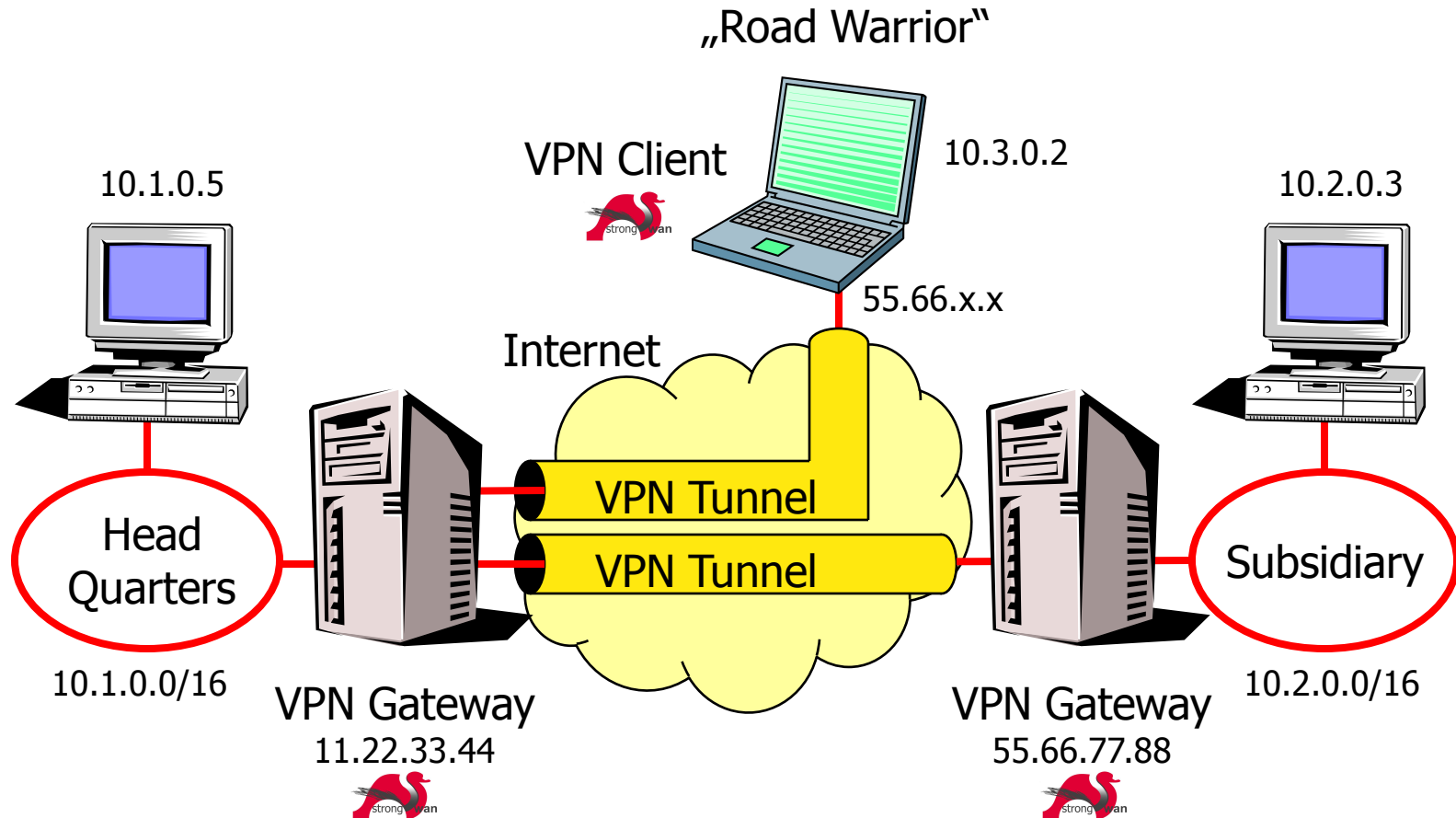
# Agenda

- What is strongSwan?
- IKEv1 versus IKEv2
- A Simple Remote Access Example
- Virtual IP Pools
- Certificate Revocation Mechanisms
- The NETKEY IPsec Stack of the Linux 2.6 Kernel
- Interaction with the Linux Netfilter Firewall
- Dead Peer Detection (DPD)
- Remote Access with Mixed Authentication
- Interoperability with the Windows 7 Agile VPN Client
- The strongSwan NetworkManager Plugin
- EAP-Radius based Authentication
- The strongSwan Architecture
- Cryptographic Plugins
- High Availability using Cluster IP
- IKEv2 Mediation Extension

# What is strongSwan?

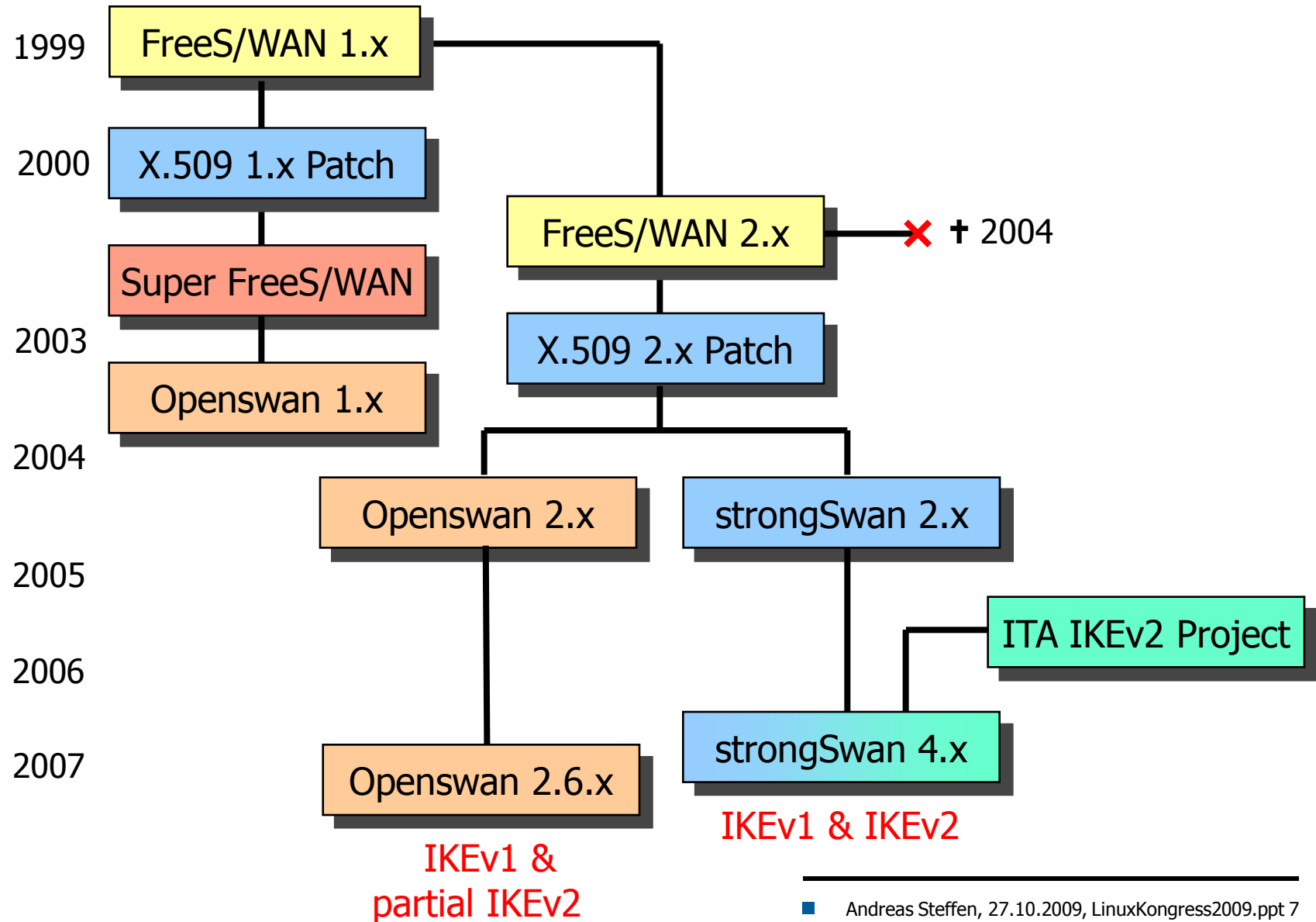


# VPN Usage Scenarios

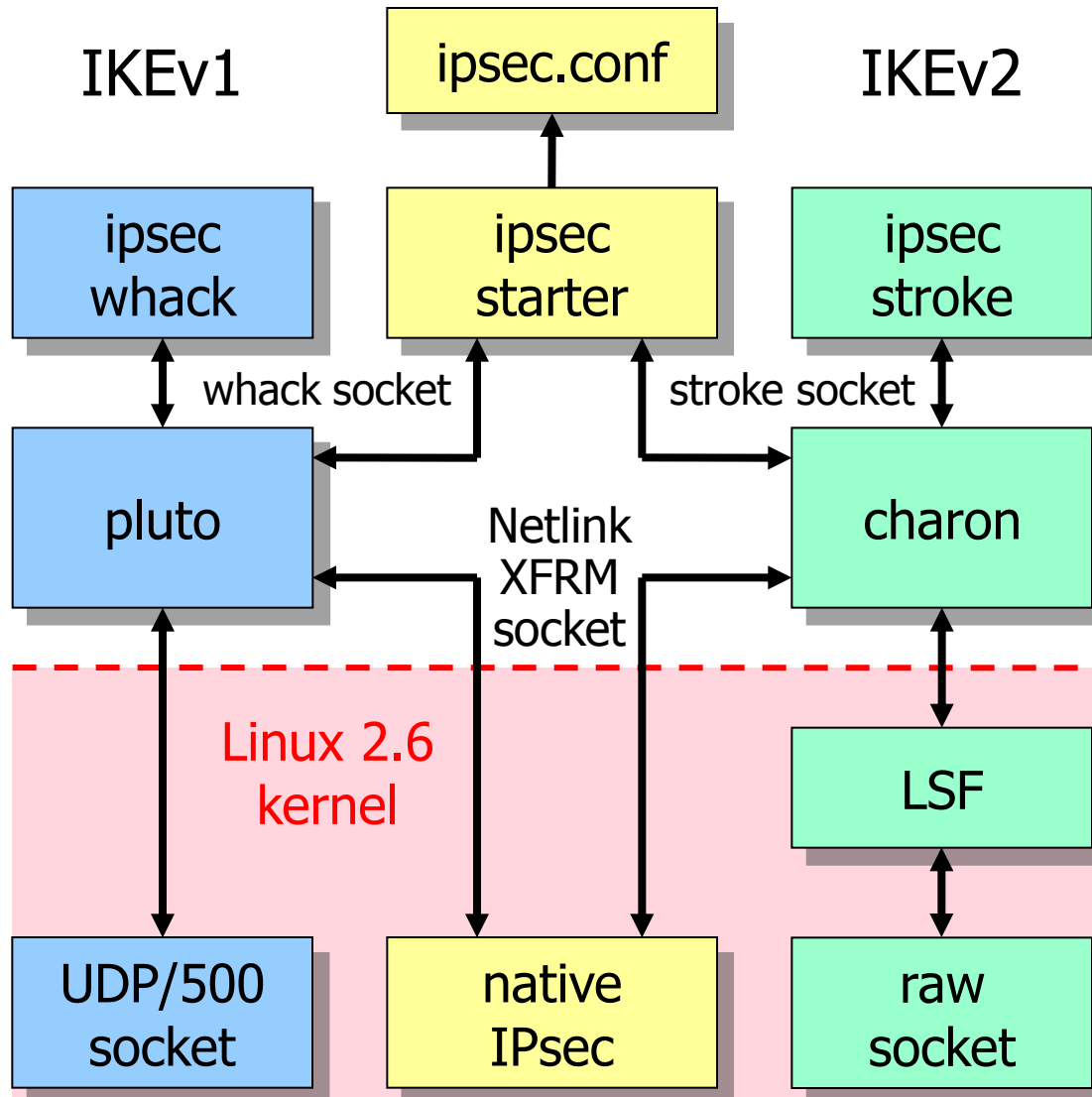


- strongSwan is an Internet Key Exchange daemon needed to automatically set up IPsec-based VPN connections.

# The FreeS/WAN Genealogy



# The strongSwan IKE Daemons



- IKEv1
  - 6 messages for IKE SA  
Phase 1 Main Mode
  - 3 messages for IPsec SA  
Phase 2 Quick Mode
- IKEv2
  - 4 messages for IKE SA and first IPsec SA  
IKE\_SA\_INIT/IKE\_AUTH
  - 2 messages for each additional IPsec SA  
CREATE\_CHILD\_SA



# IKEv2 Interoperability Workshops

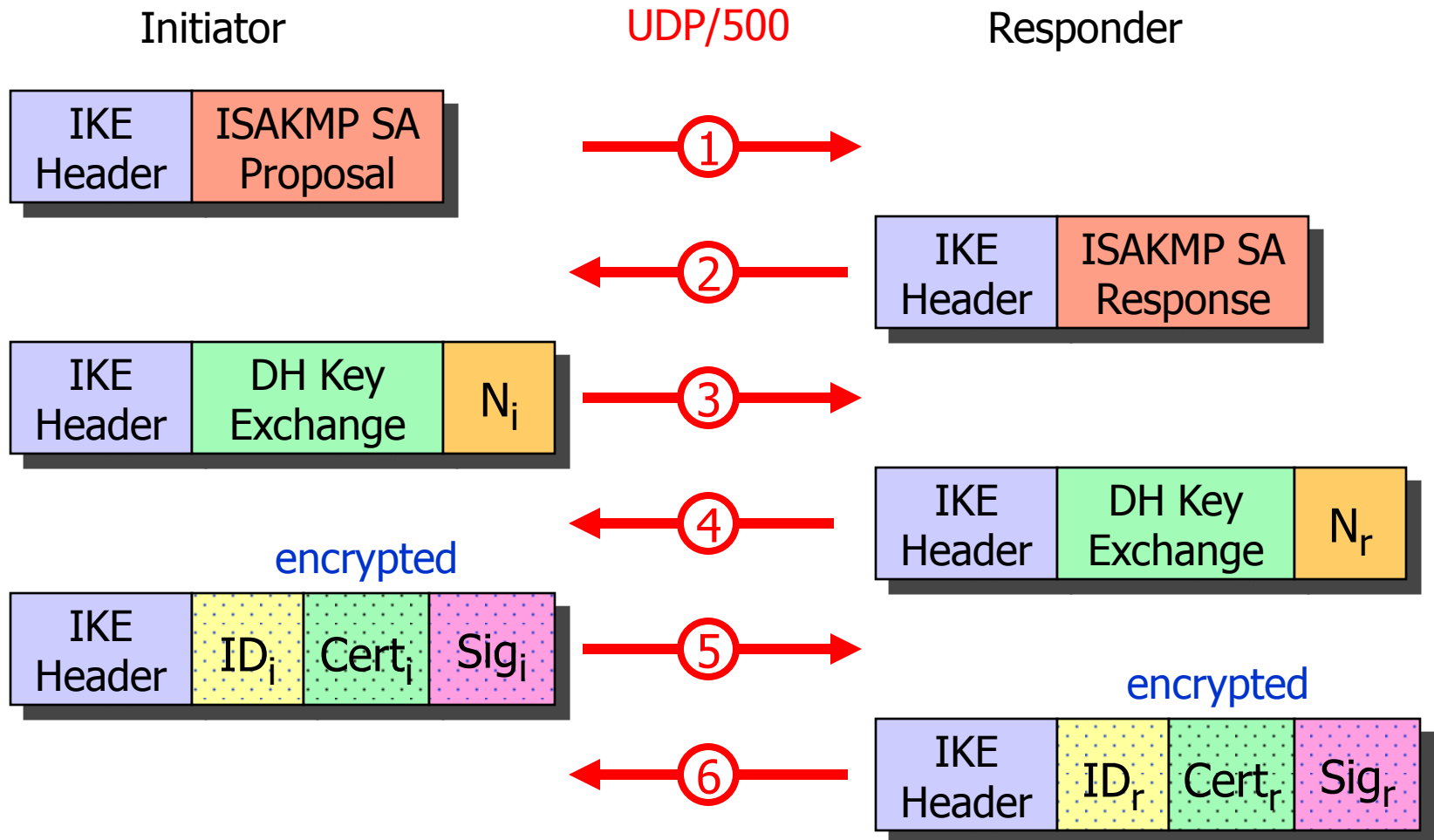


Spring 2007 in Orlando, Florida  
Spring 2008 in San Antonio, Texas

- **strongSwan** successfully interoperated with IKEv2 products from Alcatel-Lucent, Certicom, CheckPoint, Cisco, Furukawa, IBM, Ixia, Juniper, Microsoft, Nokia, SafeNet, Secure Computing, SonicWall, and the IPv6 TAHI Project.

