Network Security 17: IPsec

i.g.batten@bham.ac.uk

IPsec

- IP Security
- Provides mechanisms for encrypting and authenticating traffic between hosts and between networks
  - Transformation is packet by packet and operates on any IP packet, so works for UDP, TCP, ICMP, OSPF…
- Developed as part of IPv6: original statement was IPv6 implementations MUST support IPsec
- RFC6434 changes that to SHOULD, because plenty of IPv6 applications don’t need / can’t use IPsec and requirement for IPsec was (another) perceived barrier to deployment.
- IPsec also back-ported to IPv4 and ships on most IP stacks.

Structure

- First session: what happens to packets
- Second session: automatic keying
- Third session: a walk through a real (well, my) network, with three different, latest bits implementations, lots of cipher suites, certificates, all mod cons.

RFC6434

Delicious understatement

A range of security technologies and approaches proliferate today (e.g., IPsec, Transport Layer Security (TLS), Secure SHell (SSH), etc.) No one approach has emerged as an ideal technology for all needs and environments. **Moreover, IPsec is not viewed as the ideal security technology** in all cases and is unlikely to displace the others.

IPsec Concepts

- Two modes of encapsulation: tunnel and transport
- Two modes of use: AH (authentication), ESP (encryption) (which can also be combined for bonus complexity)
- Two mechanisms for keying: static and dynamic (IKE).
- Lots of options for cipher suites, key lengths, extra features and chocolate sprinkles
    - Lots of opportunities for implementations or configurations which don’t interwork for subtle and difficult to fix reasons.
    - For extra fun, in some cases will agree keys, establish relationship and then pass no traffic as (for example) hash functions are being used slightly differently so all packets fail to verify.

IPsec Layers

- Data plane transformation of packets has to happen at wire-speed or near-as, so focuses on performance.
- Management layer (key exchange) much less demanding.
- On computers, transformation in kernel or hardware accelerator, key exchange in user-space.
- On routers and other network elements, transformation in hardware or in network processor, key exchange on management card/processor.
- Result: even more complexity, as the cipher suites available are different, as are the implementations (and, if you are doing this seriously, the threat models are different as well).

IPsec Problems

- Implementation extremely complex
- Lots of options, some conflicting
- Key management issues
- Cynics ask if it was deliberately made too complex to analyse properly

IPsec Reassurance

- IPsec is accredited for use in classified environments up to the highest, subject to the underlying cipher suites and key management being appropriate and correct management
    - IPsec itself is not the weakest link: simple crypto constructions using well-proven building blocks
- But key exchange is complex and difficult to get right, so **practical** attacks on IPsec probably exploit this
- Manual keying used by the nervous: you can use your fancy spooky RNG to generate 256 bits and use a man with a briefcase, a chain and a gun to move it to the other end. Lots of tamper-resistant/evident hardware needed.

Cipher Suites

- For IPsec to work, you need (assuming we are encrypting) to agree:
  - A bulk cipher (today AES, historically 3DES, other options available but may not work) in a mode (CBC, CTR for pipelining, GCM to include authentication)
  - A hash function (today ideally SHA256 but often SHA1 owing to poor standardisation, historically MD5 or assorted MAC constructions)
    - Standard truncates SHA256 to 128 bits, but implementations also do 96 and 160, which then don’t interwork.
  - AES-CBC(128..256)+SHA1 pretty much guaranteed to work, everything else something of a lottery.

Cipher Suites

- Automated key exchange also requires a key establishment protocol (DH) plus some means to do mutual authentications (shared secrets, X509 certificates) plus optionally some means to do PFS (DH again)
- Break this and you have access to the keys used for the bulk ciphering.
- Complex, tricky to implement correctly, hard to assure.

Implementations

- Writing the lower layers (packet processing) fiddly rather than difficult, and intimately tied into the network processing portion of your operating system or hardware.
- Therefore most implementations specific to the OS involved.
    - Two free stacks for Linux, KAME and FreeS/WAN
- Userland not difficult to write, but hard to get right

Linux History

- Two projects: KAME (Japanese research consortium) and FreeS/WAN (John Gilmore et al).
- KAME kernel drivers merged into mainline kernel
- FreeS/WAN needed out-of-tree kernel modules
- Later work standardised kernel management interface, so FreeS/WAN userland can use standard kernel (ex-KAME) drivers
- KAME not active, but IKE daemon (“racoon”) used in OS/X. “ipsec-tools” package for most Linuxes available but deprecated.
- FreeS/WAN (dead) forked to OpenS/Wan (almost dead) and StrongS/WAN (main player now); LibreS/WAN is fork of OpenS/Wan with different priorities. IKE daemon pluto (older) or charon (more modern).
- Interworking is painful but possible.

IPsec modes

- AH (authentication header): Provides integrity of payload and immutable parts of header.
- ESP (encapsulated security payload): Provides encryption and optionally payload integrity.
- AH+ESP: provides full integrity and encryption, but tending to be dropped from modern code in favour of other mechanisms that do the same basic job (StrongS/WAN “AH+ESP bundles are not supported.”)

