

Cisco 892 router performance test

Juri Jestin
16.09.2014

About test

Test was carried out for clarify the performance characteristics of the router with different configurations during transmission of packets with different length. Some background information is present in document also for better understanding.

During the test were used:

- Cisco 892, IOS Software Universal/K9, Version 15.1(2)T3, license feature advipservices , VPN module onboard.
- JDSU SmartClass Ethernet Tester.
- Passive TAP for a traffic capturing.
- Software Wireshark for a packet inspection.

Measured L1 Rate [Mbps] results in tables are rounded. However, it is sufficient for the review. A more accurate value you can get out from Packets Measured Rate.

For example: for 64Byte packets L1 Rate is 7.6Mbps and packets rate is 9343 packets per seconds. In this case 7.6Mbps is rounded. A more accurate value is $9343 \text{ [packets/s]} \times 102 \text{ [bytes on wire]} \times 8 \text{ [bits in byte]} = 7.623888 \text{ Mbps}$.

Please note, Measured L1 Rate results for IPSec tests are end-to-end throughput. This rate does not contain the IPSec protocol overheads on the tunnel part.

For example, if IPSec ESP configuration is used DES encryption algorithm, then IPSec overhead is 56 bytes for original 64 byte packets. In case, if end-to-end throughput is 8490 packets/sec or 6,9Mbps, circuit part for IPSec tunnel must have throughput $8490 \text{ [packets/s]} \times 158 \text{ [bytes on wire]} \times 8 \text{ [bits in byte]} = 10,73\text{Mbps}$.

Background

Each layer have own unit of measure.

PDU – Protocol Data Unit

- Layer 1 Physical Layer - Bit
- Layer 2 Data Link Layer - Frame
- Layer 3 Network Layer - Packet

Ethernet frame Ethernet II / DIX

Figure 1 shows fields and lengths of Ethernet II/DIX frame.

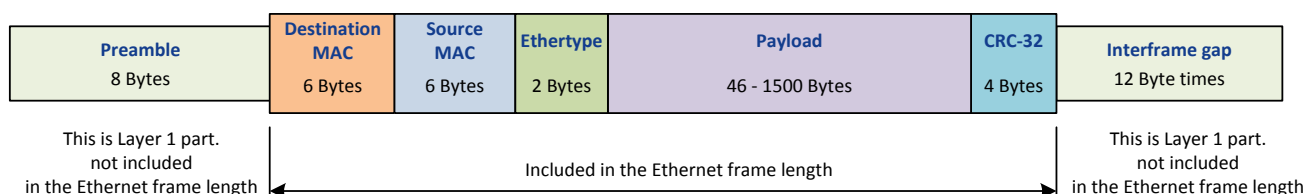


Figure 1. Untagged Ethernet frame

Maximum throughput

The interframe gap is inserted between frames during transmitting (Figure 2)

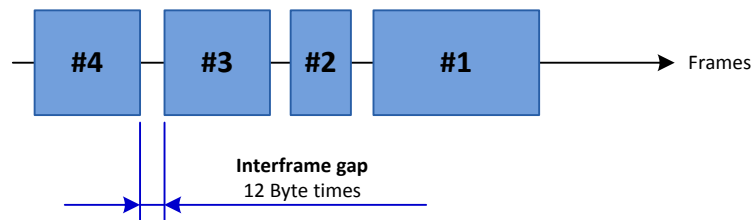


Figure 2 Ethernet interframe gap

Preamble 8 Bytes + Destination MAC 6 Bytes + Source MAC 6 Bytes + Ethertype 2Bytes + Payload 1500 bytes + FCS 4 Bytes + Interframe Gap "12 Bytes" = 1538 Bytes. In this way 1538 bytes are needed to transmit 1518 bytes untagged frame.

Layer 3 maximum throughput can't reach 100% wire speed and depends from packet length.

1538 Bytes are needed for transmitting 1500 bytes of L3 data -> $1500/1538 * 100\% = 97.53\%$ @ untagged frame.

84 Bytes are needed for transmitting 46 bytes of L3 dat -> $64/84 * 100\% = 76.19\%$ @ untagged frame.

Cisco Router order of operation

Cisco Router execution order of operations defines how the router processes traffic.

Packet is received on the port.