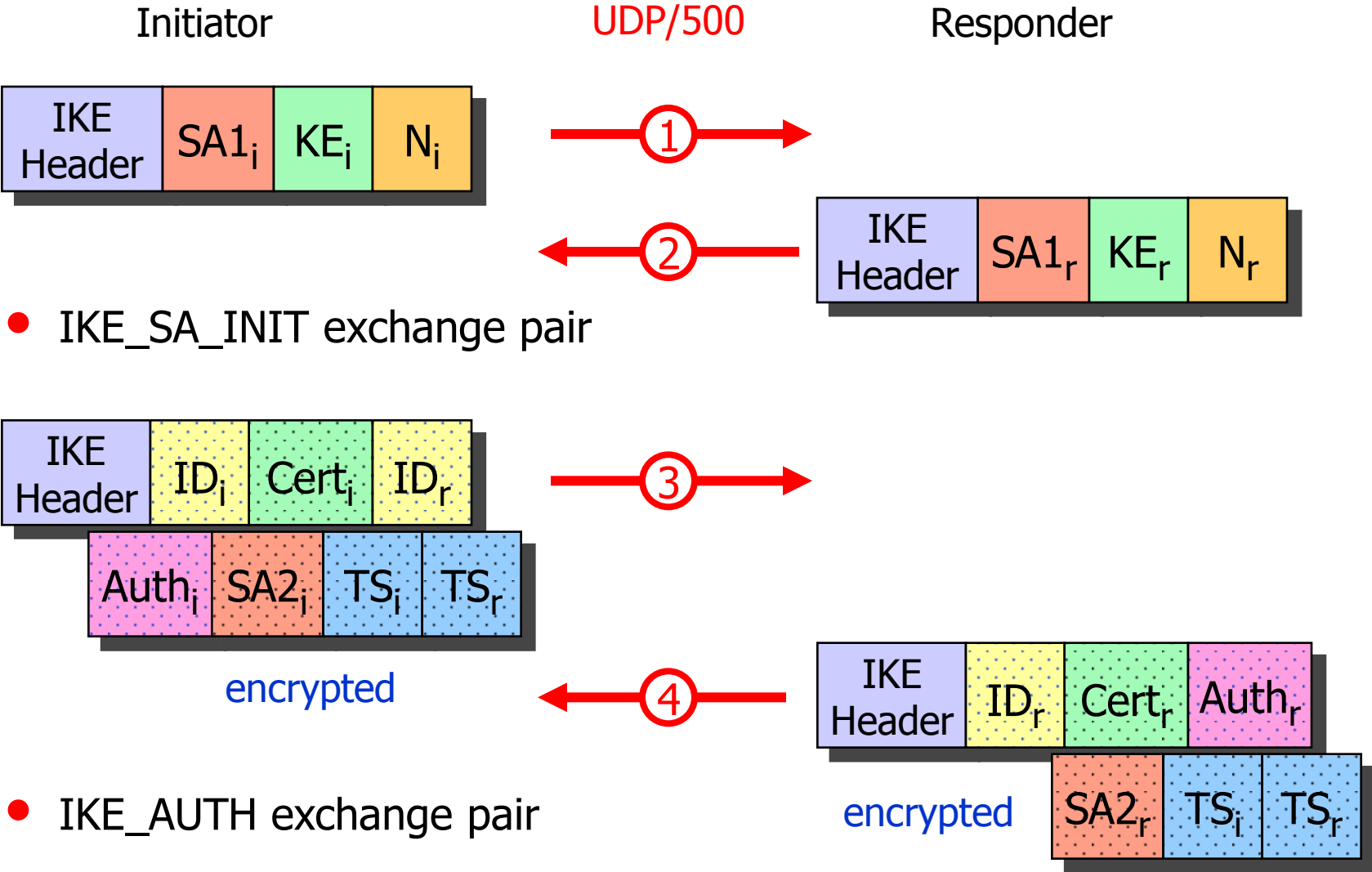
# IPsecv1 versus IPsecv2

# Internet Key Exchange – IKEv1 Main Mode



- IKEv1 Quick Mode – another three messages to negotiate traffic selectors

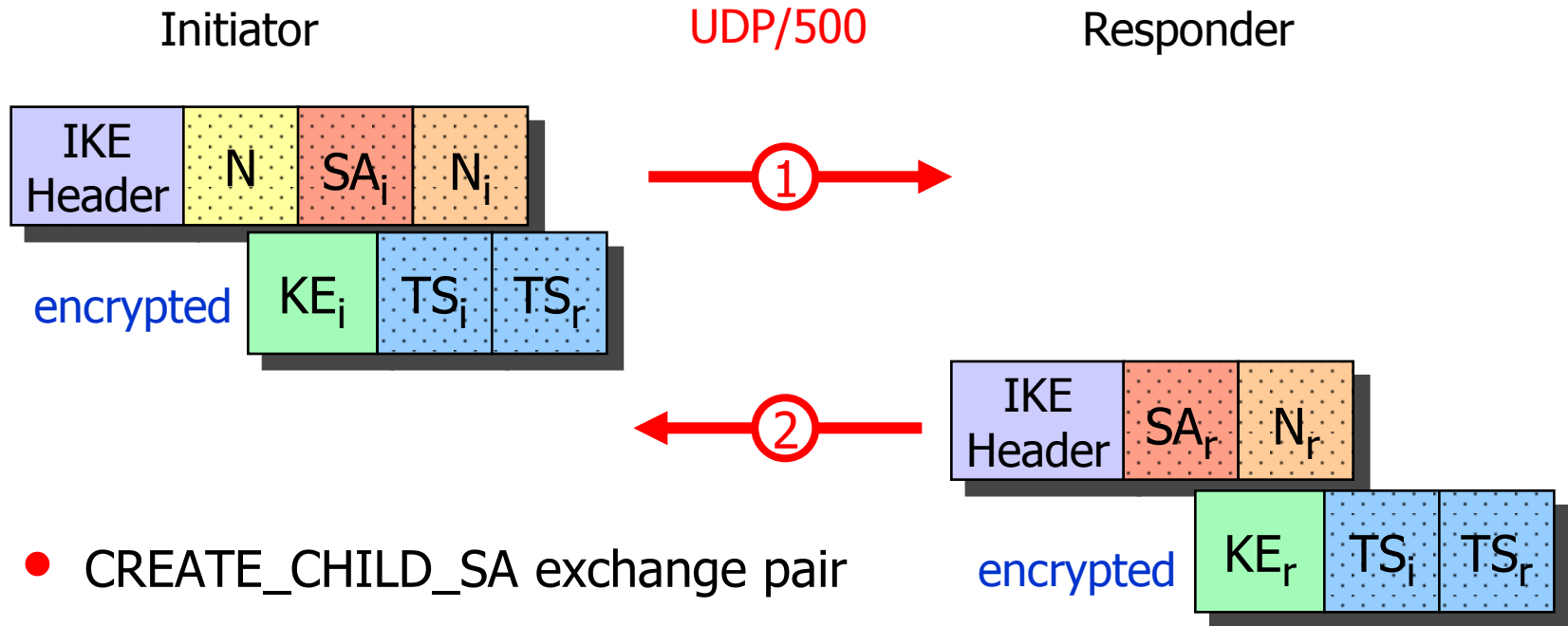
# IKEv2 – Authentication and first Child SA



- IKE\_SA\_INIT exchange pair

- IKE\_AUTH exchange pair

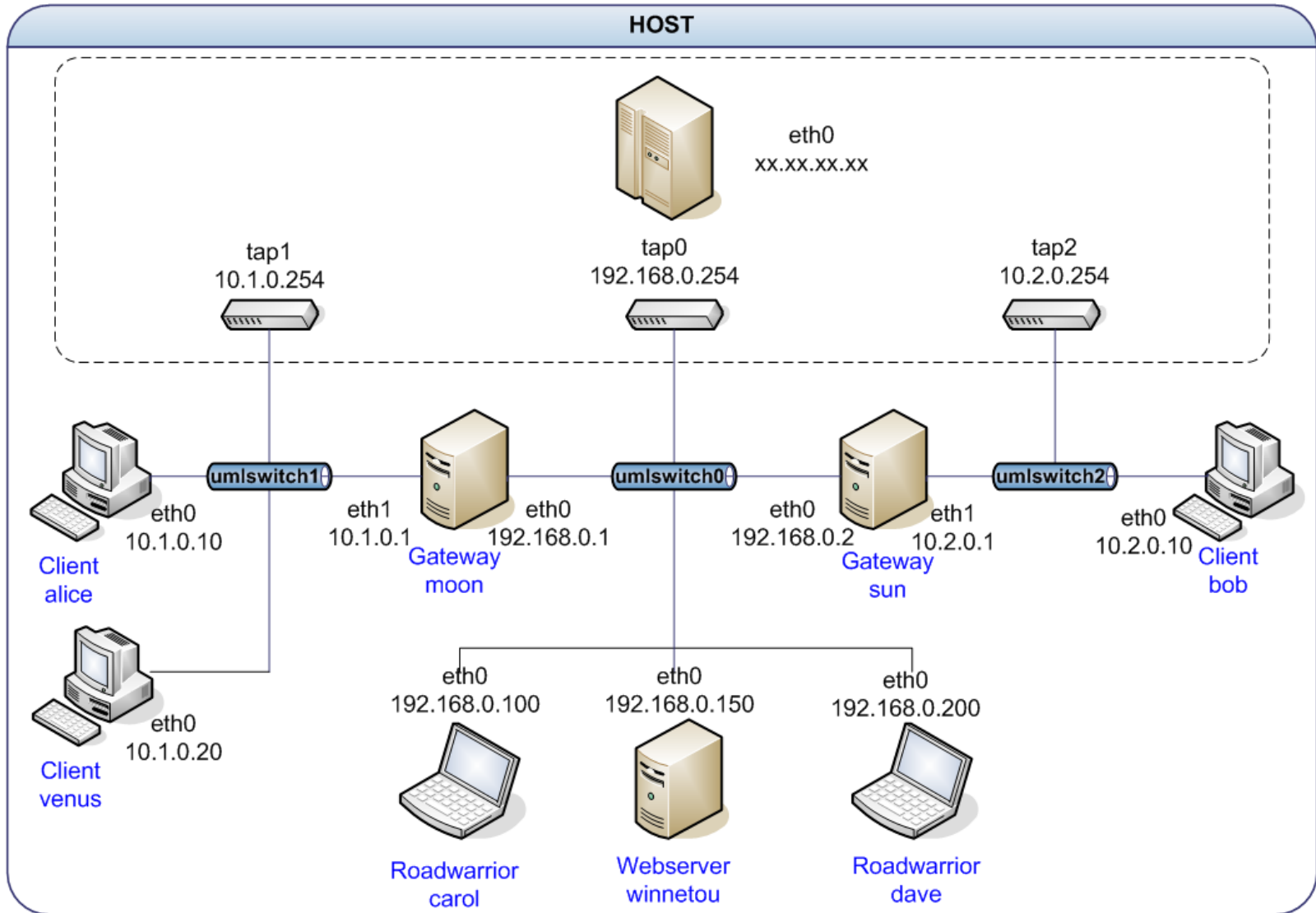
# IKEv2 – Additional Child SAs



- CREATE\_CHILD\_SA exchange pair

# A Simple Remote Access Example

# User-Mode-Linux VPN Testbed



# IKEv2 Remote Access Scenario

```
#ipsec.secrets for roadwarrior carol
: RSA carolKey.pem "nH5ZQEWtku0RJZ6"
```

```
#ipsec.conf for roadwarrior carol

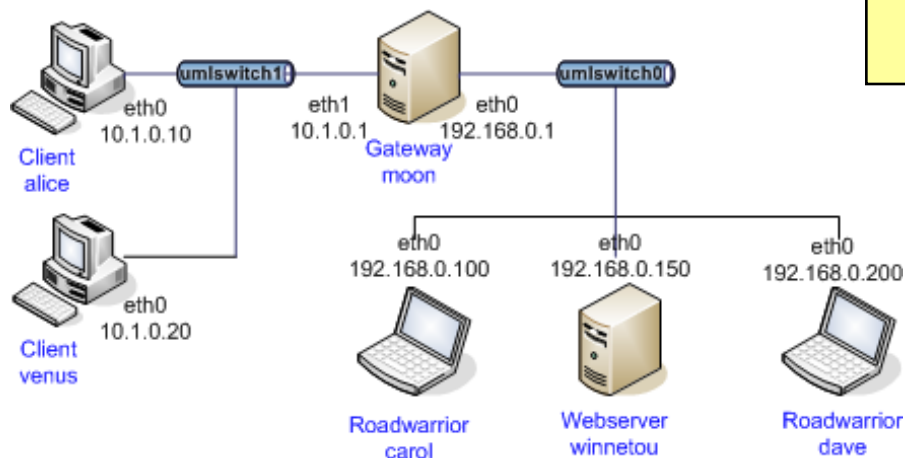
conn home
    keyexchange=ikev2
    left=%defaultroute
    leftsourceip=%config
    leftcert=carolCert.pem
    leftid=carol@strongswan.org
    leftfirewall=yes
    right=192.168.0.1
    rightid=@moon.strongswan.org
    rightsubnet=10.1.0.0/16
    auto=start
```

```
#ipsec.secrets for gateway moon
: RSA moonKey.pem
```

```
#ipsec.conf for gateway moon

config setup
    plutostart=no #IKEv1 not needed

conn rw
    keyexchange=ikev2
    left=%any
    leftsubnet=10.1.0.0/24
    leftcert=moonCert.pem
    leftid=@moon.strongswan.org
    leftfirewall=yes
    right=%any
    rightsourceip=10.3.0.0/24
    auto=add
```





# IKEv2 Connection Setup

carol

```
05[ENC] generating IKE_SA_INIT request [SA KE No N(NATD_S_IP) N(NATD_D_IP)]
05[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
06[NET] received packet: from 192.168.0.1[500] to 192.168.0.100[500]
06[ENC] parsed IKE_SA_INIT response [SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ]
06[ENC] generating IKE_AUTH request [IDi CERT CERTREQ IDr AUTH CP SA TSi TSr]
06[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
07[NET] received packet: from 192.168.0.1[500] to 192.168.0.100[500]
07[ENC] parsed IKE_AUTH response [IDr CERT AUTH CP SA TSi TSr N(AUTH_LFT)]
07[IKE] installing new virtual IP 10.3.0.1
07[AUD] established CHILD_SA successfully
```

moon

```
05[NET] received packet: from 192.168.0.100[500] to 192.168.0.1[500]
05[ENC] parsed IKE_SA_INIT request [SA KE No N(NATD_S_IP) N(NATD_D_IP)]
05[ENC] generating IKE_SA_INIT response [SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ]
05[NET] sending packet: from 192.168.0.1[500] to 192.168.0.100[500]
06[NET] received packet: from 192.168.0.100[500] to 192.168.0.1[500]
06[ENC] parsed IKE_AUTH request [IDi CERT CERTREQ IDr AUTH CP SA TSi TSr]
06[IKE] peer requested virtual IP %any
06[IKE] assigning virtual IP 10.3.0.1 to peer
06[AUD] established CHILD_SA successfully
06[ENC] generating IKE_AUTH response [IDr CERT AUTH CP SA TSi TSr N(AUTH_LFT)]
06[NET] sending packet: from 192.168.0.1[500] to 192.168.0.100[500]
```

- No port floating to 4500 with **mobike=no**

# IKEv2 Connection Setup with MOBIKE

carol

```
05[ENC] generating IKE_SA_INIT request [SA KE No N(NATD_S_IP) N(NATD_D_IP)]
05[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]
06[NET] received packet: from 192.168.0.1[500] to 192.168.0.100[500]
06[ENC] parsed IKE_SA_INIT response [SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ]
06[ENC] generating IKE_AUTH request [IDi .. N(MOBIKE_SUP) N(ADD_6_ADDR)]
06[NET] sending packet: from 192.168.0.100[4500] to 192.168.0.1[4500]
07[NET] received packet: from 192.168.0.1[4500] to 192.168.0.100[4500]
07[ENC] parsed IKE_AUTH response [IDr .. N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_6_ADDR)+]
07[IKE] installing new virtual IP 10.3.0.1
07[AUD] established CHILD_SA successfully
```

moon

```
05[NET] received packet: from 192.168.0.100[500] to 192.168.0.1[500]
05[ENC] parsed IKE_SA_INIT request [SA KE No N(NATD_S_IP) N(NATD_D_IP)]
05[ENC] generating IKE_SA_INIT response [SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ]
05[NET] sending packet: from 192.168.0.1[500] to 192.168.0.100[500]
06[NET] received packet: from 192.168.0.100[4500] to 192.168.0.1[4500]
06[ENC] parsed IKE_AUTH request [IDi .. N(MOBIKE_SUP) N(ADD_6_ADDR)]
06[IKE] peer requested virtual IP %any
06[IKE] assigning virtual IP 10.3.0.1 to peer
06[AUD] established CHILD_SA successfully
06[ENC] generating IKE_AUTH resp [IDr .. N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_6_ADDR)+]
06[NET] sending packet: from 192.168.0.1[4500] to 192.168.0.100[4500]
```

- Port floating to 4500 by default

# Narrowing Traffic Selectors

carol

```
carol> ipsec statusall

Connections:
  home: 192.168.0.100...192.168.0.1
  home: local: [carol@strongswan.org] uses public key authentication
  home: cert: "C=CH, O=Linux strongSwan, OU=Research, CN=carol@strongswan.org"
  home: remote: [moon.strongswan.org] uses any authentication
  home: child: dynamic === 10.1.0.0/16

Security Associations:
  home[1]: ESTABLISHED 14 seconds ago, 192.168.0.100[carol@strongswan.org]...
           192.168.0.1[moon.strongswan.org]
  home[1]: IKE SPIs: 23b9b14113e91e86_i* 0315c61d96ef0a4f_r, reauth. in 2 hours
  home[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
  home{1}: INSTALLED, TUNNEL, ESP SPIs: cb342ccc_i c9d6623b_o
  home{1}: AES_CBC_128/HMAC_SHA1_96, 84 bytes_i (10s ago), 84 bytes_o (10s ago),
  home{1}: 10.3.0.1/32 === 10.1.0.0/24
```

- In the most extreme case the remote access client just proposes the widest possible traffic selector of **0.0.0.0/0** and lets the VPN gateway decide which networks and protocols to grant access to.

# IKEv2 Configuration Payload

carol

```
carol> ip addr list dev eth0
eth0: inet 192.168.0.100/24 brd 192.168.0.255 scope global eth0
       inet 10.3.0.1/32 scope global eth0

carol> ip route list table 220
10.1.0.0/24 dev eth0 proto static src 10.3.0.1
```

- A virtual IP requested and obtained through `leftsourceip=%config` is directly configured by strongSwan via the RT Netlink socket

moon

```
moon> ip addr list
eth0: inet 192.168.0.1/24 brd 192.168.0.255 scope global eth0
eth1: inet 10.1.0.1/16 brd 10.1.255.255 scope global eth1

moon> ip route list table 220
10.3.0.1 dev eth0 proto static src 10.1.0.1
```

- If a host has an internal interface which is part of the negotiated traffic selectors then this source address is assigned to tunneled IP packets.