AH v ESP

- **AH: Authentication Headers**
  - Keyed hash of the whole packet and the immutable parts of the header
  - Ensures packet has not been changed in transit, and comes where it claims to have come from

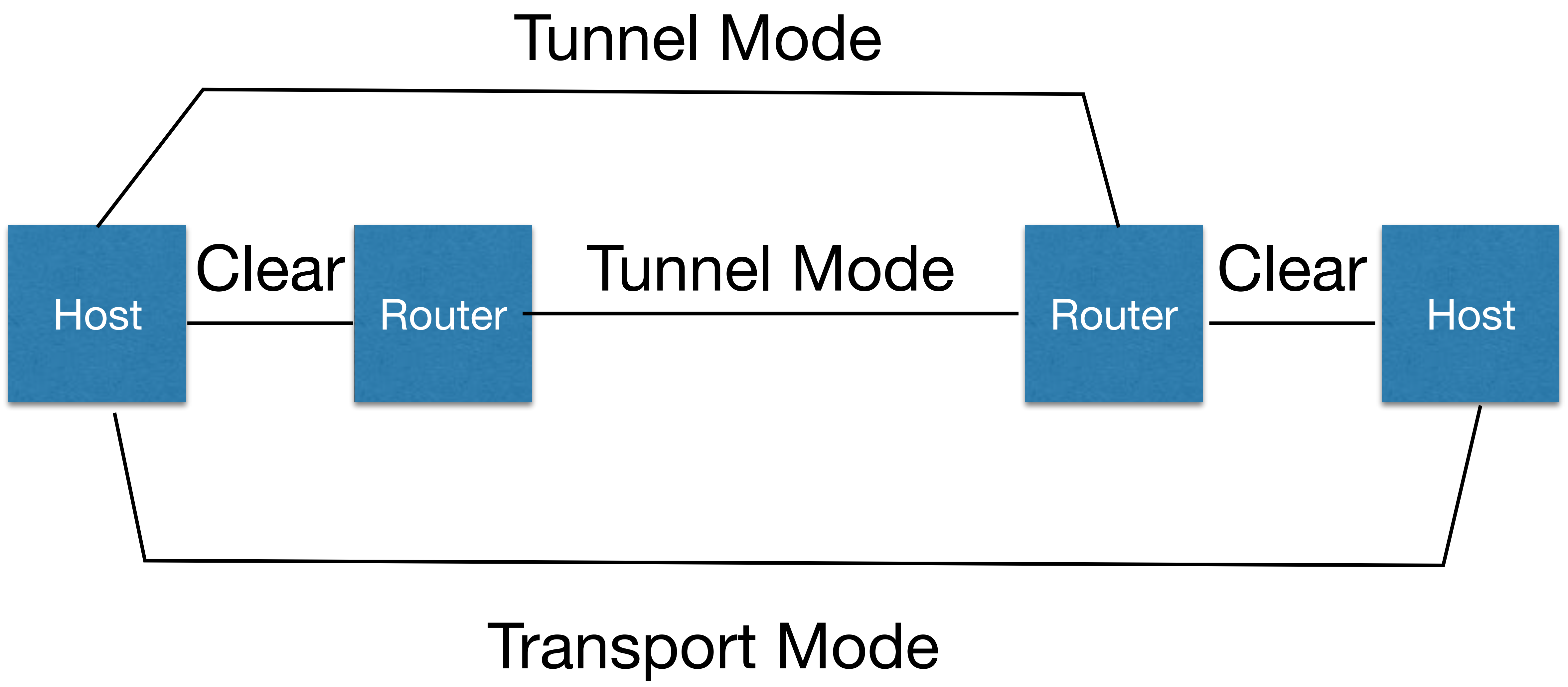
AH v ESP

- **ESP: Encapsulated Security Payload**
- Encrypts some or all of the packet so that it is confidential
    - Authentication optional, but strongly recommended.
    - Provides Confidentiality, and Integrity if used sensibly

Transport v Tunnel

- Transport mode is for communication between devices which themselves speak IPsec: packets are transformed directly.
- Tunnel mode is for communication between routers carrying traffic from other end points: packets are encapsulated (IPIP) and then transformed, so IP header is transformed as well.