- If IPSec, then check input access list
- Decryption—for Cisco Encryption Technology (CET) or IPSec
- Check input access list
- Check input rate limits
- Input accounting
- NAT outside to inside (global to local translation)
- Policy routing
- Routing
- Redirect to Web cache
- NAT inside to outside (local to global translation)
- Crypto (check map and mark for encryption)
- Check output access list
- Inspect context-based access control (CBAC)
- TCP intercept
- Encryption

Packet is transmitted to the port.

Test #1

Cisco 892 switching performance in the case No service enabled or ACL.

The router configured with the IP address, no service enabled or ACL. Router works as gateway between two subnets.
Test type: Layer 3 RFC2544.

Router has different Ethernet PHY level ICs.

Gigabit Ethernet PHY IC Marvell 88E1118 port Gi0
Fast Ethernet PHY IC Broadcom BCM5241 port Fa8
Switch Marvell 88E6097 ports Fa0-Fa7

In this case, two variants were tested.

- 1st variant: Ports are used from one IC Fa8-Gi0 (Figure 1)
- 2nd variant: Ports are used from different ICs Fa0-Gi0 (Figure 2)

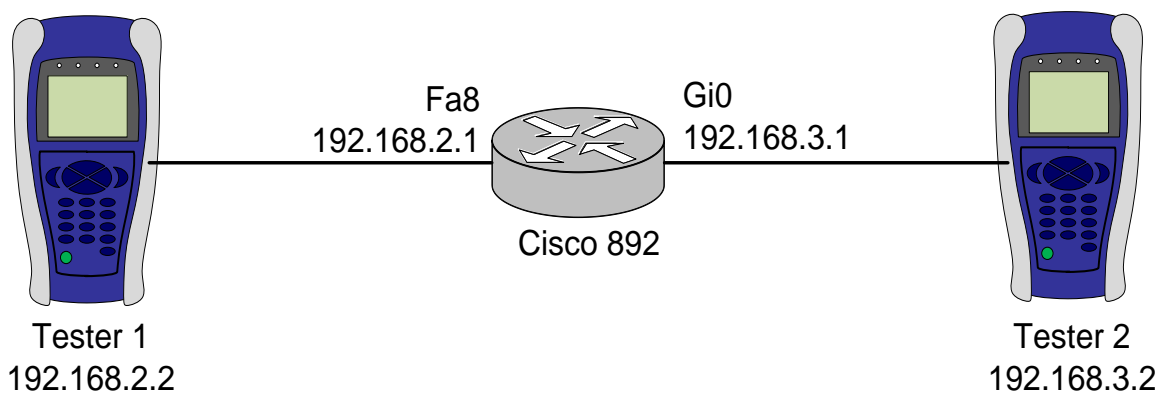


Figure 3 Schematic diagram of Test #1, 1st variant

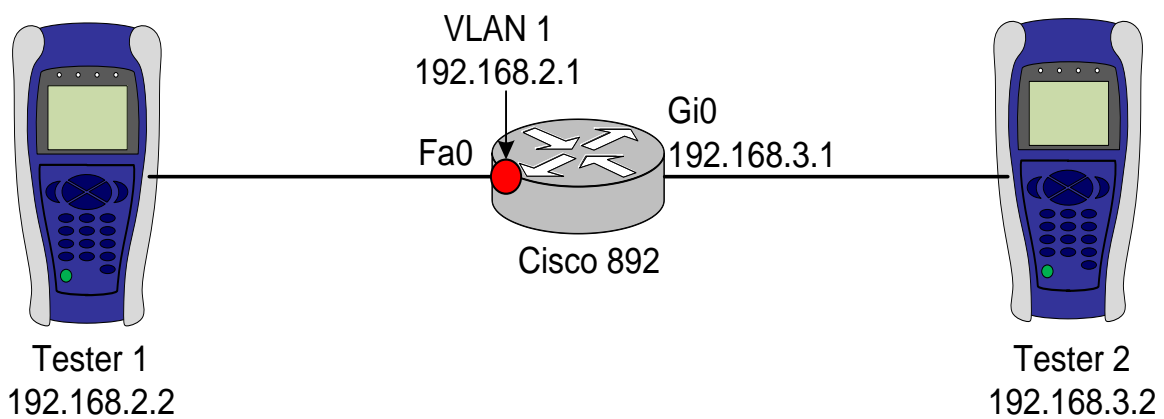
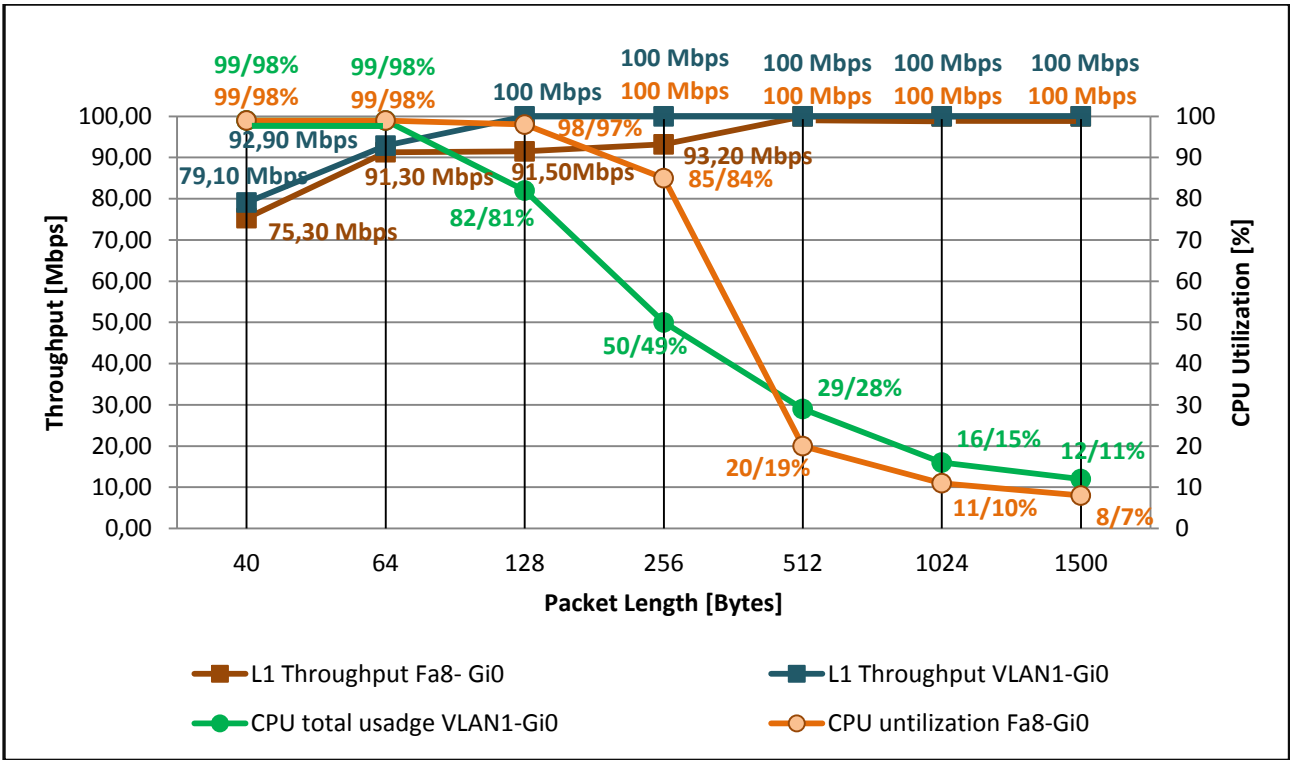


Figure 4 Schematic diagram of Test #1, 2nd variant



Graph 1. Throughput and CPU utilization. Test #1, No service enabled or ACL.

Test #2

Cisco 892 switching performance in the case of single ACL line.

The router configured with the IP address, single line ACL, no additional service enabled. Router works as gateway between two subnets. Test type: Layer 3 RFC2544.

Part of R1 router configuration is presented below:

For variant 1

```
R1#sh run | s Vlan1|access
interface Vlan1
 ip address 192.168.2.1 255.255.255.0
 ip access-group 100 in
access-list 100 permit ip host 192.168.2.2 host 192.168.3.2
```