# Virtual IP Address Pools

# Volatile RAM-based IP Address Pools

- Configuration in ipsec.conf

```
conn rw
  keyexchange=ikev2
  ...
  rightsourceip=10.3.0.0/24
  auto=add
```

- Statistics

```
ipsec leases

Leases in pool 'rw', usage: 2/255, 2 online
    10.3.0.2   online   'dave@strongswan.org'
    10.3.0.1   online   'carol@strongswan.org'
```

- Referencing and sharing a volatile pool

```
conn rw1
  keyexchange=ikev2
  ...
  rightsourceip=%rw
  auto=add
```

- SQLite database table definitions

```
http://wiki.strongswan.org/repositories/entry/strongswan/  
testing/hosts/default/etc/ipsec.d/tables.sql
```

- Creation of SQLite database

```
cat /etc/ipsec.d/table.sql | sqlite3 /etc/ipsec.d/ipsec.db
```

- Connecting to the SQLite database

```
# /etc/strongswan.conf - strongSwan configuration file  
  
libstrongswan {  
  plugins {  
    attr-sql {  
      database = sqlite:///etc/ipsec.d/ipsec.db  
    }  
  }  
}
```

- Pool creation

```
ipsec pool --add bigpool --start 10.3.0.1 --end 10.3.0.254 --timeout 48  
allocating 254 addresses... done.
```

- Configuration in ipsec.conf

```
conn rw  
    keyexchange=ikev2  
    ...  
    rightsourcetypeip=%bigpool  
    auto=add
```

- Statistics

```
ipsec pool --status  
name      start      end          timeout    size    online    usage  
bigpool   10.3.0.1   10.3.0.254  48h       254    1 ( 0%)  2 ( 0%)  
  
ipsec pool --leases --filter pool=bigpool  
name      address    status start          end          identity  
bigpool   10.3.0.1  online Oct 22 23:13:50 2009          carol@strongswan.org  
bigpool   10.3.0.2  valid  Oct 22 23:14:11 2009 Oct 22 23:14:25 2009 dave@strongswan.org
```



# Certificate Revocation Mechanisms

# HTTP or LDAP based CRL Fetching

## crlDistributionPoints extension in user certificate

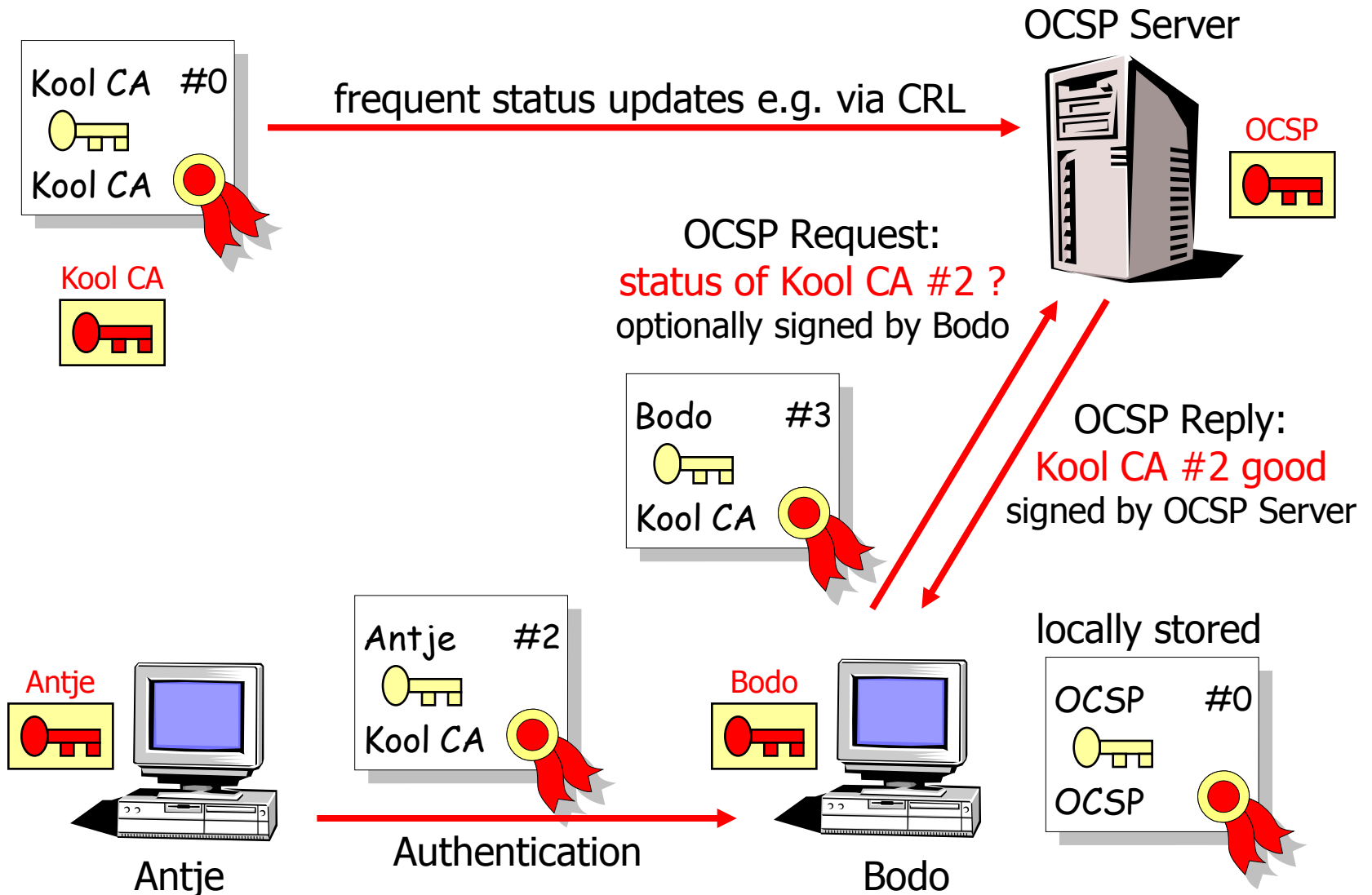
```
crlDistributionPoints = URI:http://crl.strongswan.org/strongswan.crl
```

```
# ipsec.conf
config setup
    strictcrlpolicy=yes
    cachecrls=yes

ca strongswan
    cacert=strongswanCert.pem
    crluri="ldap://ldap.strongswan.org/cn=strongSwan Root CA,
           o=Linux strongSwan, c=CH?certificateRevocationList"
    auto=add
```

```
13[CFG] checking certificate status of "C=CH, O=Linux strongSwan, OU=Research,
                                         CN=carol@strongswan.org"
13[CFG]   fetching crl from 'http://crl.strongswan.org/strongswan.crl' ...
13[CFG]   using trusted certificate "C=CH, O=Linux strongSwan,
                                         CN=strongSwan Root CA"
13[CFG]   crl correctly signed by "C=CH, O=Linux strongSwan,
                                         CN=strongSwan Root CA"
13[CFG]   crl is valid: until Nov 15 22:42:42 2009
13[CFG] certificate status is good
13[LIB]   written crl file '/etc/ipsec.d/crls/5da7...4def.crl' (942 bytes)
```

# Online Certificate Status Protocol (OCSP) with self-signed OCSP certificate



# OCSP with self-signed OCSP Certificate

moon

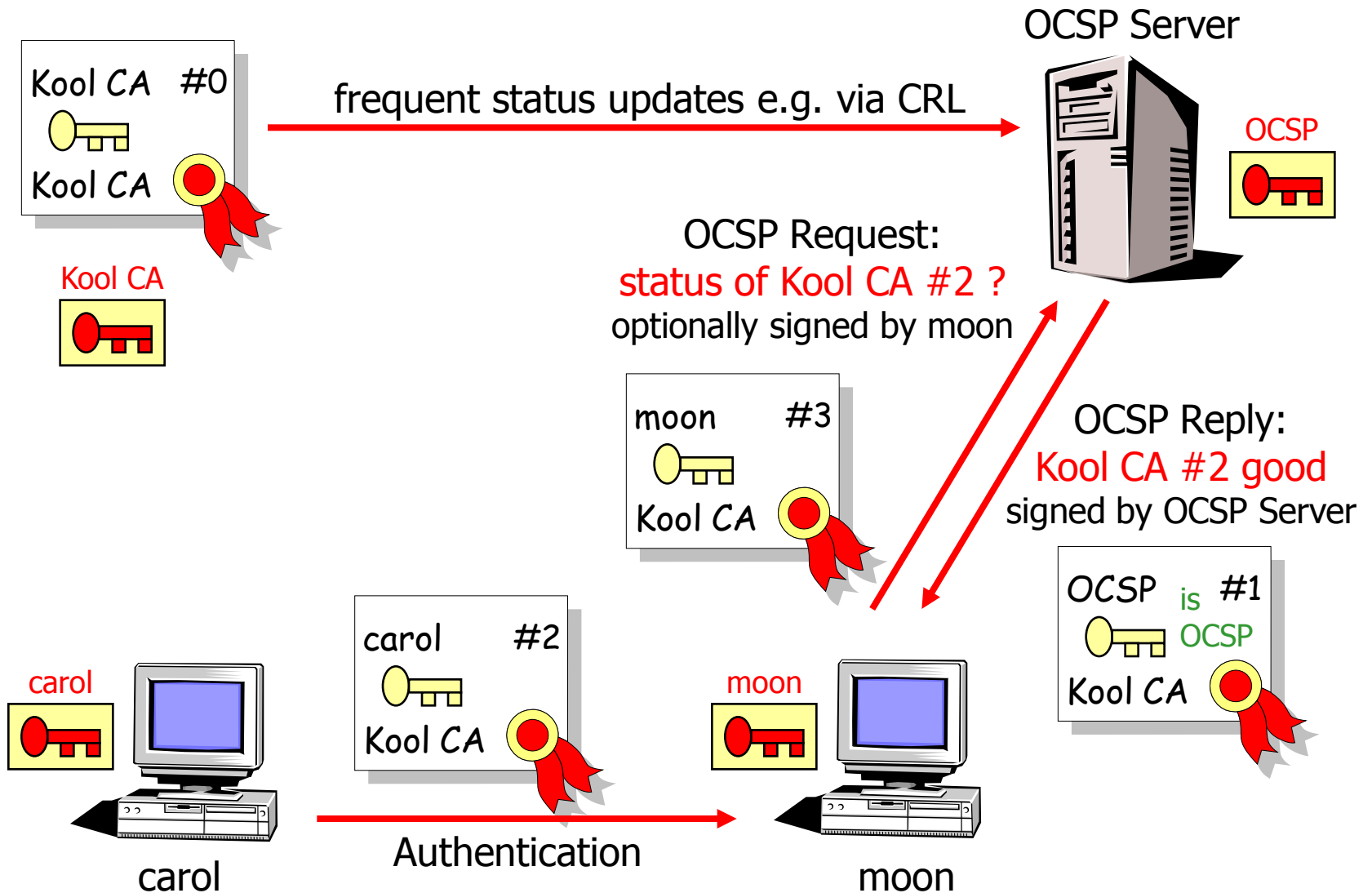
```
# /etc/ipsec.conf

ca strongswan
  cacert=strongswanCert.pem
  ocsपुरि=http://ocsp.strongswan.org:8880
  auto=add
```

```
13[CFG] checking certificate status of "C=CH, O=Linux strongSwan,
      OU=Research, CN=carol@strongswan.org"
13[CFG]   requesting ocsp status from 'http://ocsp.strongswan.org:8880' ...
13[CFG]   using trusted certificate "C=CH, O=Linux strongSwan,
      OU=OCSP Self-Signed Authority, CN=ocsp.strongswan.org"
13[CFG]   ocsp response correctly signed by "C=CH, O=Linux strongSwan,
      OU=OCSP Self-Signed Authority, CN=ocsp.strongswan.org"
13[CFG]   ocsp response is valid: until Oct 17 02:11:09 2009
13[CFG] certificate status is good
```

```
ipsec listcainfos
  authname:      "C=CH, O=Linux strongSwan, CN=strongSwan Root CA"
  authkey:       5d:a7:dd:70:06:51:32:7e:e7:b6:6d:b3:b5:e5:e0:60:ea:2e:4d:ef
  keyid:         ae:09:6b:87:b4:48:86:d3:b8:20:97:86:23:da:bd:0e:ae:22:eb:bc
  ocsपुरि:       'http://ocsp.strongswan.org:8880'
```

# Online Certificate Status Protocol (OCSP) with delegated trust



# OCSP with Delegated Trust

extendedKeyUsage flag in OCSP-signer certificate

```
extendedKeyUsage = OCSPSigning
```

carol: authorityInfoAccess extension in user certificate

```
authorityInfoAccess = OCSP;URI:http://ocsp.strongswan.org:8880
```

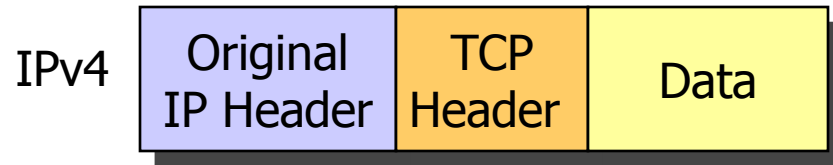
moon

```
11[CFG] checking certificate status of "C=CH, O=Linux strongSwan,  
OU=OCSP, CN=carol@strongswan.org"  
11[CFG] requesting ocp status from 'http://ocsp.strongswan.org:8880' ...  
11[CFG] using certificate "C=CH, O=Linux strongSwan,  
OU=OCSP Signing Authority, CN=ocsp.strongswan.org"  
11[CFG] using trusted ca certificate "C=CH, O=Linux strongSwan,  
CN=strongSwan Root CA"  
11[CFG] ocp response correctly signed by "C=CH, O=Linux strongSwan,  
OU=OCSP Signing Authority, CN=ocsp.strongswan.org"  
11[CFG] ocp response is valid: until Oct 17 02:13:21 2009  
11[CFG] certificate status is good
```

# The NETKEY IPsec Stack of the Linux 2.6 Kernel

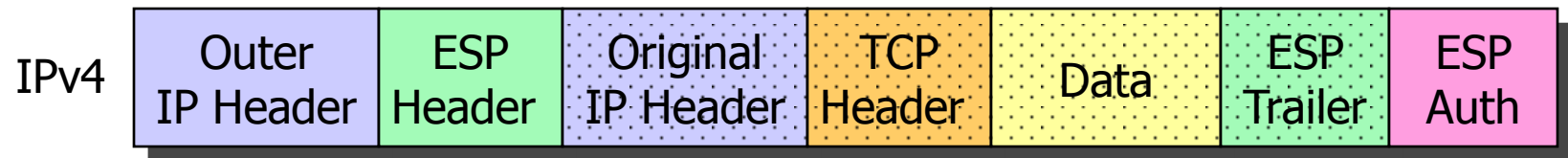
# IPsec Tunnel Mode using ESP

Before applying ESP



Encapsulating Security Payload (ESP): RFC 4303

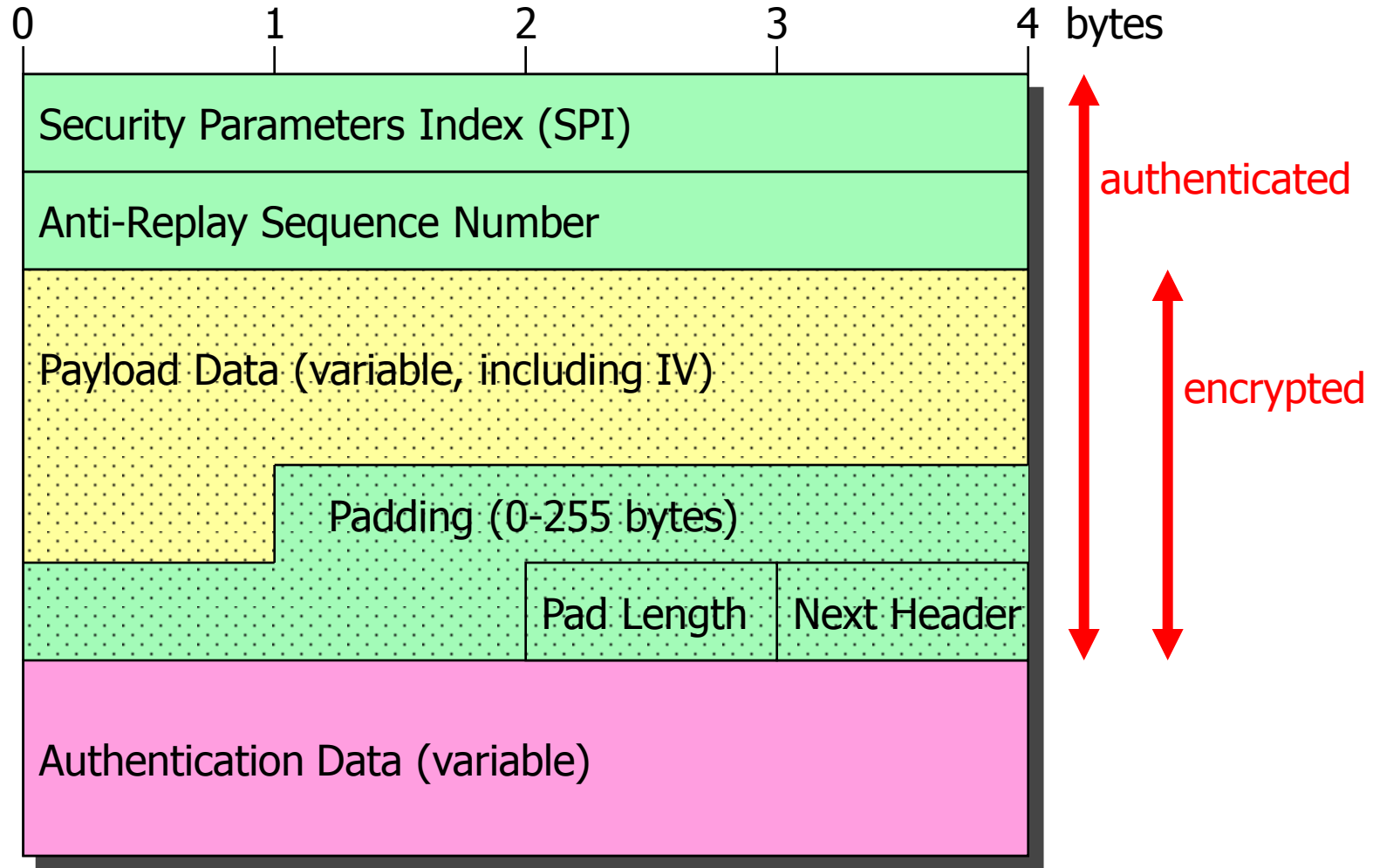
After applying ESP



- IP protocol number for ESP: **50**
- ESP authentication is optional but often used in place of AH
- Original IP Header is encrypted and therefore hidden



# ESP Header (Header / Payload / Trailer)





# IPsec Policies in the Linux Kernel

carol: ip -s xfrm policy

```
src 10.1.0.0/24 dst 10.3.0.1/32 uid 0
  dir in action allow index 800 priority 1760 ...
  lifetime config:
    limit: soft (INF) (bytes), hard (INF) (bytes)
    limit: soft (INF) (packets), hard (INF) (packets)
    expire add: soft 0(sec), hard 0(sec)
    expire use: soft 0(sec), hard 0(sec)
  lifetime current:
    0(bytes), 0(packets)
    add 2009-10-22 20:34:37 use 2009-10-22 20:34:39
  tmpl src 192.168.0.1 dst 192.168.0.100
    proto esp spi 0x00000000(0) reqid 1(0x00000001) mode tunnel

src 10.3.0.1/32 dst 10.1.0.0/24 uid 0
  dir out action allow index 793 priority 1680 ...
  lifetime config:
    limit: soft (INF) (bytes), hard (INF) (bytes)
    limit: soft (INF) (packets), hard (INF) (packets)
    expire add: soft 0(sec), hard 0(sec)
    expire use: soft 0(sec), hard 0(sec)
  lifetime current:
    0(bytes), 0(packets)
    add 2009-10-22 20:34:37 use 2009-10-22 20:34:38
  tmpl src 192.168.0.100 dst 192.168.0.1
    proto esp spi 0x00000000(0) reqid 1(0x00000001) mode tunnel
```