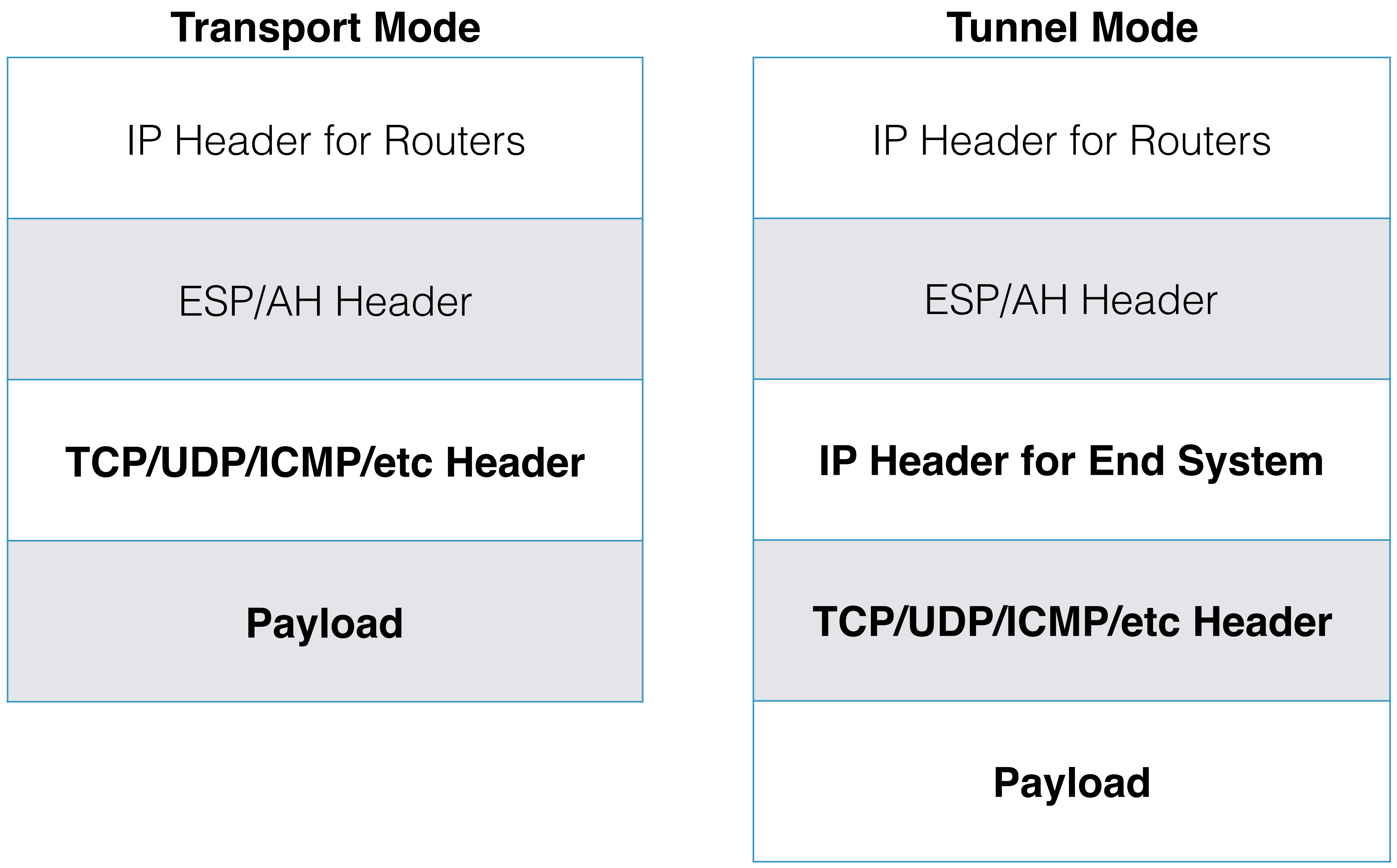
Transport v Tunnel



7:48
1
⌀ --- mpg
❄ -5.0 °C
miles
trip
133075
380.3
P R N D S 1

4
5

IPsec structure



Security Parameters (SPI)

- Defines the **algorithms** and **keys** in use for a **security association**
- SPI is (usually) chosen at random: chance of collision assumed to be vanishingly small
  - You can set them manually in some implementation: choose wrong just doesn’t work.
- Each packet is encrypted, decrypted, signed or verified by looking up the SPI in a table

Security Parameters

```
src 2a04:92c7:e:5db::965d dst 2a00:1630:66:25:f585:a218:7189:22b
  proto esp spi 0xc49393c8 reqid 518 mode tunnel
  replay-window 32 flag af-unspec
  auth-trunc hmac(sha1) 0x16e9338ae105dab513c250a8b281948413dc5728 96
  enc cbc(aes) 0x18436568ebf6af3d02fd70ab70485149
src 2001:8b0:1111:1111:0:ffff:5102:4fdc dst 2a04:92c7:e:5db::965d
  proto esp spi 0xc6137585 reqid 521 mode tunnel
  replay-window 32 flag af-unspec
  auth-trunc hmac(sha256) 0xf67e61cbc06…lots chopped…15b57 128
  enc cbc(aes) 0xfbf4e11a8c8c952b3db136eae1c8db25
src 2a04:92c7:e:5db::965d dst 2001:8b0:129f:a90e:3141:5926:5359:3
  proto esp spi 0xc88763fa reqid 517 mode tunnel
  replay-window 32 flag af-unspec
  aead rfc4106(gcm(aes)) 0x16ef97036f5c64b06644b1a6be2afdd9f7ec1996 128
```

- Three SPIs, AES+SHA1, AES+SHA256, AES-GCM.
- Note different truncations (96, 128), and that AES-GCM doesn’t need separate authentication.
- Note I have chopped section out of the (much larger) SHA256 secret