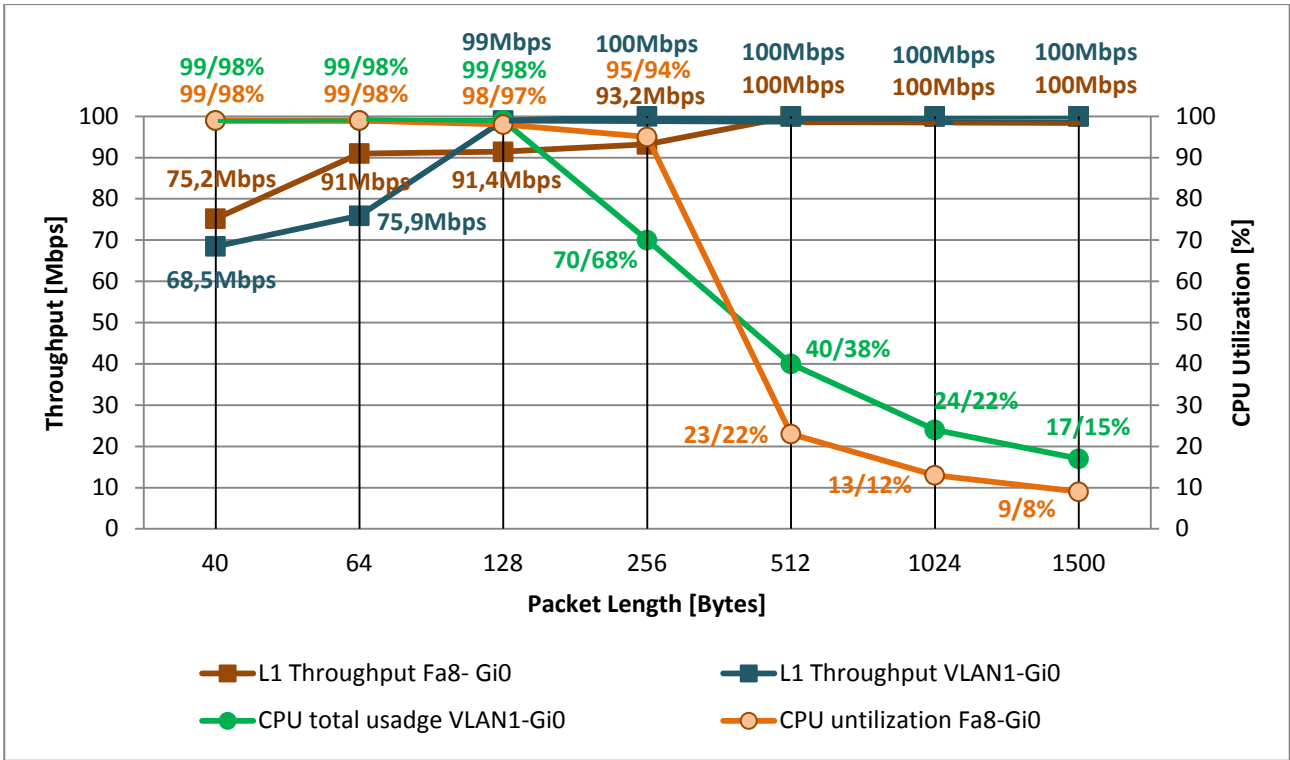
For variant 2

```
R1#sh run | s FastEthernet8|access
interface FastEthernet8
 ip address 192.168.2.1 255.255.255.0
 duplex full
 speed 100
 ip access-group 100 in
access-list 100 permit ip host 192.168.2.2 host 192.168.3.2
```

Test results are present in the Table 2 and on the Graph 2.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	75,20	99	98	111933	68,50	99	98	101930
64	91,00	99	98	111551	75,90	99	98	93053
128	91,40	98	97	68844	99,00	99	98	75224
256	93,20	95	94	39632	100,00	70	68	42514
512	100,00	23	22	22725	100,00	40	38	22725
1024	100,00	13	12	11769	100,00	24	22	11769
1500	100,00	9	8	8127	100,00	17	15	8127

Table 2. Throughput and CPU utilization. Test #2 with a single ACL line



Graph 2. Throughput and CPU utilization. Test #2 with a single ACL line.

Test #3

Cisco 892 switching performance in the case of two ACL lines.

New ACL rules are added into the configuration. The router configured with the IP address, with ACL lines, no additional service enabled. Router works as gateway between two subnets.

Test type: Layer 3 RFC2544.