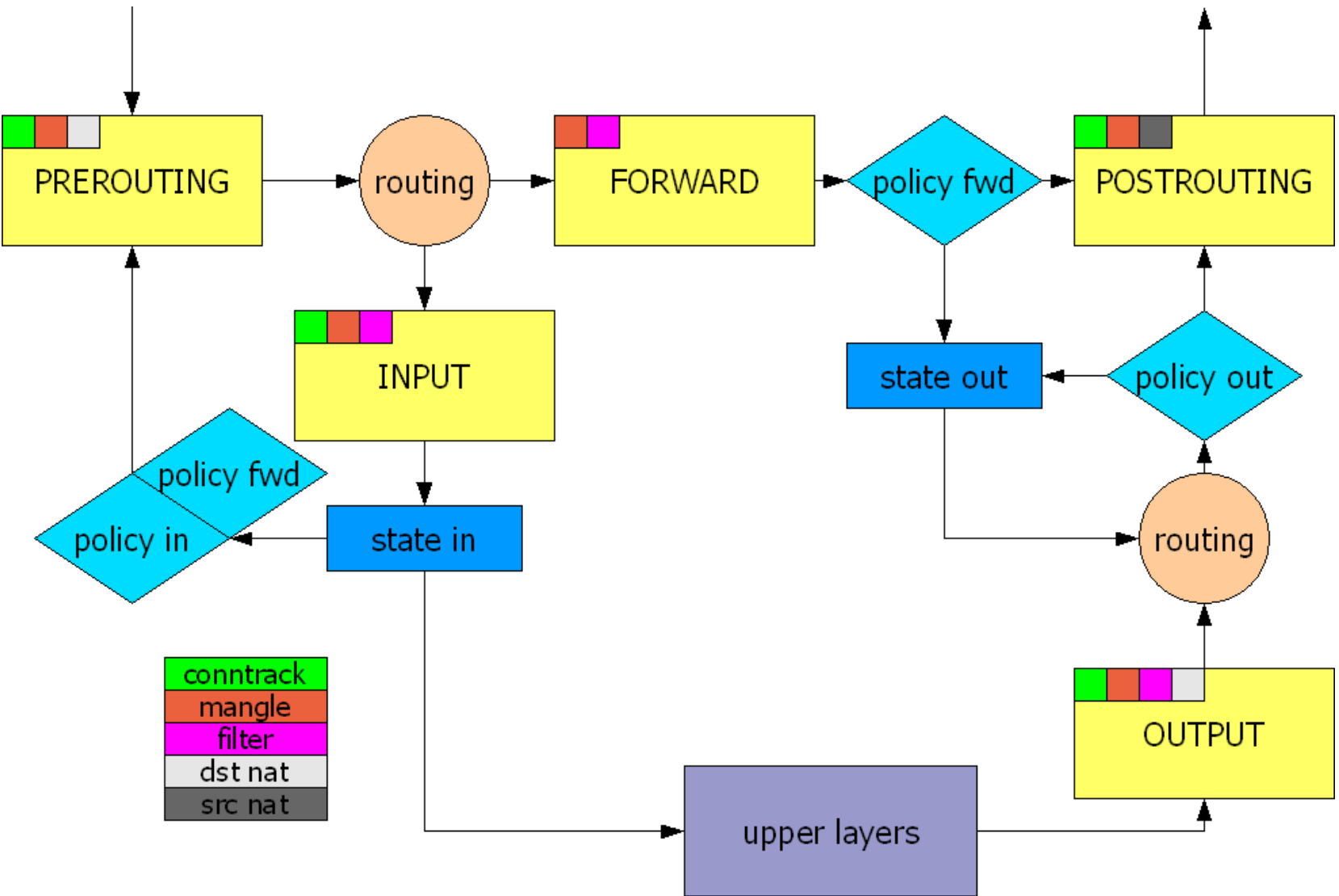
# IPsec Policies in the Linux Kernel

moon: ip -s xfrm policy

```
src 10.3.0.1/32 dst 10.1.0.0/24 uid 0
  dir fwd action allow index 954 priority 1680 ...
  lifetime config:
    limit: soft (INF) (bytes), hard (INF) (bytes)
    limit: soft (INF) (packets), hard (INF) (packets)
    expire add: soft 0(sec), hard 0(sec)
    expire use: soft 0(sec), hard 0(sec)
  lifetime current:
    0(bytes), 0(packets)
    add 2009-10-22 20:34:36 use 2009-10-22 20:34:39
  tmpl src 192.168.0.100 dst 192.168.0.1
    proto esp spi 0x00000000(0) reqid 1(0x00000001) mode tunnel

src 10.1.0.0/24 dst 10.3.0.1/32 uid 0
  dir out action allow index 937 priority 1760 ...
  lifetime config:
    limit: soft (INF) (bytes), hard (INF) (bytes)
    limit: soft (INF) (packets), hard (INF) (packets)
    expire add: soft 0(sec), hard 0(sec)
    expire use: soft 0(sec), hard 0(sec)
  lifetime current:
    0(bytes), 0(packets)
    add 2009-10-22 20:34:36 use 2009-10-22 20:34:39
  tmpl src 192.168.0.1 dst 192.168.0.100
    proto esp spi 0x00000000(0) reqid 1(0x00000001) mode tunnel
```

# NETKEY Hooks in Linux Netfilter



# IKE SA and IPsec SA Rekeying Criteria

```
#ipsec.conf
conn %default
    ikelifetime=180m          # default 180m
    lifetime=60m             # default 60m
    lifebytes=500000000      # default 0 (no hard limit)
    lifepackets=1000000      # default 0 (no hard limit)
    margintime=6m            # default 9m
    marginbytes=50000000     # default 0 (no soft limit)
    marginpackets=100000     # default 0 (no soft limit)
    rekeyfuzz=50%           # default 100%
    keyingtries=1           # default 0 (forever)
    rekey=yes                # default yes
    reauth=no                # default yes
```

Legacy parameters:

- keylife: synonym for **lifetime**
- rekeymargin: synonym for **margintime**

# IPsec Security Associations in the Kernel

carol: ip -s xfrm state

```
src 192.168.0.100 dst 192.168.0.1
  proto esp spi 0xc77ca4c3(3346834627) reqid 1(0x00000001) mode tunnel
  replay-window 32 seq 0x00000000 flag 20 (0x00100000)
  auth hmac(sha1) 0x98fe271fd31ba795f158ae17487cb85f8682aefc (160 bits)
  enc cbc(aes) 0x3d0567d65694fcbbfd3257f55b497d6a (128 bits)
  lifetime config:
    limit: soft 445929743(bytes), hard 500000000(bytes)
    limit: soft 871034(packets), hard 1000000(packets)
    expire add: soft 3065(sec), hard 3600(sec)
    expire use: soft 0(sec), hard 0(sec)
  lifetime current:
    84(bytes), 1(packets)
    add 2009-10-22 20:34:37 use 2009-10-22 20:34:38
  stats:
    replay-window 0 replay 0 failed 0

src 192.168.0.1 dst 192.168.0.100
  proto esp spi 0xc46038e1(3294640353) reqid 1(0x00000001) mode tunnel
  replay-window 32 seq 0x00000000 flag 20 (0x00100000)
  auth hmac(sha1) 0x4e2b044e2835297d3f73bbf99289f369ae2d3ed5 (160 bits)
  enc cbc(aes) 0xd2d83f5d7e496ddc9483be57dbbb2757 (128 bits)
  ...
```

# Interaction with the Linux Netfilter Firewall



carol

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in  out  source      destination
 1     84  ACCEPT all  eth0 *    10.1.0.0/24 10.3.0.1
                                     policy match dir in  pol ipsec reqid 1 proto 50
 1    152  ACCEPT esp  eth0 *    0.0.0.0/0   0.0.0.0/0
 2   2069  ACCEPT udp  eth0 *    0.0.0.0/0   0.0.0.0/0   udp spt:500 dpt:500

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in  out  source      destination
 1     84  ACCEPT all  *    eth0 10.3.0.1    10.1.0.0/24
                                     policy match dir out pol ipsec reqid 1 proto 50
 1    152  ACCEPT esp  *    eth0 0.0.0.0/0   0.0.0.0/0
 2   2456  ACCEPT udp  *    eth0 0.0.0.0/0   0.0.0.0/0   udp spt:500 dpt:500
```

- After the successful establishment/deletion of a CHILD\_SA the **updown** plugin dynamically inserts and removes an INPUT and OUTPUT Netfilter **IPsec ESP policy matching** rule via the iptables command executed by the **/usr/libexec/ipsec/\_updown** shell script.

moon

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in out source destination
 2   304 ACCEPT esp eth0 * 0.0.0.0/0 0.0.0.0/0
 4   4896 ACCEPT udp eth0 * 0.0.0.0/0 0.0.0.0/0 udp spt:500 dpt:500

Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in out source destination
 1     84 ACCEPT all eth0 * 10.3.0.2 10.1.0.20
                                     policy match dir in pol ipsec reqid 2 proto 50
 1     84 ACCEPT all * eth0 10.1.0.20 10.3.0.2
                                     policy match dir out pol ipsec reqid 2 proto 50
 1     84 ACCEPT all eth0 * 10.3.0.1 10.1.0.0/24
                                     policy match dir in pol ipsec reqid 1 proto 50
 1     84 ACCEPT all * eth0 10.1.0.0/24 10.3.0.1
                                     policy match dir out pol ipsec reqid 1 proto 50

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in out source destination
 2   304 ACCEPT esp * eth0 0.0.0.0/0 0.0.0.0/0
 4   4138 ACCEPT udp * eth0 0.0.0.0/0 0.0.0.0/0 udp spt:500 dpt:500
```

- On gateways the **updown** plugin dynamically inserts and removes two FORWARD Netfilter **IPsec ESP policy matching** rules per CHILD\_SA.
- **lefthostaccess=yes** additionally adds an input/output rule to access the GW itself.

# Static IPsec ESP Policy Matching Rules

moon

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in  out  source      destination
 2    304 ACCEPT esp  eth0 *    0.0.0.0/0   0.0.0.0/0
 4   4896 ACCEPT udp  eth0 *    0.0.0.0/0   0.0.0.0/0   udp spt:500 dpt:500

Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in  out  source      destination
 2    168 ACCEPT all  *    *    0.0.0.0/0   0.0.0.0/0
                                     policy match dir in  pol ipsec proto 50
 2    168 ACCEPT all  *    *    0.0.0.0/0   0.0.0.0/0
                                     policy match dir out pol ipsec proto 50

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot in  out  source      destination
 2    304 ACCEPT esp  *    eth0 0.0.0.0/0   0.0.0.0/0
 4   4138 ACCEPT udp  *    eth0 0.0.0.0/0   0.0.0.0/0   udp spt:500 dpt:500
```

- `leftfirewall=yes` can be omitted and the `updown` plugin doesn't have to be built (`./configure --disable-updown`).

# Dead Peer Detection (DPD)

# Activation of Dead Peer Detection

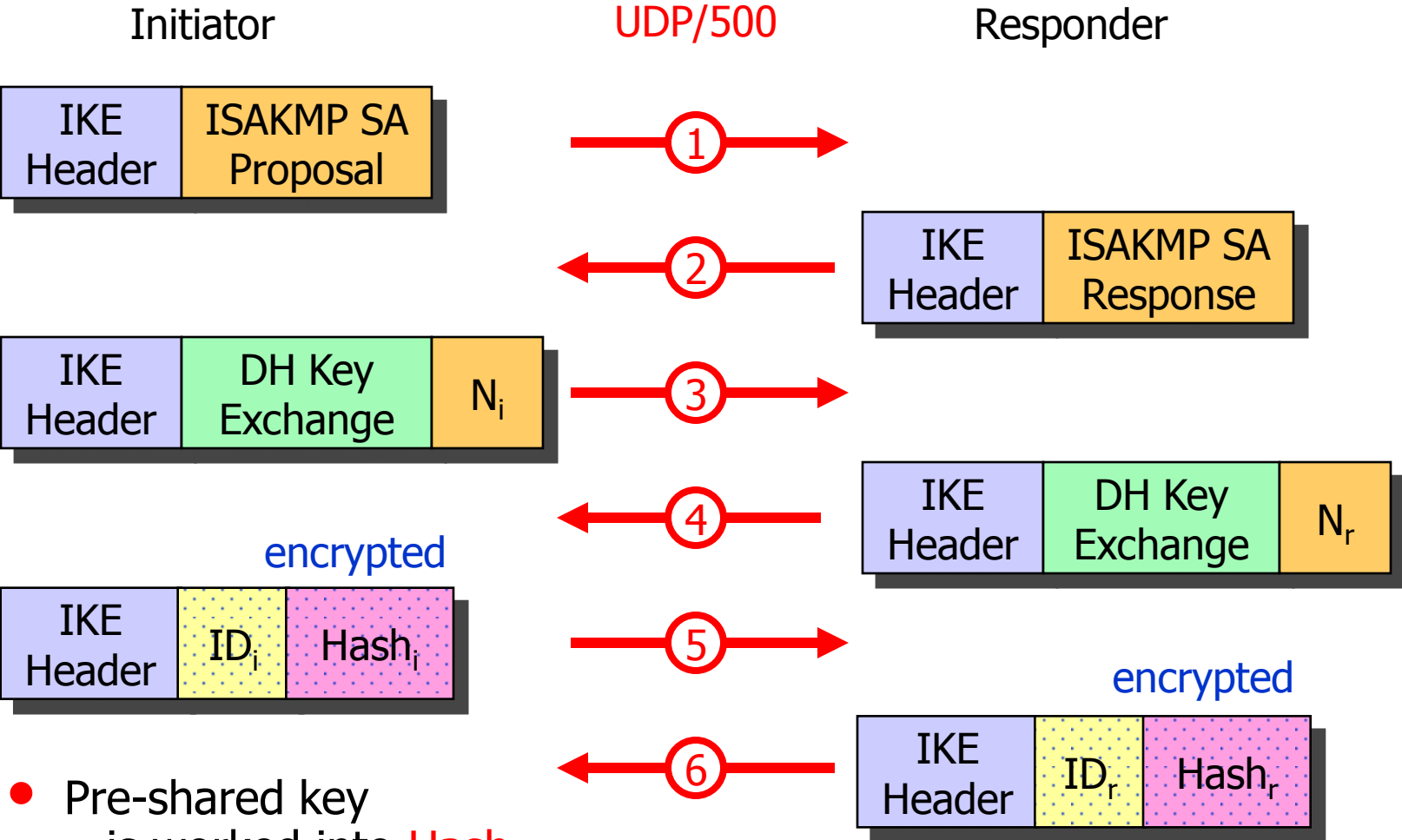
```
#ipsec.conf for roadwarrior carol  
  
conn %default  
    dpddelay=60  
    dpdaction=restart
```

```
#ipsec.conf for gateway moon  
  
conn %default  
    dpddelay=60  
    dpdaction=clear
```

```
Oct 24 11:45:10 13[IKE] CHILD_SA home{1} established with SPIs c50810d9_i c8485f4a_o  
Oct 24 11:46:10 16[NET] received packet: from 192.168.0.1[500] to 192.168.0.100[500]  
Oct 24 11:46:10 16[ENC] parsed INFORMATIONAL request 0 [ ]  
Oct 24 11:46:10 16[ENC] generating INFORMATIONAL response 0 [ ]  
Oct 24 11:46:10 16[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:47:09 09[IKE] sending DPD request  
Oct 24 11:47:09 09[ENC] generating INFORMATIONAL request 2 [ ]  
Oct 24 11:47:09 09[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:47:13 03[IKE] retransmit 1 of request with message ID 2  
Oct 24 11:47:13 03[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:47:20 11[IKE] retransmit 2 of request with message ID 2  
Oct 24 11:47:20 11[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:47:33 08[IKE] retransmit 3 of request with message ID 2  
Oct 24 11:47:33 08[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:47:56 12[IKE] retransmit 4 of request with message ID 2  
Oct 24 11:47:56 12[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:48:38 14[IKE] retransmit 5 of request with message ID 2  
Oct 24 11:48:38 14[NET] sending packet: from 192.168.0.100[500] to 192.168.0.1[500]  
Oct 24 11:49:54 16[IKE] giving up after 5 retransmits  
Oct 24 11:49:54 16[IKE] restarting CHILD_SA home  
Oct 24 11:49:54 16[IKE] initiating IKE_SA home[2] to 192.168.0.1
```

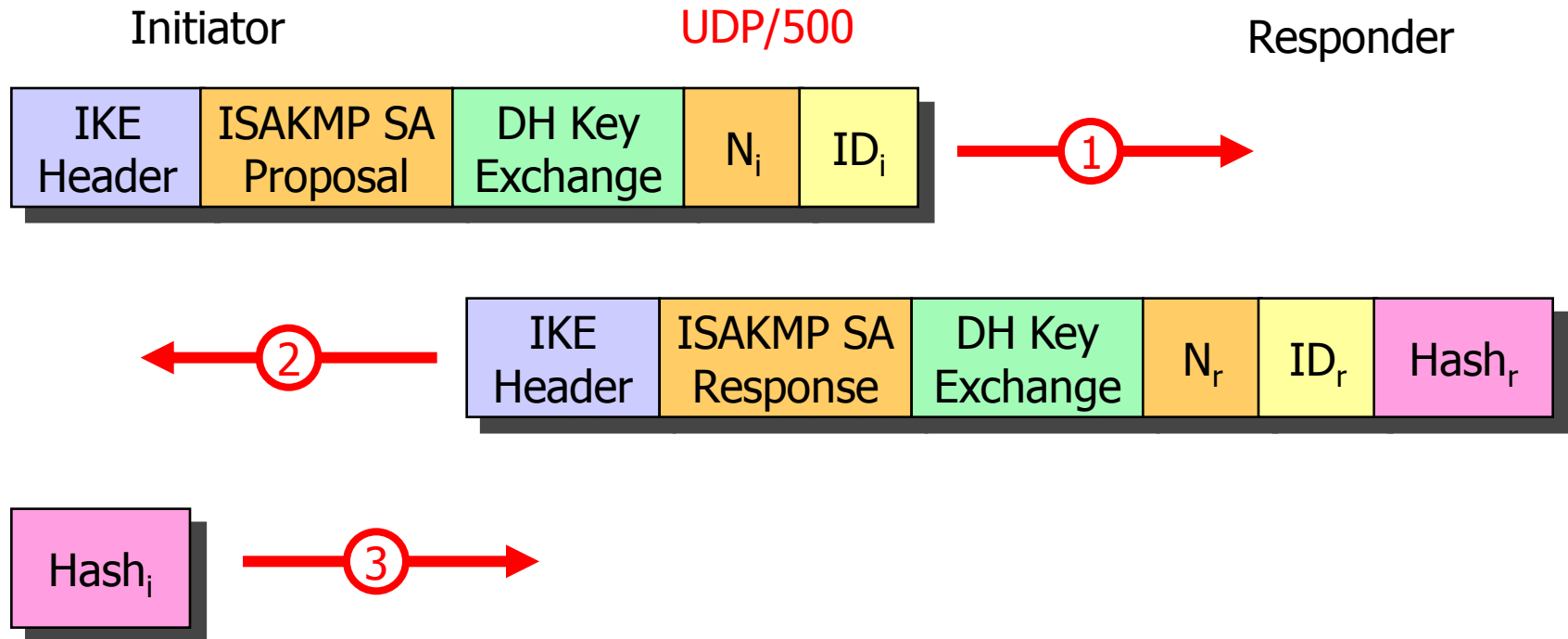
# Remote-Access with Mixed Authentication

# IKEv1 Main Mode using Pre-Shared Keys



- Pre-shared key
  - is worked into Hash
  - is part of the IKE session key

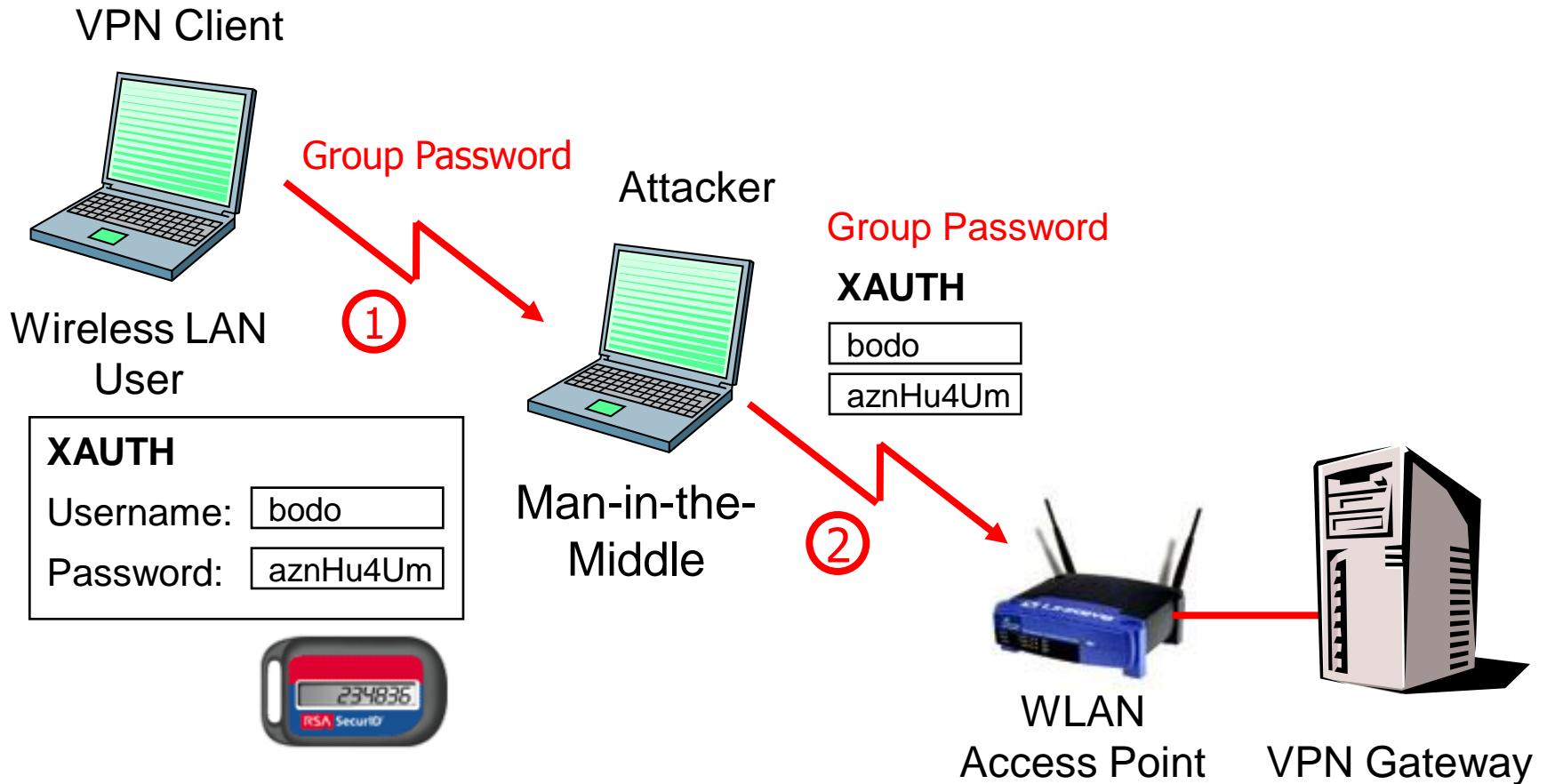
# IKEv1 Aggressive Mode using PSK



- Unencrypted IKE Aggressive Mode messages carrying cleartext IDs can be easily sniffed by a passive attacker.
- Pre-Shared Key is worked into Hash<sub>r</sub>, together with other known parameters, so that an off-line cracking attack becomes possible.



# Man-in-the-Middle Attack possible with IKEv1 Aggressive Mode and XAUTH



- With IKE Aggressive Mode, use One-Time Password scheme (e.g. SecureID).

# IKEv2 Mixed PSK/RSA Authentication