AH (protocol 51)

Offsets	Octet$_{16}$	0								1								2								3							
Octet$_{16}$	Bit$_{10}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Next Header*								*Payload Len*								*Reserved*															
4	32	*Security Parameters Index (SPI)*																															
8	64	*Sequence Number*																															
C	96	*Integrity Check Value (ICV)* …																															
…	…																																

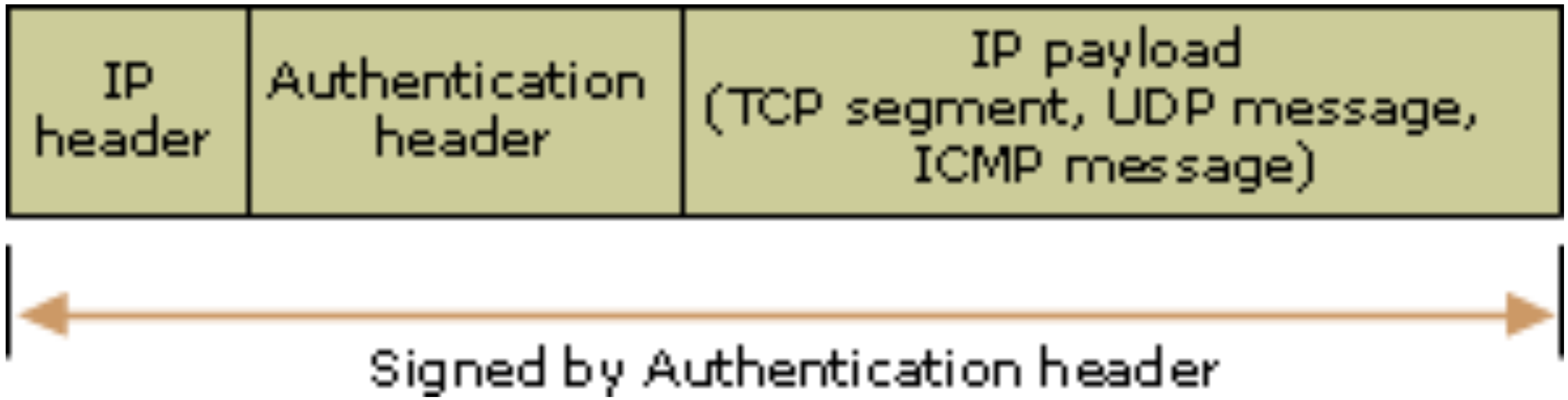
AH Headers

- Next header: protocol number of next header (usually indicating 17 UDP, 6 TCP or 1 ICMP)
- SPI: Security Parameters Index. 32 bit quantity identifying this particular IPsec relationship.
    - **NOT** the key, but a pointer to the key

ICV

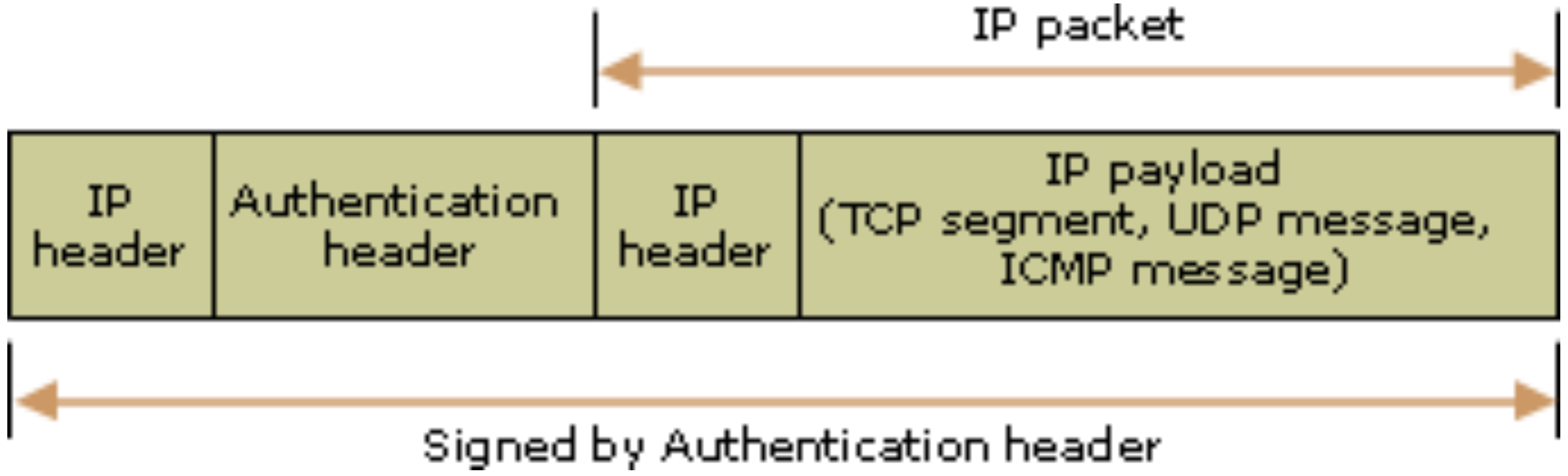
- Integrity Check Value
- A Keyed Hash, Keyed MAC or some other object that provides the assurance you need
    - Agreed “out of band” by participants for manual keying, or as part of IKE negotiation for automatic
- Causes interworking problems as truncation of hash functions often got wrong, and then other implementations need to interwork with broken code.
    - Matrix of interoperable pairs worryingly sparse for anything other than SHA1.