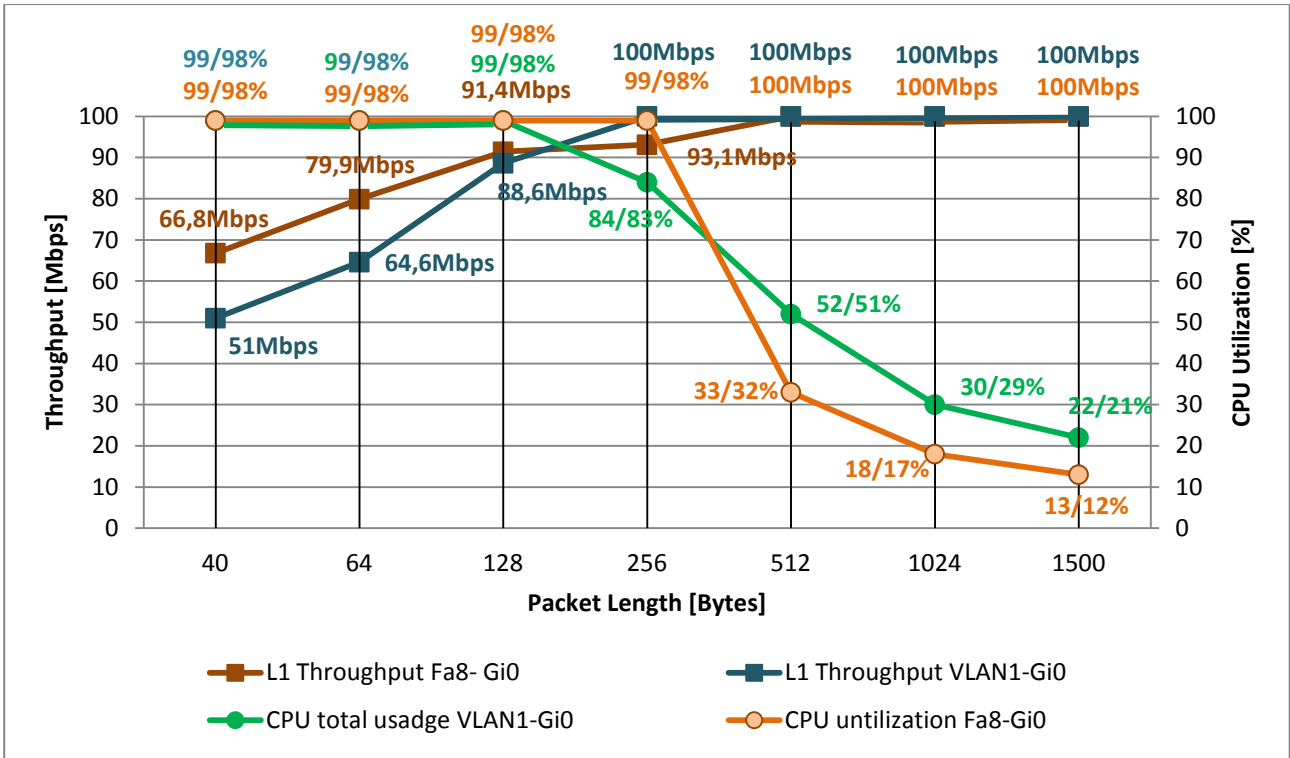
Part of R1 router configuration is presented below:

```
R1(config)#do sh run | s Vlan1|GigabitEthernet0|access
interface GigabitEthernet0
 ip address 192.168.3.1 255.255.255.0
 ip access-group 100 in
 ip access-group 100 out
 duplex full
 speed 100
interface Vlan1
 ip address 192.168.2.1 255.255.255.0
 ip access-group 100 in
 ip access-group 100 out
access-list 100 permit ip host 192.168.2.2 host 192.168.3.2
access-list 100 permit ip host 192.168.3.2 host 192.168.2.2
```

CPU checks ACL lines 6 times for each packet, 2 times for a forward path and 4 times for a return path. Test results are present in the Table 3 and on the Graph 3.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	66,80	99	98	99403	51,00	99	98	75894
64	79,90	99	98	97863	64,60	99	98	79166
128	91,4	99	98	68802	88,60	99	98	66719
256	93,1	99	98	39597	100,00	84	83	42515
512	100,00	33	32	22725	100,00	52	51	22725
1024	100,00	18	17	11769	100,00	30	29	11769
1500	100,00	13	12	8127	100,00	22	21	8127

Table 3. Throughput and CPU utilization. Test #3 with 2 ACL lines.



Graph 3. Throughput and CPU utilization. Test #3 with 2 ACL lines.

Test #4

Cisco 892 switching performance in the case of 12 ACL lines.

New ACL rules are added into the configuration. Additional 10 lines with non-existent IP address do additional checking for each packet. The CPU checks ACL lines 46 times for each packet, 22 times for a forward path and 24 times for a return path.