```
#ipsec.secrets for roadwarrior carol
carol@strongswan.org : PSK "gaga5"
```

```
#ipsec.conf for roadwarrior carol

conn home
    keyexchange=ikev2
    leftauth=psk
    left=%defaultroute
    leftid=carol@strongswan.org
    leftfirewall=yes
    rightauth=pubkey
    right=192.168.0.1
    rightid=@moon.strongswan.org
    rightsubnet=0.0.0.0/0
    auto=start
```

- With weak PSKs vulnerable to MITM dictionary attacks since user sends credentials first!
- Users choose weak passwords!

```
#ipsec.secrets for gateway moon
: RSA moonKey.pem

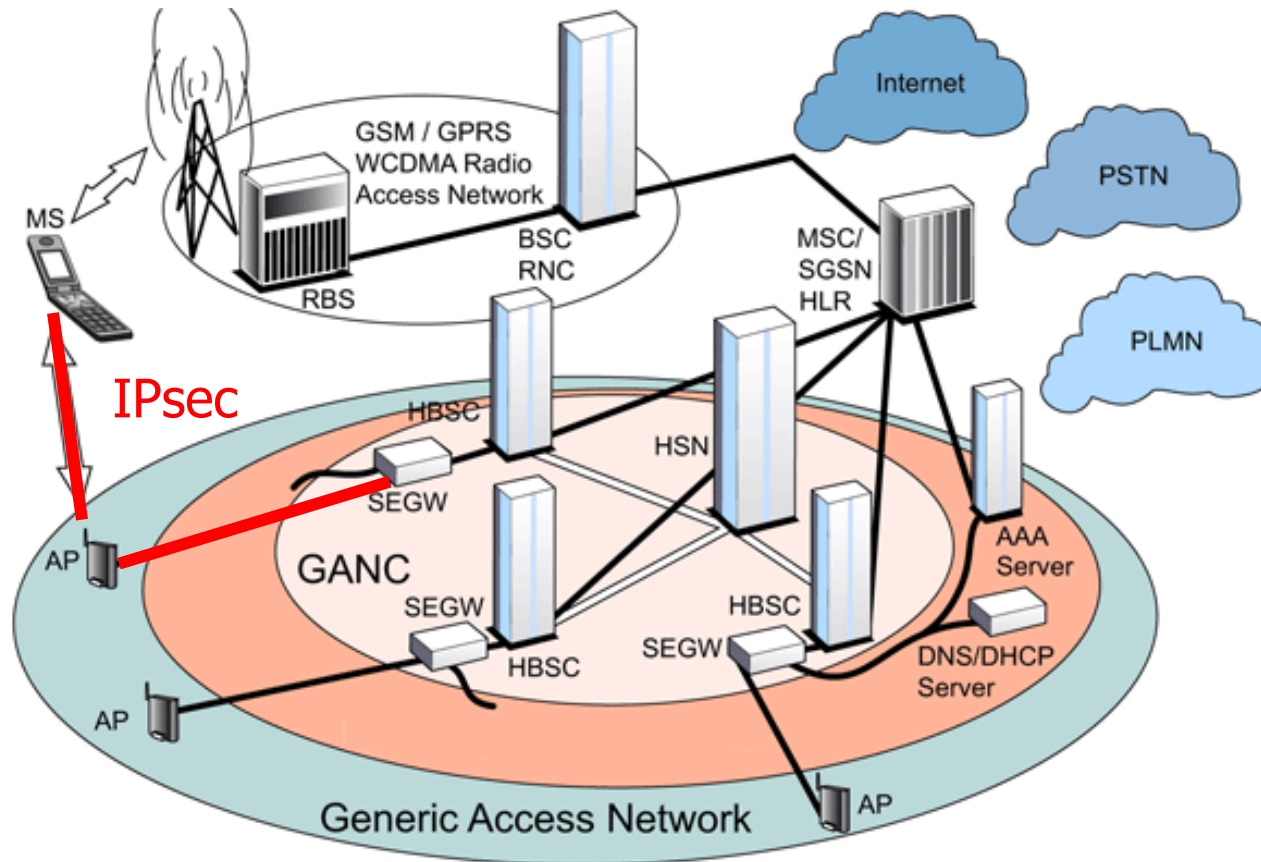
carol@strongswan.org : \
    PSK „gaga5"

dave@strongswan.org : \
    PSK "jVzONCF02ncsgiSImIXeqhGN"
```

```
#ipsec.conf for gateway moon

conn rw
    keyexchange=ikev2
    leftauth=pubkey
    left=%any
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    leftid=@moon.strongswan.org
    leftfirewall=yes
    rightauth=psk
    right=%any
    rightsourceip=10.3.0.0/24
    auto=add
```

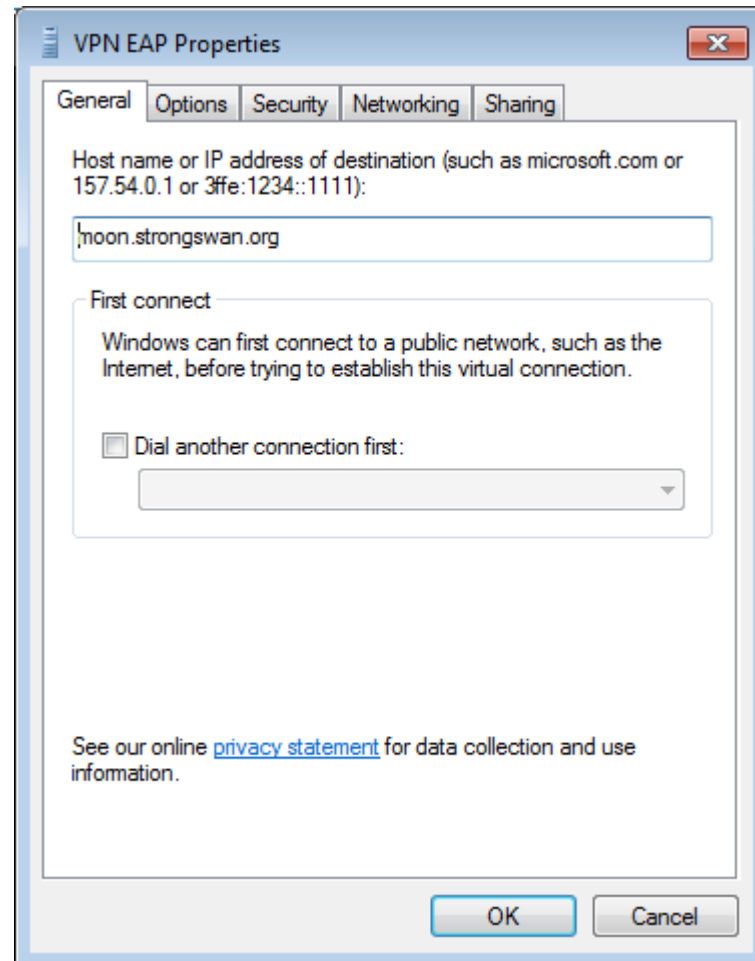
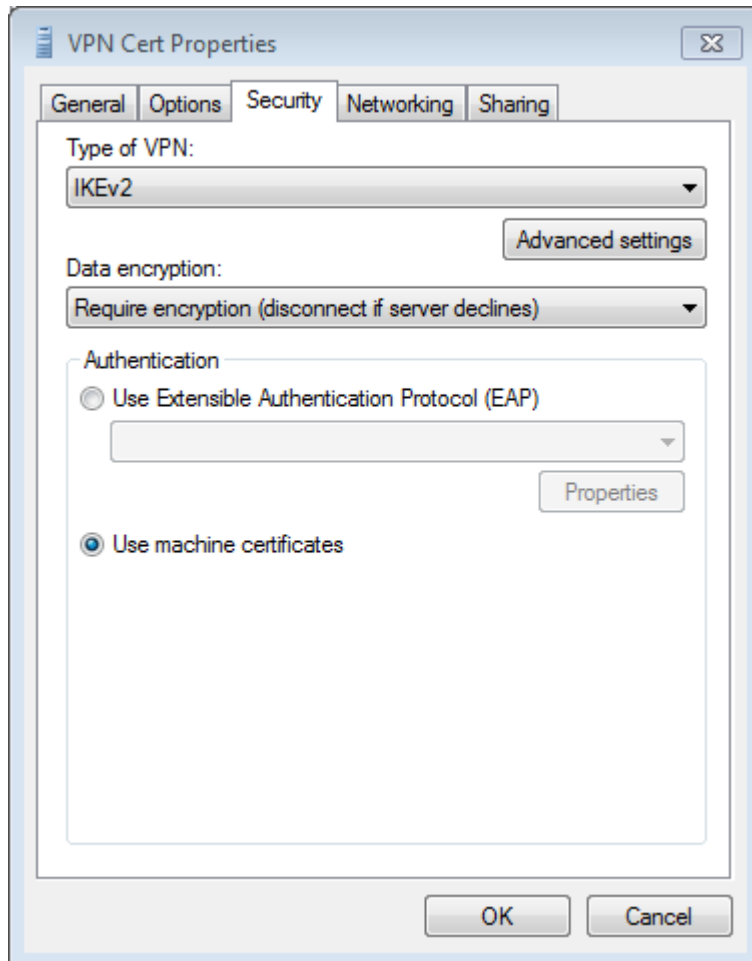
# IKEv2 EAP Authentication



- The 3GPP Generic Access Network (GAN) enables GSM and UMTS services to be delivered over unlicensed WLAN Access Points (APs). Using **IKEv2 EAP-SIM** or **EAP-AKA** authentication the Mobile Station (MS) sets up an IPsec tunnel to the GAN Controller (GANC).

# Interoperability with the Windows 7 Agile VPN Client

# Windows 7 VPN with Machine Certificates



- Gateway certificate must contain host name [or IP address] and the **serverAuth** extendedKeyUsage flag.

# Windows 7 VPN Cert Status

VPN Cert Status

General Details

Property	Value
Device Name	WAN Miniport (IKEv2)
Device Type	vpn
Authentication	Machine certificate
Encryption	IPsec: AES 256
Mobike Supported	Yes
Client IPv4 address	10.3.0.1
Server IPv4 address	0.0.0.0
NAP State	Not NAP-capable
Network Adapter Used	Local Area Connection
Origin address	10.0.2.15
Destination address	192.168.0.1

Close

Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DN...	
Description	VPN Cert
Physical Address	
DHCP Enabled	No
IPv4 Address	10.3.0.1
IPv4 Subnet Mask	255.255.255.255
IPv4 Default Gateway	
IPv4 DNS Servers	62.2.17.60 62.2.24.162
IPv4 WINS Server	10.1.0.10
NetBIOS over Tcpi... En...	Yes

```
# strongswan.conf for gateway moon

charon {
  dns1=62.2.17.60
  dns2=62.2.24.162
  nbns1=10.1.0.10
}
```

# Local EAP Credentials Management

```
# ipsec.secrets for gateway moon

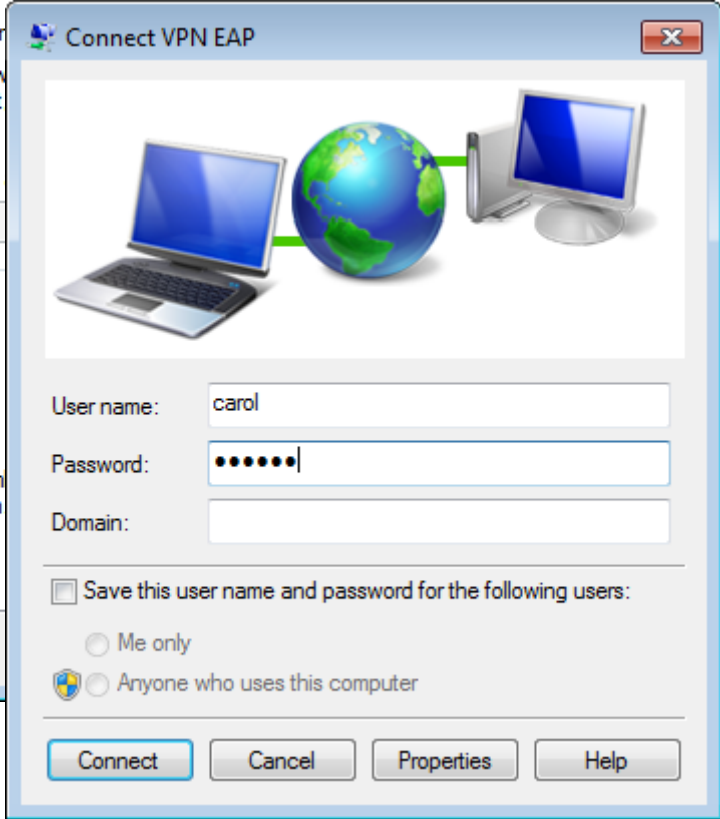
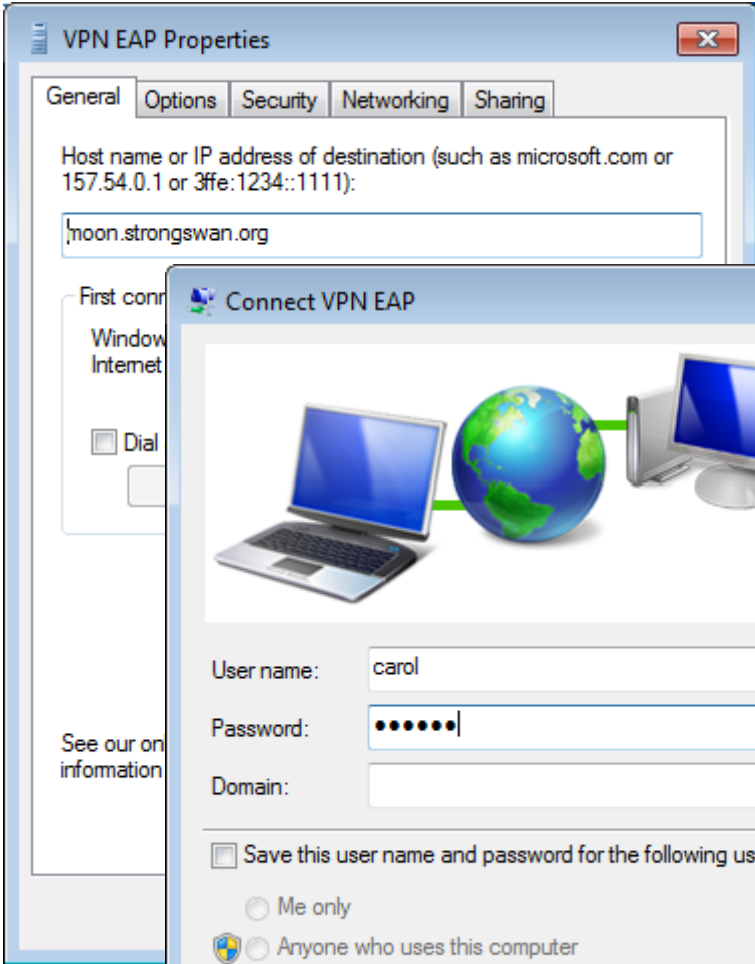
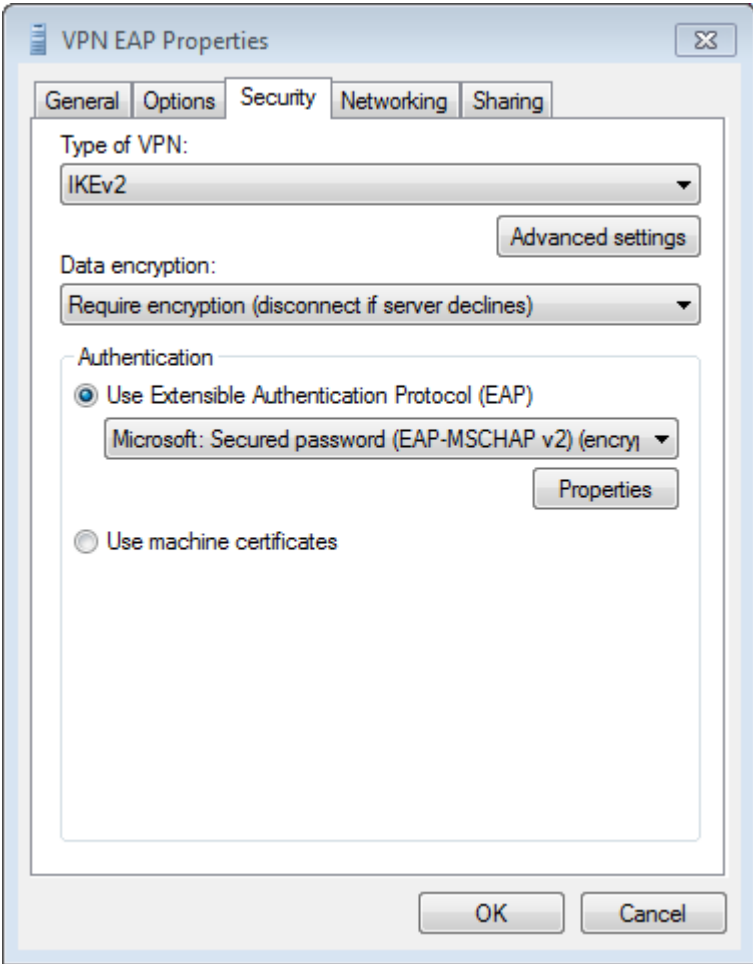
: RSA moonKey.pem