AH Transport


IP header
Authentication header
IP payload (TCP segment, UDP message, ICMP message)
Signed by Authentication header

IP Header is proto 51,
"Next Header" is proto 6 for TCP,
Payload is then a TCP header and data

AH Tunnel


IP packet
IP header
Authentication header
IP header
IP payload (TCP segment, UDP message, ICMP message)
Signed by Authentication header

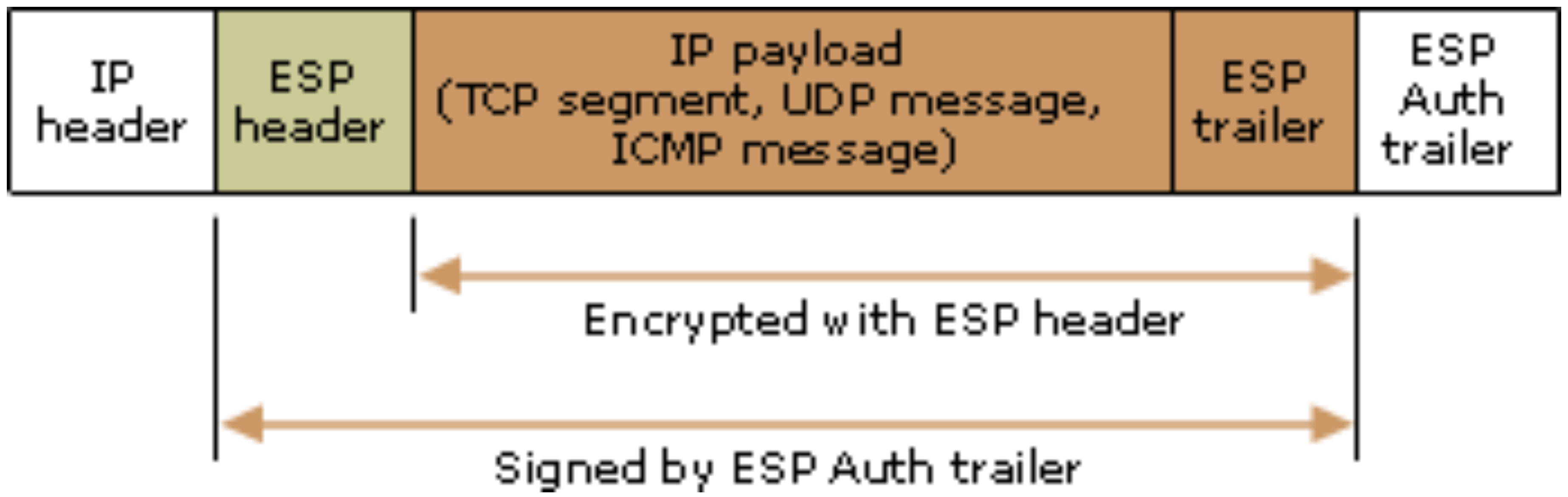
Outer IP Header is proto 51,
“Next Header” is proto 4 for IP,
Inner IP Header is proto 6 for TCP
Payload is then a TCP segment

ESP (Protocol 50)

Offsets	Octet$_{16}$	0								1								2								3							
Octet$_{16}$	Bit$_{10}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Security Parameters Index (SPI)*																															
4	32	*Sequence Number*																															
8	64	*Payload data*																															
…	…																																
…	…																																
…	…									*Padding (0-255 octets)*																							
…	…																	*Pad Length*								*Next Header*							
…	…	*Integrity Check Value (ICV)*																															
…	…	…																															