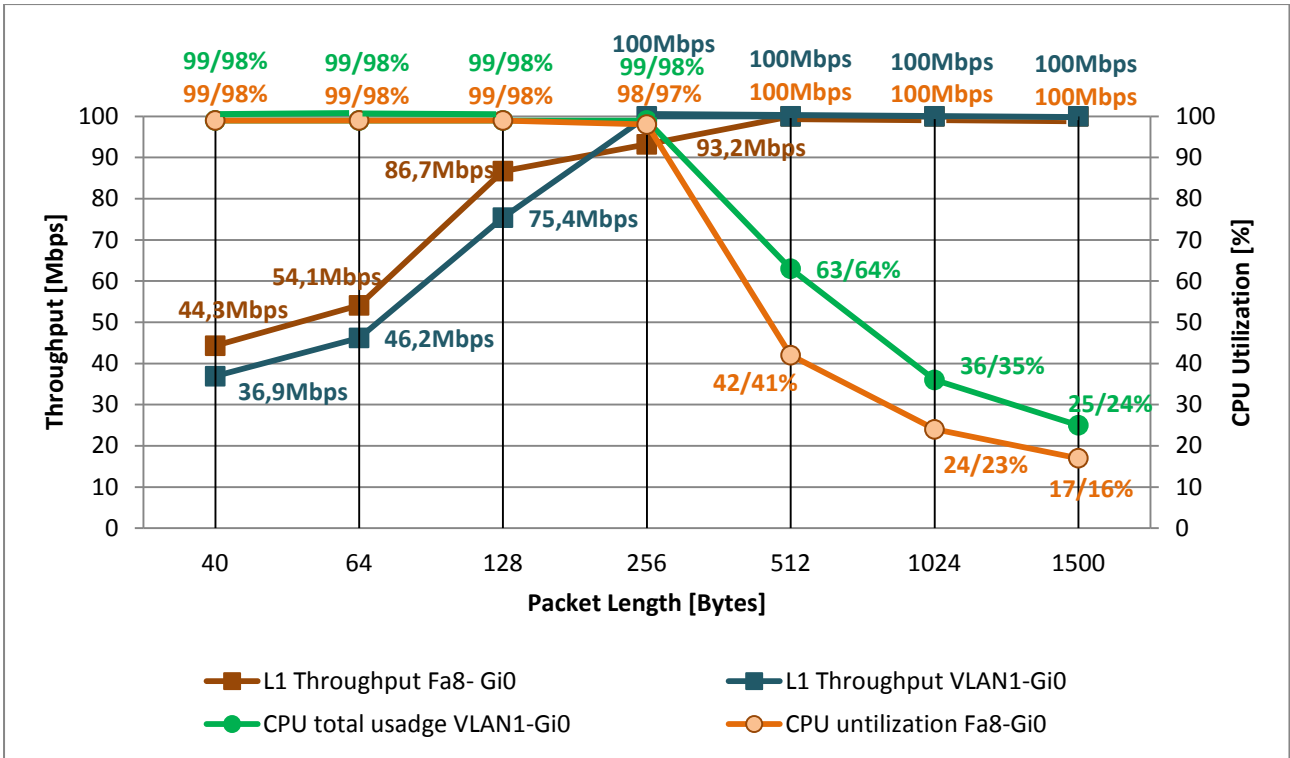
```
R1(config)#access-list 100 permit ip host 192.150.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.151.3.2 host 192.168.2.2
R1(config)#access-list 100 permit ip host 192.152.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.153.3.2 host 192.168.2.2
R1(config)#access-list 100 permit ip host 192.154.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.155.3.2 host 192.168.2.2
R1(config)#access-list 100 permit ip host 192.156.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.157.3.2 host 192.168.2.2
R1(config)#access-list 100 permit ip host 192.158.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.159.3.2 host 192.168.2.2
R1(config)#access-list 100 permit ip host 192.168.2.2 host 192.168.3.2
R1(config)#access-list 100 permit ip host 192.168.3.2 host 192.168.2.2
```

<----- real IP addresses
<----- real IP addresses

Test results are present in the Table 4 and on the Graph 4.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	44,30	99	98	65856	36,90	99	98	54868
64	54,10	99	98	66304	46,20	99	98	56655
128	86,70	99	98	65273	75,40	99	98	56813
256	93,20	98	97	39621	100,00	99	98	42514
512	100,00	42	41	22725	100,00	63	62	22725
1024	100,00	24	23	11769	100,00	36	35	11769
1500	100,00	17	16	8127	100,00	25	24	8127

Table 4. Throughput and CPU utilization. Test #4 with 12 ACL lines.