carol : EAP "tuxmux"
dave  : EAP "grummel"
```

```
# ipsec.conf for gateway moon

conn rw
    keyexchange=ikev2
    ike=aes128-aes256-sha256-sha1-modp2048-modp1024!
    esp=aes128-aes256-sha1!
    left=192.168.0.1
    leftsubnet=10.1.0.0/16
    leftcert=moonCert.pem
    leftid=@moon.strongswan.org
    leftauth=pubkey
    leftfirewall=yes
    right=%any
    rightsendcert=never
    rightsourceip=10.3.0.0/24
    rightauth=eap-mschapv2
    eap_identity=%any
    auto=add
```

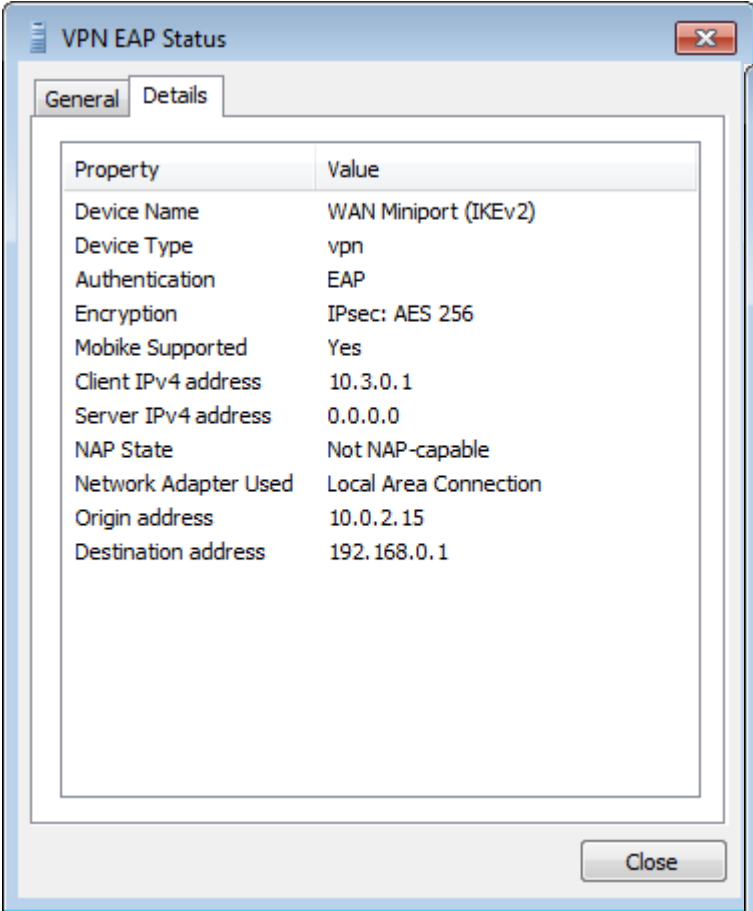
# Windows 7 VPN with EAP Authentication



- Gateway certificate must contain host name [or IP address] and the **serverAuth** extendedKeyUsage flag.

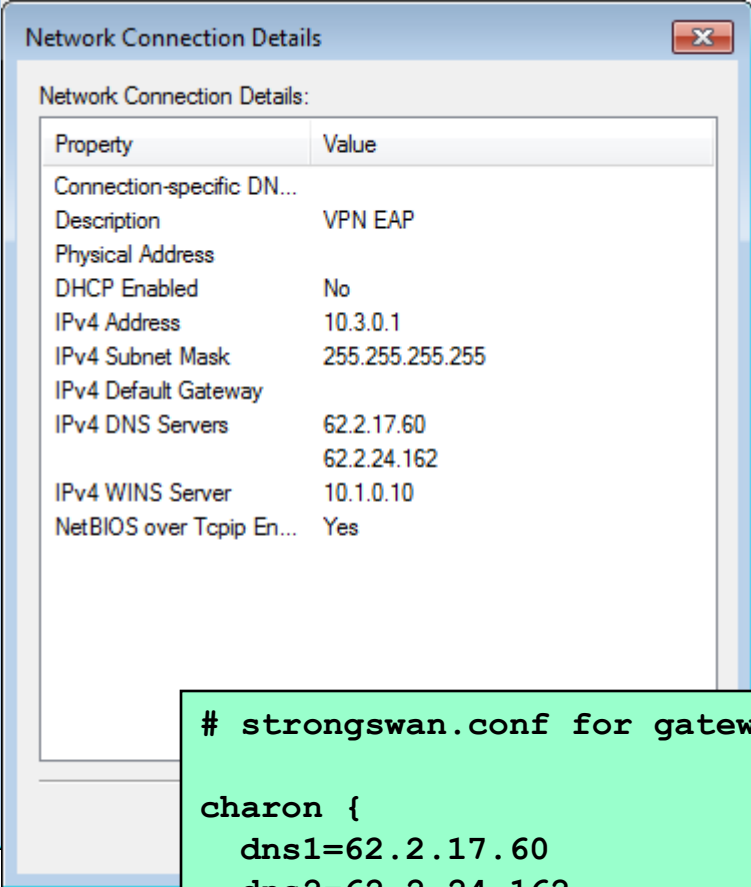


# Windows 7 VPN EAP Status



The screenshot shows the 'VPN EAP Status' dialog box with the 'General' tab selected. It displays a table of properties and values for the VPN connection.

Property	Value
Device Name	WAN Miniport (IKEv2)
Device Type	vpn
Authentication	EAP
Encryption	IPsec: AES 256
Mobike Supported	Yes
Client IPv4 address	10.3.0.1
Server IPv4 address	0.0.0.0
NAP State	Not NAP-capable
Network Adapter Used	Local Area Connection
Origin address	10.0.2.15
Destination address	192.168.0.1



The screenshot shows the 'Network Connection Details' dialog box for the VPN EAP connection. It displays a table of network properties and values.

Property	Value
Connection-specific DN...	
Description	VPN EAP
Physical Address	
DHCP Enabled	No
IPv4 Address	10.3.0.1
IPv4 Subnet Mask	255.255.255.255
IPv4 Default Gateway	
IPv4 DNS Servers	62.2.17.60 62.2.24.162
IPv4 WINS Server	10.1.0.10
NetBIOS over Tcpi... En...	Yes

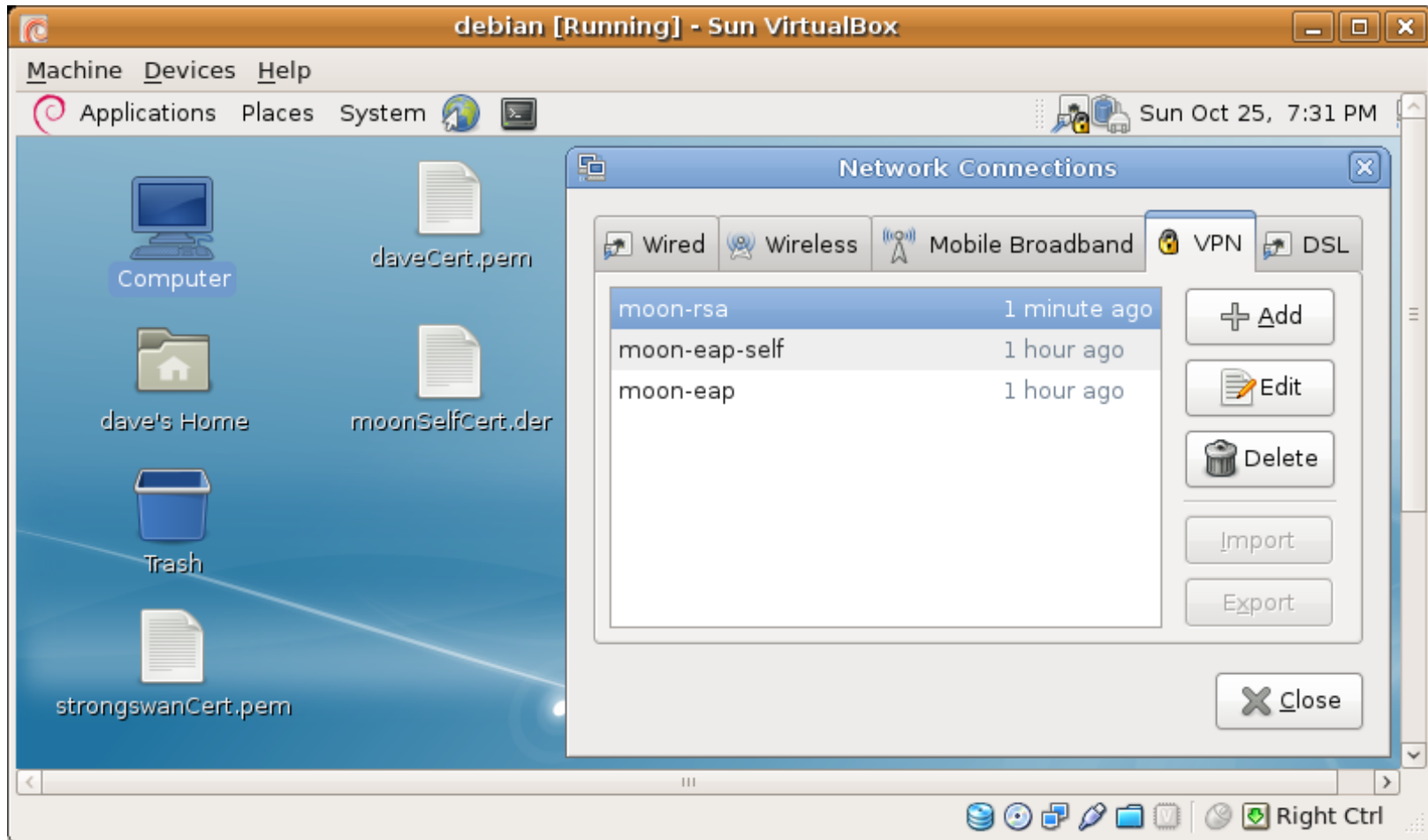
```
# strongswan.conf for gateway moon

charon {
  dns1=62.2.17.60
  dns2=62.2.24.162
  nbns1=10.1.0.10
}
```

# The strongSwan NetworkManager Plugin



# strongSwan NetworkManager VPN Plugin



- User authentication: RSA, EAP-GTC, EAP-MSCHAPv2, EAP-...
- Server authentication: Self-Signed Certificate or CA Certificate

# strongSwan NetworkManager with EAP I

Editing moon-eap

Connection name: moon-eap

Connect automatically

VPN IPv4 Settings

**Gateway**

Address: moon.strongswan.org

Certificate: strongswanCert.pem

**Client**

Authentication: EAP

Username: dave

**Options**

Request an inner IP address

Enforce UDP encapsulation

Use IP compression

Available to all users

Cancel Apply

- If a CA root certificate is specified then the hostname [or IP address] of the VPN gateway must be contained as a subjectAltName in the received gateway certificate.

VPN password required

EAP password required to establish VPN connection:

Password: ●●●●●●●

Forget password immediately

Remember password until you logout

Remember forever

Cancel Connect

# strongSwan NetworkManager with EAP II

Editing moon-eap-self

Connection name: moon-eap-self

Connect automatically

VPN IPv4 Settings

**Gateway**

Address: 192.168.0.1

Certificate: moonSelfCert.der

**Client**

Authentication: EAP

Username: dave

**Options**

Request an inner IP address

Enforce UDP encapsulation

Use IP compression

Available to all users

Cancel Apply

- As an alternative the self-signed certificate of the VPN gateway can be directly imported.

VPN password required

EAP password required to establish VPN connection:

Password: ●●●●●●●

Forget password immediately

Remember password until you logout

Remember forever

Cancel Connect

# strongSwan NetworkManager with RSA

Editing moon-rsa

Connection name: moon-rsa

Connect automatically

VPN IPv4 Settings

**Gateway**

Address: moon.strongswan.org

Certificate: strongswanCert.pem

**Client**

Authentication: Certificate/ssh-agent

Certificate: daveCert.pem

**Options**

Request an inner IP address

Enforce UDP encapsulation

Use IP compression

Available to all users

Cancel Apply

- The private RSA key stored in `.ssh/id_rsa` in PKCS#1 PEM format is managed by the `ssh-agent` and can be directly by strongSwan via the `agent` plugin.

# Connection status on gateway moon

```
Virtual IP pools (size/online/offline):
  rw: 255/2/0
Listening IP addresses:
Connections:
  rw: 192.168.0.1...%any
  rw: local: [moon.strongswan.org] uses public key authentication
  rw: cert: "C=CH, O=Linux strongSwan, CN=moon.strongswan.org"
  rw: remote: [%any] uses EAP_MSCHAPV2 authentication with EAP identity '%any'
  rw: child: 10.1.0.0/16 === dynamic
Security Associations:
  rw[1]: ESTABLISHED 5 minutes ago, 192.168.0.1[moon.strongswan.org]...
          192.168.0.254[10.0.2.15]
  rw[1]: IKE SPIs: 6d64603959c40c35_i cedb9920fa698283_r*,
  rw[1]: IKE proposal: AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
  rw{1}: INSTALLED, TUNNEL, ESP in UDP SPIs: c07e2ac6_i 2566d5f3_o
  rw{1}: AES_CBC_256/HMAC_SHA1_96, 480 bytes_i (53s ago), 480 bytes_o (53s ago)
  rw{1}: 10.1.0.0/16 === 10.3.0.1/32
  rw[3]: ESTABLISHED 20 seconds ago, 192.168.0.1[moon.strongswan.org]...
          192.168.0.254[dave]
  rw[3]: IKE SPIs: d090764d9d84fa0e_i f80f74f0e109e453_r*,
  rw[3]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_2048
  rw{2}: INSTALLED, TUNNEL, ESP in UDP SPIs: c73ddb5f5_i c60375cc_o
  rw{2}: AES_CBC_128/HMAC_SHA1_96, 840 bytes_i (1s ago), 840 bytes_o (1s ago),
  rw{2}: 10.1.0.0/16 === 10.3.0.2/32
```

# EAP-Radius based Authentication



# RADIUS Server Configuration

## moon

```
# strongswan.conf of gateway moon
charon {
  plugins {
    eap-radius {
      secret = gv6URkSs
      server = 10.1.0.10
    }
  }
}
```

```
# ipsec.conf of gateway moon
conn rw-eap
  left=192.168.0.1
  leftsubnet=10.1.0.0/16
  leftid=@moon.strongswan.org
  leftcert=moonCert.pem
  leftauth=pubkey
  leftfirewall=yes
  right=%any
  rightsendcert=never
  rightsourceip=10.3.0.0/24
  rightauth=eap-radius
  eap_identity=%any
  auto=add
```

## radius server

```
# /etc/raddb/clients.conf
client 10.1.0.1 {
  secret      = gv6URkSs
  shortname = moon
```