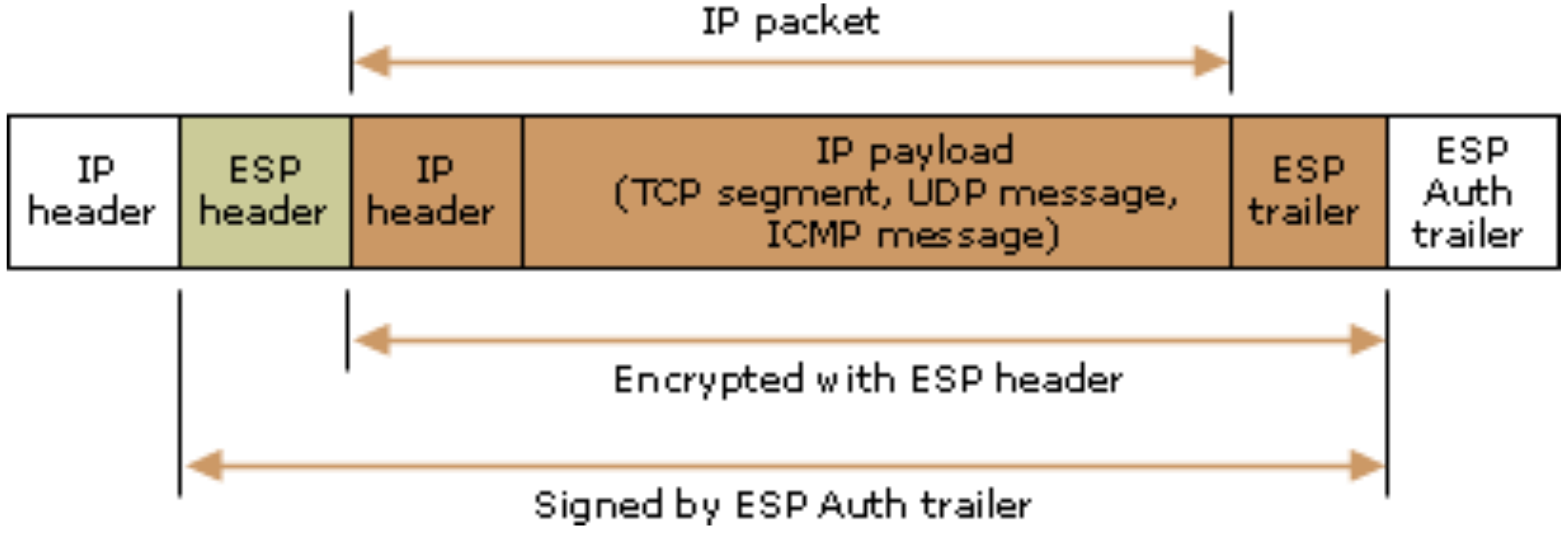
Optional “Authentication Trailer”

ESP Transport


IP header
ESP header
IP payload (TCP segment, UDP message, ICMP message)
ESP trailer
ESP Auth trailer
Encrypted with ESP header
Signed by ESP Auth trailer

IP Header is proto 50,
“Next Header” is proto 6 for TCP,
Payload is then a TCP header and data
Auth Trailer is OPTIONAL (but in reality it should always be used)

ESP Tunnel


IP packet
IP header
ESP header
IP header
IP payload (TCP segment, UDP message, ICMP message)
ESP trailer
ESP Auth trailer
Encrypted with ESP header
Signed by ESP Auth trailer

Outer IP Header is proto 50,
“Next Header” is proto 4 for IP,
Inner IP Header is proto 6 for TCP
Payload is then a TCP segment

ESP does not protect IP header

- ESP in transport mode does not protect the IP header from alteration; in particular, the source IP address can be changed undetectably.
- One choice is ESP in tunnel mode
- Alternatively, use AH and ESP at the same time, as AH includes the immutable parts of the IP header in the signature.
  - This is often not needed, as ESP provides origin assurance if you choose the keys properly.
  - No longer supported in many implementations

ESP Auth Performance

- Common cipher suites (AES+SHA1, AES+SHA256, AES in CBC or CTR mode) need two passes over the packet, one to make cipher text, one to form keyed hash for ICV.
- CTR and GCM mode popular in hardware as it means keystream can be formed in advance, and only an XOR is needed at the point of transformation.
  - CTR requires an XOR, GCM and XOR and a Galois Field multiplication.
  - Both allow the AES encryption to be done in advance, as part of a pipeline.

Sequence Numbers

- Can be used to discard old packets to prevent replay attacks, with a window based on risk tolerance and network reliability
- Sequence numbers **not** used to drive any sort of ACK or retransmission strategy, just to prevent replays.

NAT Problems

- NAT essentially uses port numbers to extend the range of available source or destination addresses
- Port numbers are a UDP/TCP thing; there are no port numbers in IP packets without layering another protocol.
- IPsec is IP, not UDP/TCP, therefore doesn’t have port numbers available to do this mapping.
- NAT-Traversal IPsec embeds IPsec into UDP Port 4500 packets (necessary for roadwarrior VPNs, coming soon, as you’re sure to be NAT’d by your hotel/mobile operator and probably home). Details in **RFC3947**: be brave!
- Had to be bolted on to protocol later as IPsec was originally for IPv6, remember?