Graph 4. Throughput and CPU utilization. Test #4 with 12 ACL lines.

IPsec VPN Background

IPsec functionality provides data encryption at the IP packet level. Basic IPsec provides secure point-to-point tunnels between two peers, such as two routers. During tunnel establishment, the two peers negotiate security associations (SA) that govern authentication, encryption, encapsulation, and key management.

These negotiations involve two phases:

Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages.

Phase 2 creates the tunnel that protects data.

SAs are unidirectional and are established per security protocol, either Authentication Header (AH) or Encapsulating Security Payload (ESP).

Combination of IPsec protection can be used: authentication only or both authentication and encryption. These methods have different protocol overheads. It influences differently to router performance and to total circuit throughput.

IPsec **tunnel mode** is the default mode. With tunnel mode, the entire original IP packet is protected by IPsec. This means IPsec wraps the original packet, protect it, adds a new IP header and sends it to the other side of the VPN tunnel (IPsec peer).

In the test case, new IP header has IPsec peer's IP addresses 192.168.1.1 and 192.168.1.2 instead tester's IP addresses 192.168.2.2 and 192.168.3.2.

IPsec **transport mode** is used for end-to-end communications, for example, if traffic and tunnel originated in same equipment. The original IP headers remain intact, except that the IP protocol field is changed to ESP (50) or AH (51), and the original protocol value is saved in the IPsec trailer to be restored when the packet is decrypted. IPsec transport mode is usually used when another tunneling protocol (like GRE) is used to first encapsulate the IP data packet, then IPsec is used to protect the GRE tunnel packets. IPsec protects the GRE tunnel traffic in transport mode.

Transport mode was configured also during test. However, traffic between testers is transit for routers and they ignored the transport mode in configuration and automatically used tunnel mode. For example:

```
crypto ipsec transform-set AH_TEST ah-md5-hmac
mode transport
```

But you can check `ipsec sa` and see that `tunnel mode` is used.

```
R1#sh crypto ipsec sa
***
inbound ah sas:
spi: 0x7D1(2001)
transform: ah-md5-hmac ,
in use settings ={Tunnel, }
```

General schemes for testing IPsec performance are shown on Figures 5 and 6. Passive taps are used for capturing and inspecting packets.

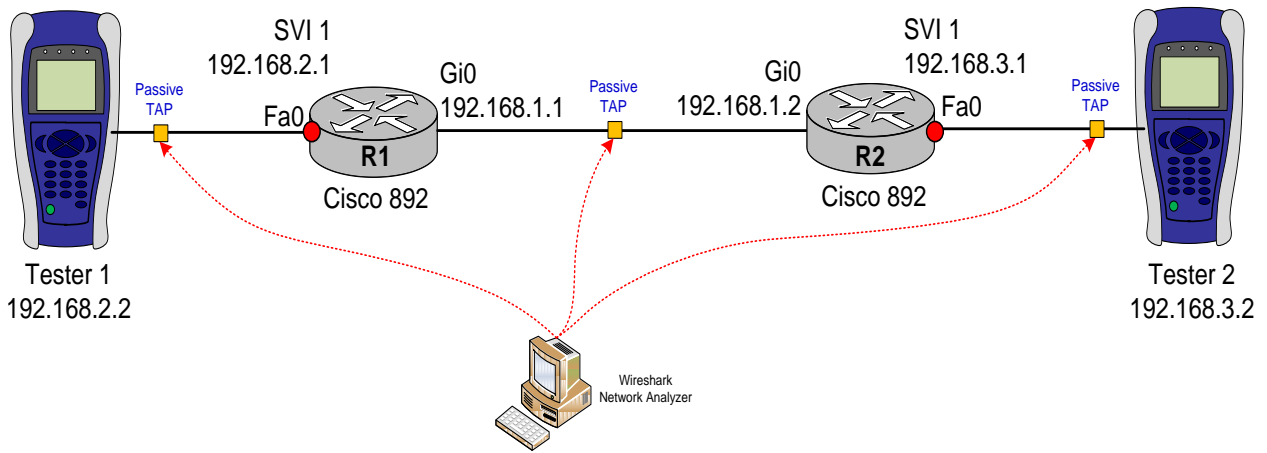


Figure 5. General scheme for testing IPsec in the case VLAN1 – Gi0.