```
# /etc/raddb/eap.conf
eap {
  default_eap_type = md5
  md5 {
  }
}
```

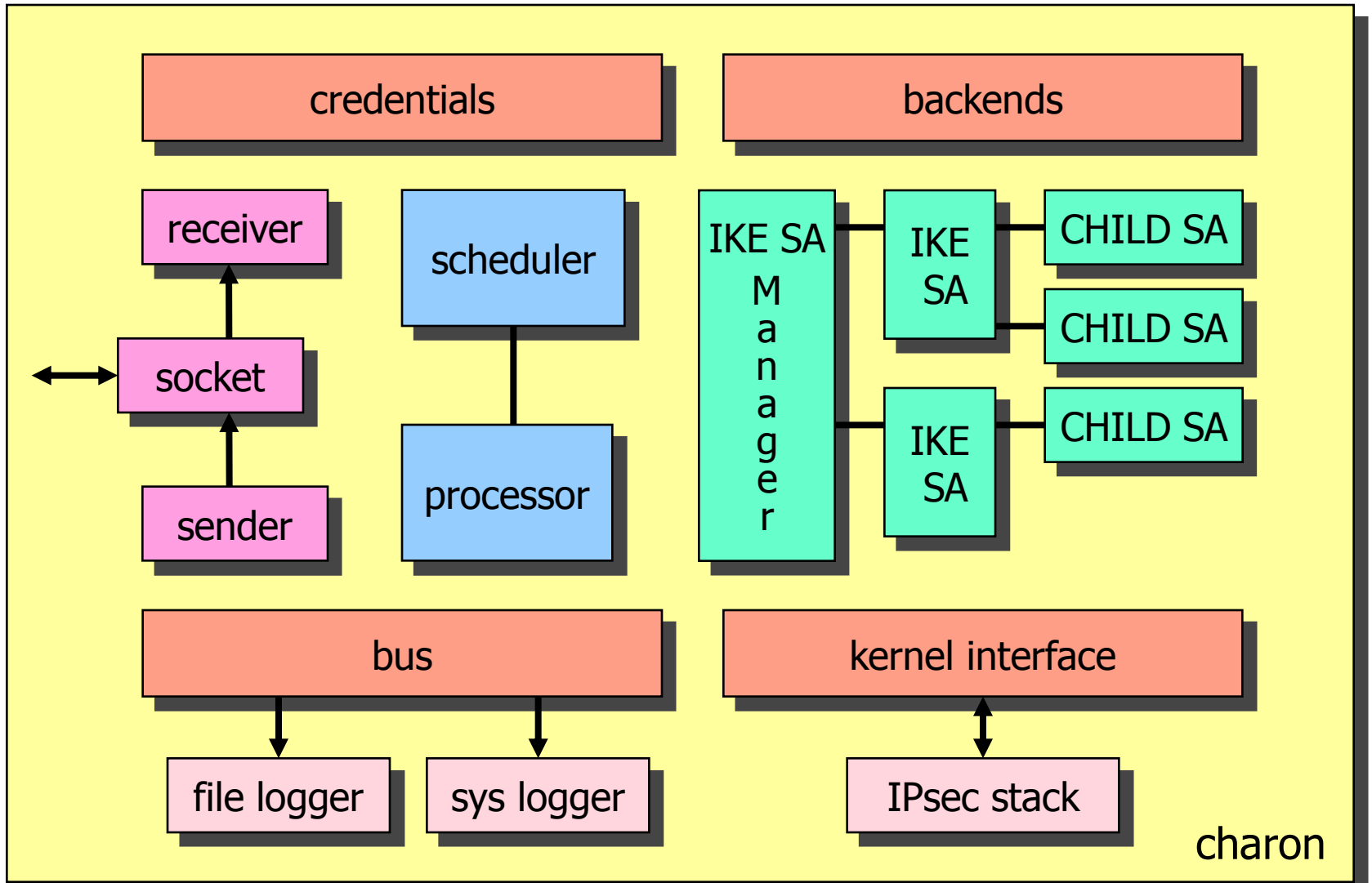
```
# /etc/raddb/proxy.conf
realm LOCAL {
  type      = radius
  authhost = LOCAL
  accthost = LOCAL
}
```

```
# /etc/raddb/users
carol  Cleartext-Password := "tuxmux"
dave   Cleartext-Password := "grummel"
```

# strongSwan Software Architecture

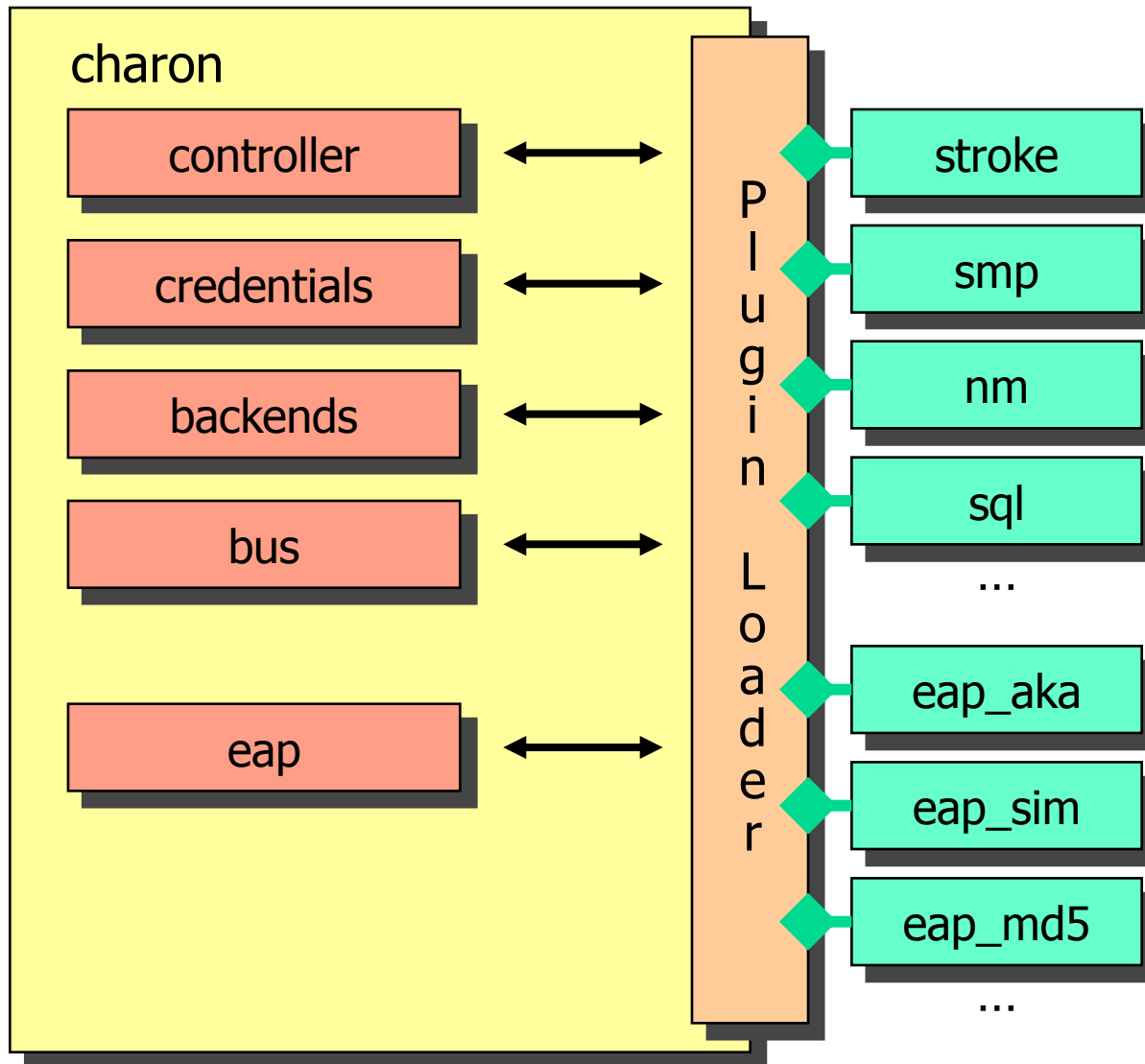


# IKEv2 Daemon – Software Architecture



16 concurrent worker threads

# Plugins for charon



- smp  
XML-based control and management protocol.  
Implementation: [strongSwan Manager](#)
- nm  
DBUS-based plugin for [NetworkManager](#)
- sql  
Generic SQL interface for configurations, credentials & logging.  
Implementations: [SQLite & MySQL](#)
- eap\_x  
Any EAP protocol.

# strongSwan Manager

The screenshot shows the 'strongSwan Manager' web interface in a Mozilla Firefox browser window. The page title is 'IKE SA overview'. The main content area displays details for an IKE SA and its associated IPsec SAs.

**IKE SA #82:** hsr-net [IKE #82]: asteffen@hsr.ch <-> sidv0150.hsr.ch

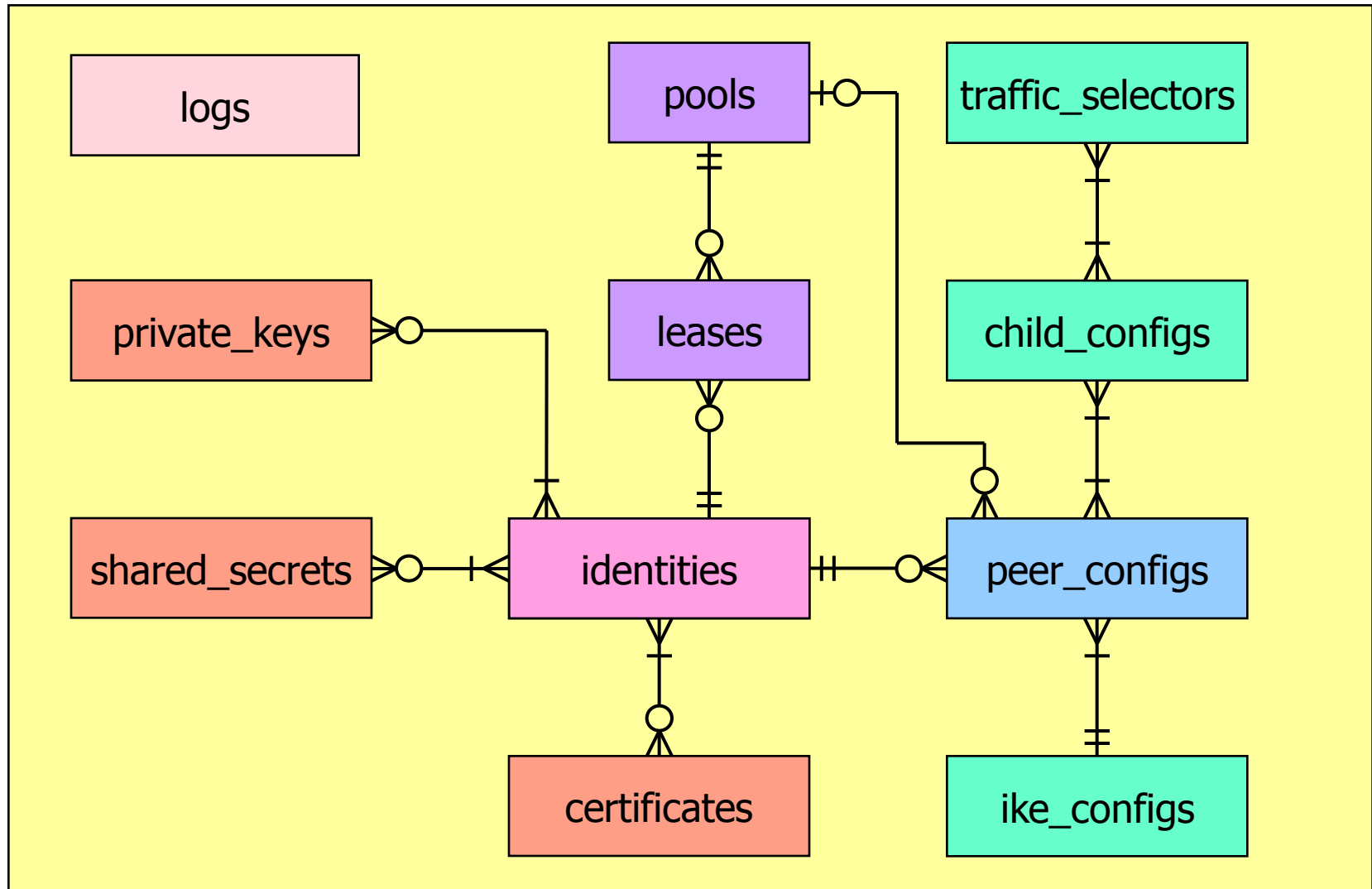
**IPsec #164:** hsr-net [IPsec #164]:

Local Network	Local SPI	Direction	Remote SPI	Remote Network
10.10.0.0/23	cf6ae6b3	<-	cfbd1f88	152.96.52.150/32
62.2.17.60/31	c4c0d2a5	<-	cd3586ed	152.96.52.150/32

Red annotations on the right side of the screenshot indicate actions to be taken:

- A red circle with a red 'X' over it is placed over the IKE SA header, with a red arrow pointing to the text 'take down IKE SA'.
- A red circle with a red 'X' over it is placed over the IPsec SA header, with a red arrow pointing to the text 'take down IPsec SA'.

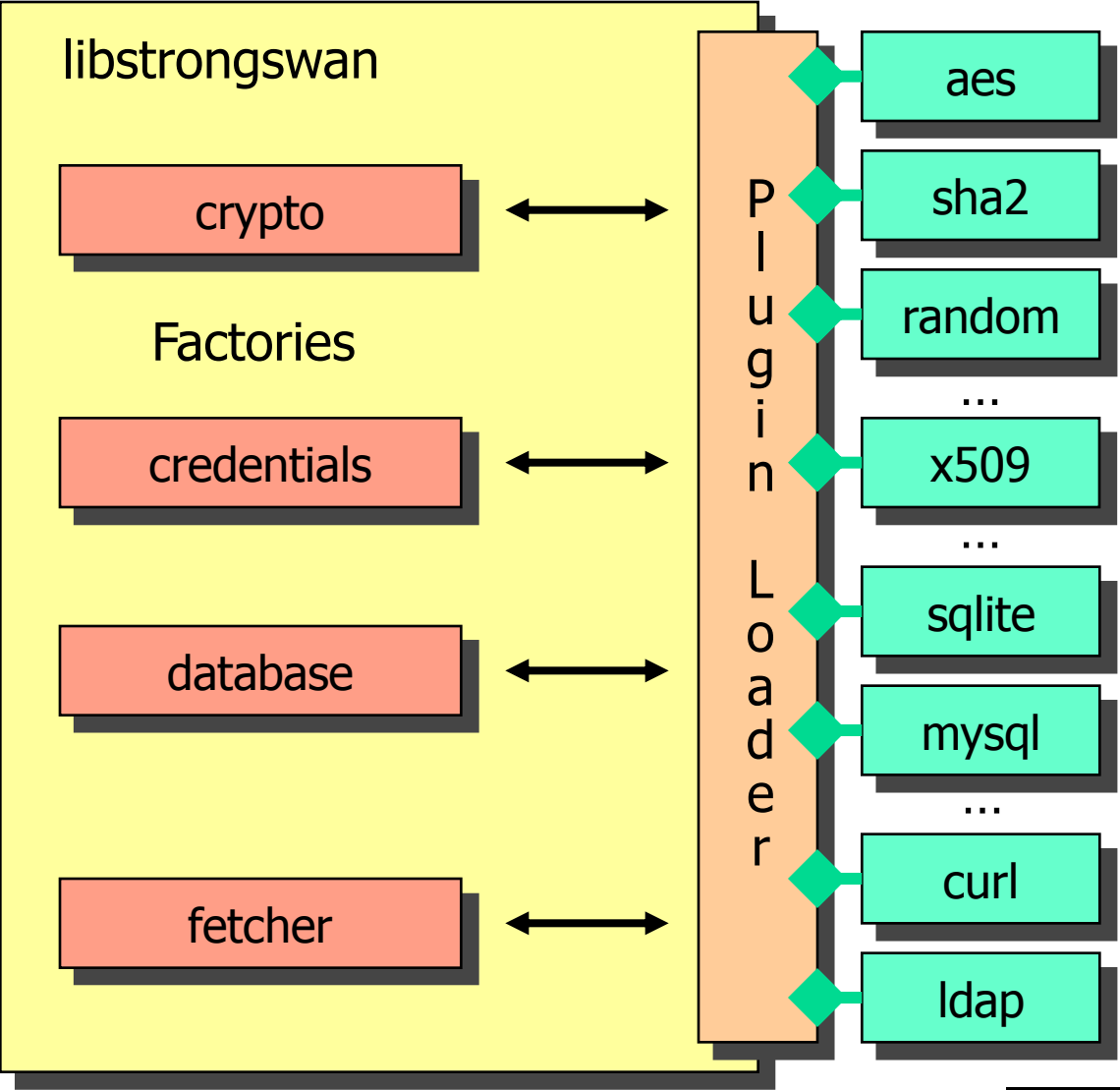
# strongSwan Entity Relationship Diagram



SQLite and MySQL implementations

# Cryptographic Plugins

# Plugins for libstrongswan

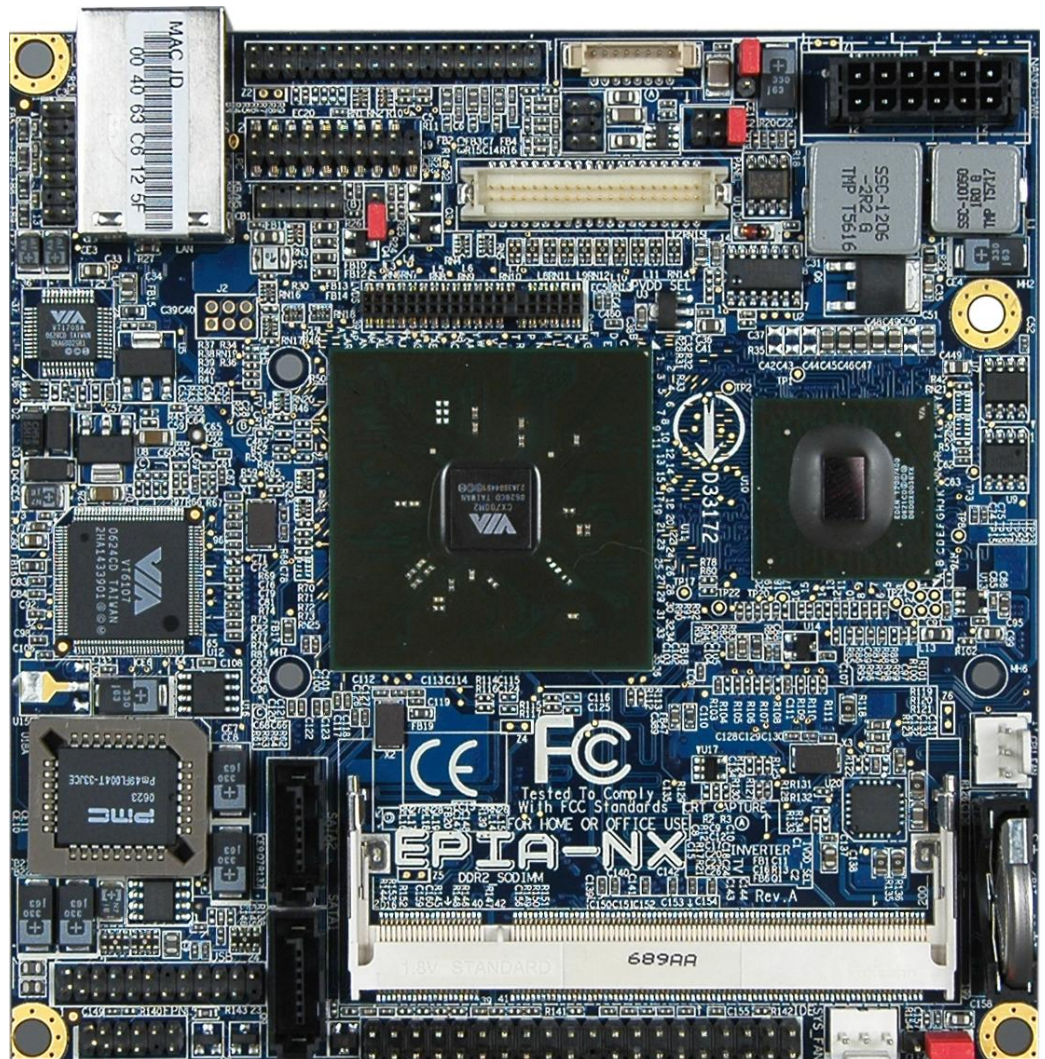


- Non-US crypto code
- No OpenSSL library
- ECCN: No License Required (NLR)
  
- Certificate retrieval (HASH-and-URL)
- CRL fetching, OCSP



# VIA EPIA-NX PadLock Crypto-Processor

- **padlock** plugin  
AES/SHA  
HW acceleration
- **openssl** plugin  
uses libcrypto-0.9.8  
OpenSSL library
  - ECP DH groups
  - ECDSA signatures
  - HW engine support



# The strongSwan PKI function