ESP in Action

```
[igb@offsite6 ~]$ ntpq -p -6 pi-one.batten.eu.org
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
xGPS_NMEA(0)     .GPS.            3 l    -   16  377    0.000    0.000   0.004
oPPS(0)          .PPS.            0 l    -    8  377    0.000    0.000   0.004
*ntp0.linx.net   .PPS.            1 u   11   64  377   13.795    0.497   3.041
+ntp1.linx.net   .PPS.            1 u    -   64  377   13.784    0.575   2.221
+ntp2.linx.net   .GPS.            1 u    -   64  377   15.477    0.981   4.928
[igb@offsite6 ~]$
```

Traffic Flows

```
14:05:52.880509 IP6 offsite6.batten.eu.org > pi-one-out.home.batten.eu.org: ESP(spi=0x054e459a,seq=0x31), length 68
14:05:52.881668 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x59), length 84
14:05:52.925064 IP6 offsite6.batten.eu.org > pi-one-out.home.batten.eu.org: ESP(spi=0x054e459a,seq=0x32), length 68
14:05:52.926505 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x5a), length 532
14:05:52.927140 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x5b), length 116
14:05:52.957383 IP6 offsite6.batten.eu.org > pi-one-out.home.batten.eu.org: ESP(spi=0x054e459a,seq=0x33), length 68
14:05:52.958816 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x5c), length 532
14:05:52.959440 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x5d), length 116
14:05:52.990874 IP6 offsite6.batten.eu.org > pi-one-out.home.batten.eu.org: ESP(spi=0x054e459a,seq=0x34), length 68
14:05:52.992432 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org: ESP(spi=0x0a632b19,seq=0x5e), length 532
```

ESP Setup

```
spdadd -6 ::/0[53] ::/0[any] udp -P out none;
spdadd -6 ::/0[53] ::/0[any] tcp -P out none;
spdadd -6 ::/0[any] ::/0[53] udp -P in none;
spdadd -6 ::/0[any] ::/0[53] tcp -P in none;

# or ssh
spdadd -6 ::/0[22] ::/0[any] tcp -P out none;
spdadd -6 ::/0[any] ::/0[22] tcp -P in none;
```

```
spdadd -6 offsite6.batten.eu.org pi-one-in.home.batten.eu.org any -P out IPsec esp/transport//require;
spdadd -6 pi-one-in.home.batten.eu.org offsite6.batten.eu.org any -P in IPsec esp/transport//require;
spdadd -6 offsite6.batten.eu.org pi-one-out.home.batten.eu.org any -P out IPsec esp/transport//require;
spdadd -6 pi-one-out.home.batten.eu.org offsite6.batten.eu.org any -P in IPsec esp/transport//require;
```

Keys are configured (how this is done is later)

```
2001:8b0:129f:a90e:3141:5926:5359:1 2a00:7b80:3019:12::579c:4928
   esp mode=transport spi=174271257(0x0a632b19) reqid=0(0x00000000)
   E: aes-cbc  acbfc369 623a8cd4 50f5eeda 42c8be7e 3e3b40fc 6ccc5ece
   A: hmac-sha256  07e62293 80e975f8 d52539ac b096fbe1 e5bac932 59989e78 d6f4bdcc ad04d3af
   seq=0x00000000 replay=4 flags=0x00000000 state=mature
   created: Mar 10 14:03:17 2015 current: Mar 10 14:07:23 2015
   diff: 246(s)  hard: 86400(s)  soft: 69120(s)
   last: Mar 10 14:03:28 2015 hard: 0(s)  soft: 0(s)
   current: 26144(bytes) hard: 0(bytes)soft: 0(bytes)
   allocated: 99 hard: 0 soft: 0
   sadb_seq=1 pid=15547 refcnt=0
2a00:7b80:3019:12::579c:4928 2001:8b0:129f:a90e:3141:5926:5359:1
   esp mode=transport spi=89015706(0x054e459a) reqid=0(0x00000000)
   E: aes-cbc  8c041c0f cdae4c14 bb46f4f8 a3700ebf 13e4a50f 6df95ab1
   A: hmac-sha256  93f8e3ee 72228356 acf21816 f4e248f4 b6fbf69d e6edf786 90db00ef c7cd54ff
   seq=0x00000000 replay=4 flags=0x00000000 state=mature
   created: Mar 10 14:03:17 2015 current: Mar 10 14:07:23 2015
   diff: 246(s)  hard: 86400(s)  soft: 69120(s)
   last: Mar 10 14:03:28 2015 hard: 0(s)  soft: 0(s)
   current: 1080(bytes)  hard: 0(bytes)soft: 0(bytes)
   allocated: 54 hard: 0 soft: 0
   sadb_seq=2 pid=15547 refcnt=0
```

Simple Connection (no IPsec)

```
mini-server:~ igb$ telnet pi-one 1234
Trying 2001:8b0:129f:a90e:3141:5926:5359:1...
Connected to pi-one.batten.eu.org.
Escape character is '^]'.
pi-one.home.batten.eu.org, DNS, DHCP, NTP (GPS), heyu, odds and ends.

*** This is a private machine.  If you are not personally authorised by
*** igb@batten.eu.org then your access is illegal.

Connection closed by foreign host.
mini-server:~ igb$
```

Plaintext Traffic

```
14:27:32.822472 IP pi-one.batten.eu.org.search-agent > mini-server.home.batten.eu.org.60350: Flags [P.],
seq 1:196, ack 1, win 453, options [nop,nop,TS val 14735272 ecr 344971307], length 195
   0x0000:  0026 bb60 07ce b827 ebe1 9651 0800 4500  .&.`...'...Q..E.
   0x0010:  00f7 bcc4 4000 4006 b4a0 51bb 96d3 0a5c  ....@.@...Q....\
   0x0020:  d5b1 04d2 ebbe 42d6 6b62 3361 f57f 8018  ......B.kb3a....
   0x0030:  01c5 f432 0000 0101 080a 00e0 d7a8 148f  ...2............
   0x0040:  d82b 7069 2d6f 6e65 2e68 6f6d 652e 6261  .+pi-one.home.ba
   0x0050:  7474 656e 2e65 752e 6f72 672c 2044 4e53  tten.eu.org,.DNS
   0x0060:  2c20 4448 4350 2c20 4e54 5020 2847 5053  ,.DHCP,.NTP.(GPS
   0x0070:  292c 2068 6579 752c 206f 6464 7320 616e  ),.heyu,.odds.an
   0x0080:  6420 656e 6473 2e0a 0a2a 2a2a 2054 6869  d.ends...***.Thi
   0x0090:  7320 6973 2061 2070 7269 7661 7465 206d  s.is.a.private.m
   0x00a0:  6163 6869 6e65 2e20 2049 6620 796f 7520  achine...If.you.
   0x00b0:  6172 6520 6e6f 7420 7065 7273 6f6e 616c  are.not.personal
   0x00c0:  6c79 2061 7574 686f 7269 7365 6420 6279  ly.authorised.by
   0x00d0:  0a2a 2a2a 2069 6762 4062 6174 7465 6e2e  .***.igb@batten.
   0x00e0:  6575 2e6f 7267 2074 6865 6e20 796f 7572  eu.org.then.your
   0x00f0:  2061 6363 6573 7320 6973 2069 6c6c 6567  .access.is.illeg
   0x0100:  616c 2e0a 0a                             al...
```

Via IPsec ESP

```
[igb@offsite6 ~]$ telnet 2001:8b0:129f:a90e:3141:5926:5359:1  1234
Trying 2001:8b0:129f:a90e:3141:5926:5359:1...
Connected to 2001:8b0:129f:a90e:3141:5926:5359:1.
Escape character is '^]'.
pi-one.home.batten.eu.org, DNS, DHCP, NTP (GPS), heyu, odds and ends.

*** This is a private machine.  If you are not personally authorised by
*** igb@batten.eu.org then your access is illegal.

Connection closed by foreign host.
[igb@offsite6 ~]$
```

ESP in Action

```
14:41:13.207506 IP6 pi-one-out.home.batten.eu.org > offsite6.batten.eu.org:
ESP(spi=0x0a632b19,seq=0x9b), length 276
   0x0000:  6000 0000 0114 3240 2001 08b0 129f a90e  `.....2@........
   0x0010:  3141 5926 5359 0001 2a00 7b80 3019 0012  1AY&SY..*.{.0...
   0x0020:  0000 0000 579c 4928 0a63 2b19 0000 009b  ....W.I(.c+.....
   0x0030:  b035 8270 e510 ffed fcdc a419 589b 287c  .5.p........X.(|
   0x0040:  3444 0fd7 9a8a bb6c 7051 a47b 5253 ebbd  4D.....lpQ.{RS..
   0x0050:  7cad d27a 440f 189a afcb f700 8f05 f677  |..zD..........w
   0x0060:  9ab1 da2b d833 f1e0 aed5 bf2f 65e6 d605  ...+.3...../e...
   0x0070:  1f1b 8598 2702 9d07 246e 57f8 c751 3ba3  ....'...$nW..Q;.
   0x0080:  48f3 ba8d 34d8 1ebd 29c7 2fa5 78ef d754  H...4...)./.x..T
   0x0090:  73fb 0413 e029 9c9f 9ca8 2138 557b 4735  s....)....!8U{G5
   0x00a0:  da39 9db2 7a8d 0e0e f4aa f024 6977 b1b8  .9..z......$iw..
   0x00b0:  8fb7 dfcc 840d c740 988c fa0f d30a 1b08  .......@........
   0x00c0:  f505 c00c 8041 52a2 2727 9916 2955 02d5  .....AR.''..)U..
   0x00d0:  4ef2 95dd 118e 2804 9482 9f0a 1b57 269e  N.....(......W&.
   0x00e0:  2277 eb0d 370c 2035 d78d 6071 8aae b11e  "w..7..5..`q....
   0x00f0:  ea57 8af8 b53a 3cae 752a 91f9 1e1d 7d8d  .W...:<.u*....}.
   0x0100:  00b6 a304 c851 568b f452 e7e5 bc0b efd9  .....QV..R......
   0x0110:  708d 02d8 5607 bca7 3c7c c417 3333 5944  p...V...<|..33YD
   0x0120:  d7cc e4d0 34ef 2232 5af6 4ffe 401b 77df  ....4."2Z.O.@.w.
   0x0130:  02f9 6e10 0aae e5f8 37cb 4518            ..n.....7.E.
```