```
ipsec pki --gen --type ecdsa --size 521 > strongswanKey.der
ipsec pki --self --in strongswanKey.der --type ecdsa --lifetime 3650
--dn "C=CH, O=strongSwan, CN=strongSwan EC CA"
--ca --digest sha512 > strongswanCert.der

ipsec pki --gen --type ecdsa --size 384 > moonKey.der
ipsec pki --req --in moonKey.der --type ecdsa --digest sha384
--dn "C=CH, O=strongSwan, CN=moon.strongswan.org"
--san moon.strongswan.org > moonReq.der

ipsec pki --gen --type ecdsa --size 256 > carolKey.der
ipsec pki --req --in carolKey.der --type ecdsa --digest sha256
--dn "C=CH, O=strongSwan, CN=carol@strongswan.org"
--san carol@strongswan.org > carolReq.der

cat pki.opt
--type pkcs10 --lifetime 1825 --crl http://crl.strongswan.org/ecdsa.crl
--cacert strongswanCert.der --cakey strongswanKey.der --digest sha512

ipsec pki --issue --options pki.opt --in moonReq.der --flag serverAuth
--serial 01 > moonCert.der
ipsec pki --issue --options pki.opt --in carolReq.der
--serial 02 > carolCert.der
```

# Suite B offers constant 128/192 Bit Security

```
# ipsec.secrets for gateway moon
: ECDSA moonKey.der
```

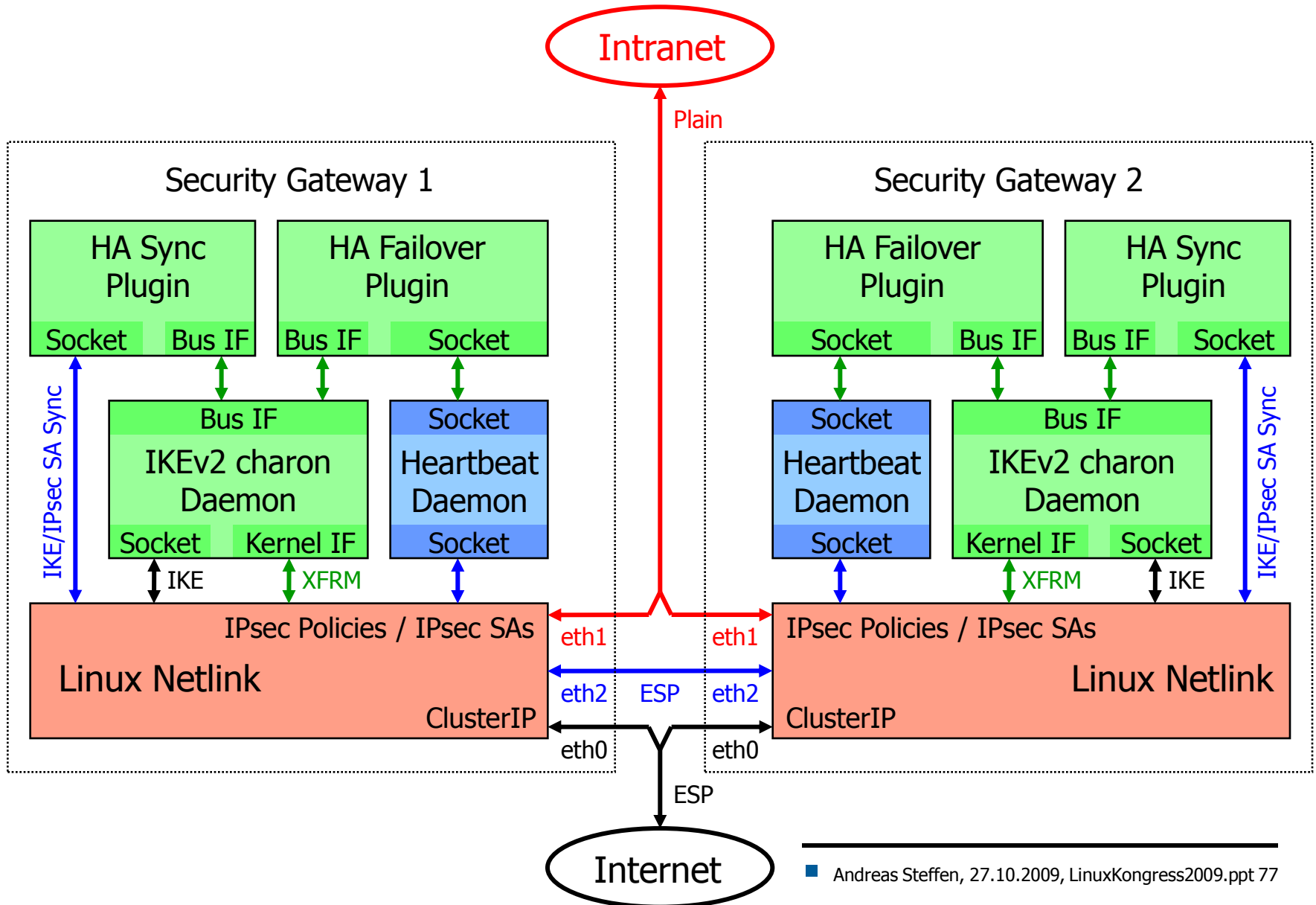
```
# ipsec.conf for gateway moon
conn rw
    keyexchange=ikev2
    ike=aes256-sha384-ecp384,aes128-sha256-ecp256!
    esp=aes256gcm16,aes128gcm16!
    leftsubnet=10.1.0.0/24
    leftcert=moonCert.der
    leftid=@moon.strongswan.org
    right=%any
    rightsourceip=10.3.0.0/24
    auto=add
```

```
rw[1]: ESTABLISHED 9 seconds ago, 192.168.0.1[moon.strongswan.org]...
      192.168.0.100[carol@strongswan.org]
rw[1]: IKE SPIs: 7c1dcd22a8266a3b_i 12bc51bc21994cdc_r*,
rw[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
rw{1}:  INSTALLED, TUNNEL, ESP SPIs: c05d34cd_i c9f09b38_o
rw{1}:  AES_GCM_16_128, 84 bytes_i (6s ago), 84 bytes_o (6s ago),
rw{1}:  10.1.0.0/24 === 10.3.0.1/32
```

- 128 bit security requires 3072 bit RSA keys and DH groups!
- In 2005 NSA proposes use of efficient elliptic curve cryptography.
- Suite B use for IPsec defined in RFC 4869.

# High Availability using Cluster IP

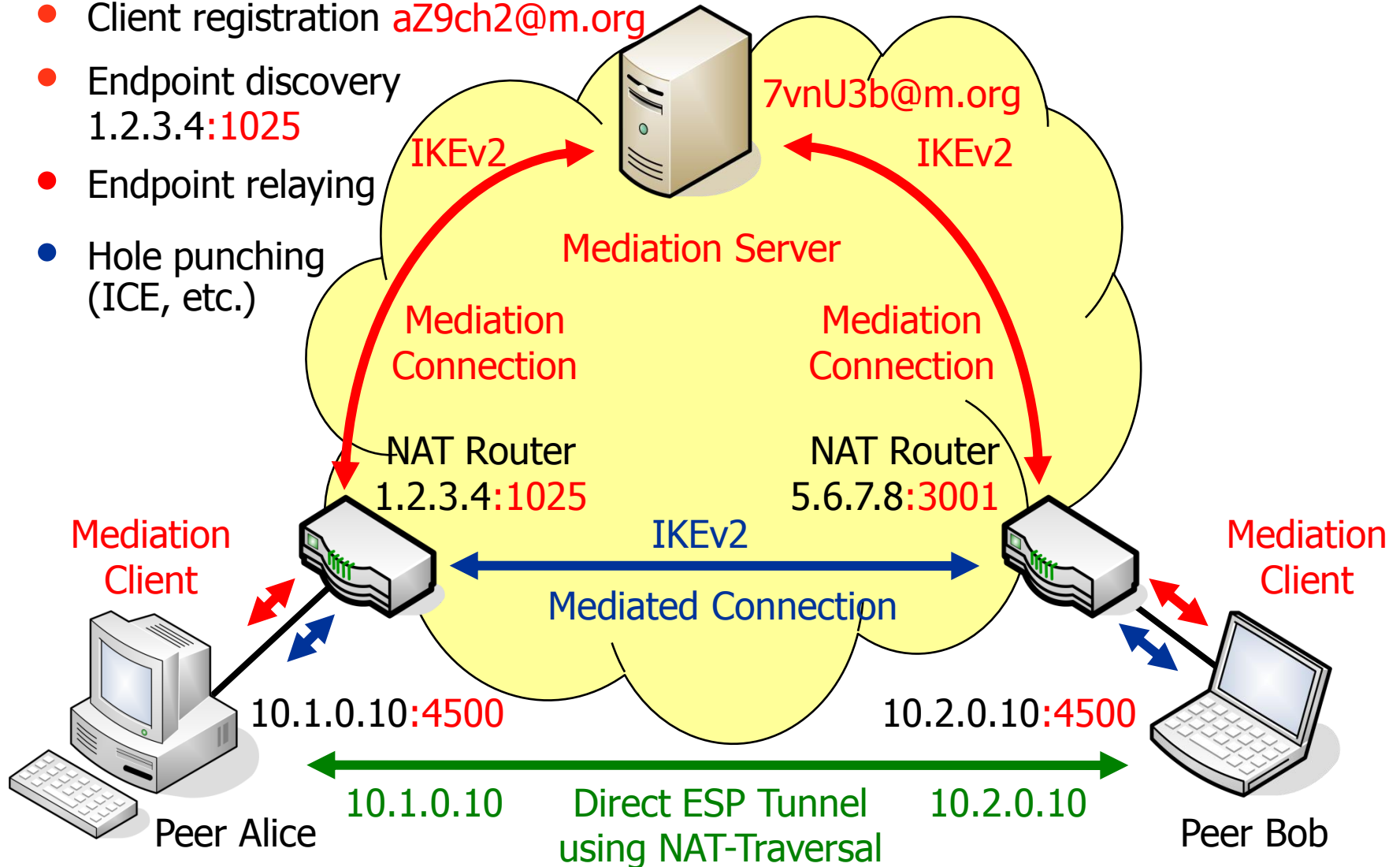
# strongSwan High-Availability Architecture



# IKEv2 Mediation Extension

# Peer-to-Peer NAT-Traversal for IPsec

- Client registration **aZ9ch2@m.org**
- Endpoint discovery **1.2.3.4:1025**
- Endpoint relaying
- Hole punching (ICE, etc.)



# draft-brunner-ikev2-mediation released

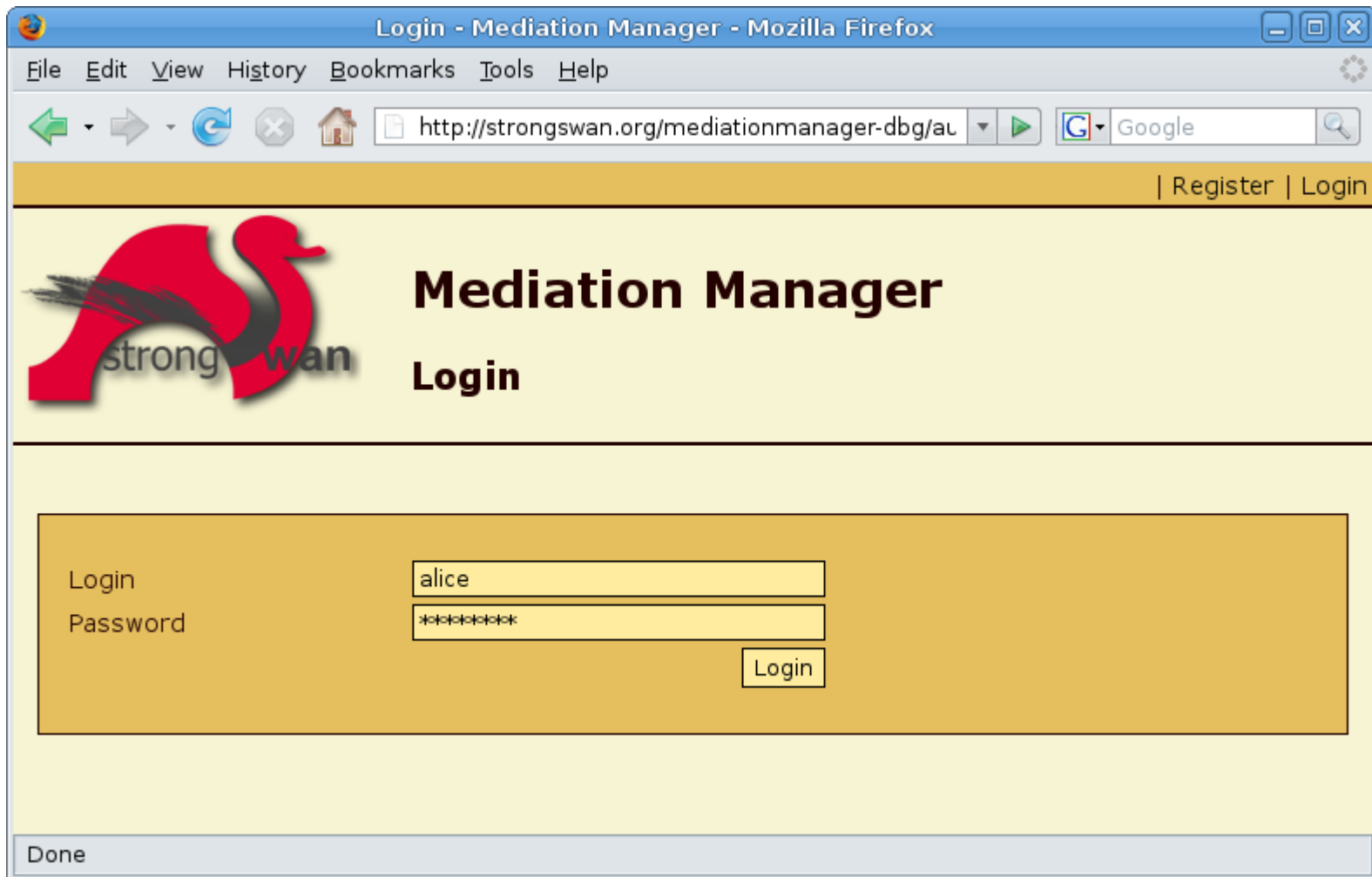
The screenshot shows a Mozilla Firefox browser window displaying the 'The Daily Dose of IETF' website. The page title is 'The Daily Dose of IETF' with the subtitle 'Be conservative in what you send and liberal in what you accept'. The issue is identified as 'Issue 605 — 2008-04-17'. The main content is organized into several sections:

- IETF-Announce List:** Contains three entries related to PCEP, BRPC, and RFC Errata.
- Drafts Sent to IESG:** Lists several drafts including Bidirectional Forwarding Detection, Generic Application of BFD, BFD For MPLS LSPs, BFD for Multihop Paths, and BFD for IPv4 and IPv6 (Single Hop).
- IESG Progress:** Lists drafts such as Contexts for IMAP4, Sieve Email Filtering: Environment Extension, EAP Tunneled TLS Authentication Protocol Version 0, IMAP CONVERT extension, NFS Direct Data Placement, and Remote Direct Memory Access Transport for Remote Procedure Call.
- New RFCs:** A section for recently published RFCs.
- New and Revived Drafts:** Features the draft-brunner-ikev2-mediation entry, which is highlighted in a yellow box. The entry includes the author 'Tobias Brunner', the date '16-Apr-08', and links for 'TXT', 'HTML', and 'PDF'. The abstract describes the IKEv2 Mediation Extension (IKE-ME) as a connectivity extension to the Internet Key Exchange (IKEv2) that allows two peers, each behind one or more Network Address Translators (NATs) or firewalls, to establish a direct and secure connection without the need to configure any of the intermediate network devices.

The left sidebar contains navigation links for IETF Home, About Tools, Tools (diffs, spell, xml2rfc, nits), News, Get Passwd, IETF-71 (Rooms, Agenda, Calendar), Documents, RFCs, Doc fetch, Wikis (IESG, IRTF, IAOC, Trust, Chairs, Edu, Tools, BOFs), NomCom, Areas, WGs (lóng, ólowpan, óman, Adslmb, Ancp, Autoconf, Avt, Behave, Bfd), and Done.



# Login at the strongSwan Mediation Manager



The screenshot shows a Mozilla Firefox browser window titled "Login - Mediation Manager - Mozilla Firefox". The address bar contains the URL "http://strongswan.org/mediationmanager-dbg/au". The page features a yellow header with navigation links for "Register" and "Login". Below the header is the strongSwan logo, which consists of a red swan silhouette and the text "strong swan". To the right of the logo, the text "Mediation Manager" and "Login" is displayed in a large, bold font. The main content area contains a login form with two input fields: "Login" (containing the text "alice") and "Password" (containing "\*\*\*\*\*"). A "Login" button is positioned to the right of the password field. The browser's status bar at the bottom shows "Done".

# Register a Peer with the Mediation Manager

File Edit View History Bookmarks Tools Help

http://strongswan.org/mediationmanager-dbg/pe Google

Peers | Edit Account | Logout

## Mediation Manager

### Add Peer

Alias	<input type="text" value="bob"/>
Public Key	<pre>-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIBCgKCAQEArjMFJ0w89iQRkW9uHfTA aUVCnsWR3cU5N6zWMkgauZW5881l2efU3sxpjNPia92PYPUoC44yEfwtlXQRWZVD qPqjRkVDM5Br0qEsByYcsKH8fk6YqUlsGT8SZf9SVuh2br9MCzYFgWLqhVqmRanG eDJVl02gQh4dWLIIEQ/ef21uD4oaeC7yqWdP74mTvuywSEwT/AwxSbGAe6jE42cK 8o83xdsnIeCCLJbFUDrJwdV85o7mLyYDiNQHfeDhrAVR3l7ghjvbq21yPzpZwHqL Cd\lONforIuLW7xLI8ZsIfuZWBToNzyTA8qIIBA8qgaYasepU5hys1WuZgMXrvoFq OwIDAQAB -----END PUBLIC KEY-----</pre>

Back Add

Done

# List of Registered Peers

Peers - Mediation Manager - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://strongswan.org/mediationmanager-dbg/pe

Peers | Edit Account | Logout

## Mediation Manager

### Peers

Alias	Identifier	
alice	c8:96:95:23:4b:62:d4:fd:8a:e7:9e:e1:58:82:e5:ea:34:6f:76:61	 
bob	c0:9b:84:91:f8:d8:d4:d4:39:b2:61:81:e3:04:20:7d:41:19:21:a7	 

Add Peer

Done