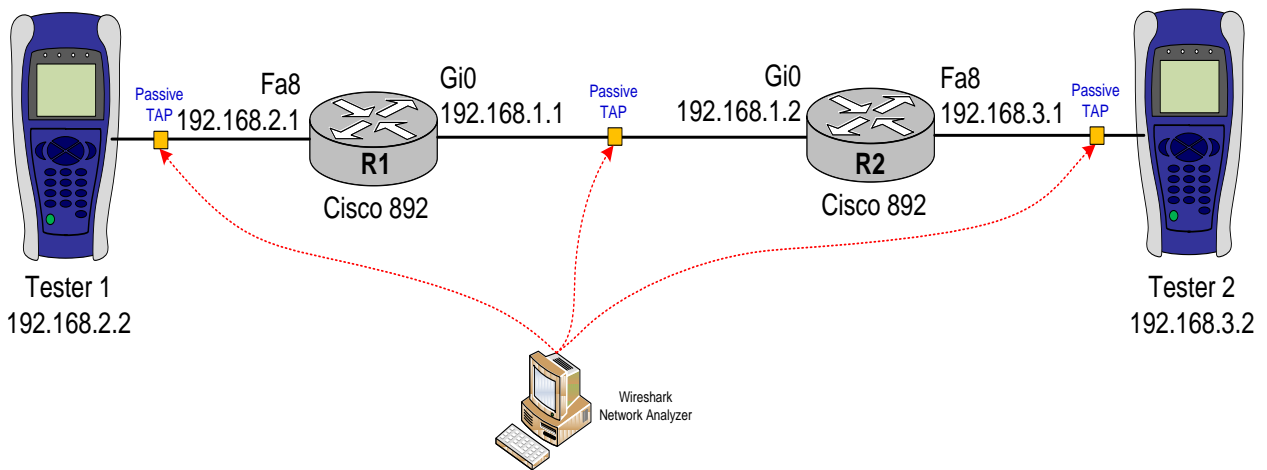


Figure 6. General scheme for testing IPsec in the case Fa8 – Gi0.

Test #5

Cisco 892 switching performance in the case of IPSec with AH MD5 Authentication and without ESP encryption.

AH Authentication's job is to protect the entire packet. The AH does not protect all of the fields in the New IP Header because some change in transit, and the sender cannot predict how they might change. The AH protects everything that does not change in transit. AH is identified in the new IP header with an IP protocol ID of 51.

In this test the manual keying is used.

Manual keying is usually only necessary when a Cisco device is configured to encrypt traffic to another vendor's device which does not support Internet Key Exchange (IKE). If IKE is configurable on both devices, it is preferable to use automatic keying

Part of router configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set AH_TEST ah-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.2
  set session-key inbound ah 2001 00
  set session-key outbound ah 1002 10
set transform-set AH_TEST
match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set AH_TEST ah-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.1
  set session-key inbound ah 1002 10
  set session-key outbound ah 2001 00
set transform-set AH_TEST
match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

Verification:

Router has no active ISAKMP SA, because we use manual keying. You can check it with the following command:

```
R1#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state          conn-id status
```

IPSec security association has only AH authentication.

```
R1#sh crypto ipsec sa
interface: GigabitEthernet0
  Crypto map tag: TEST, local addr 192.168.1.1

protected vrf: (none)
local ident (addr/mask/prot/port): (192.168.2.2/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (192.168.3.2/255.255.255.255/0/0)
current_peer 192.168.1.2 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 13052329, #pkts encrypt: 13052329, #pkts digest: 13052329
#pkts decaps: 4175662, #pkts decrypt: 4175662, #pkts verify: 4175662
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 73523, #recv errors 0

local crypto endpt.: 192.168.1.1, remote crypto endpt.: 192.168.1.2
path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet0
current outbound spi: 0x3EA(1002)
PFS (Y/N): N, DH group: none

inbound esp sas: ← no ESP

inbound ah sas: ← only AH
spi: 0x7D1(2001)
  transform: ah-md5-hmac ,
  in use settings ={Tunnel, }
  conn id: 33, flow_id: Onboard VPN:33, sibling_flags 80000056, crypto map: TEST
  no sa timing
  replay detection support: N
  Status: ACTIVE

inbound pcp sas:

outbound esp sas: ← no ESP

outbound ah sas: ← only AH
spi: 0x3EA(1002)
  transform: ah-md5-hmac ,
  in use settings ={Tunnel, }
  conn id: 34, flow_id: Onboard VPN:34, sibling_flags 80000056, crypto map: TEST
  no sa timing
  replay detection support: N
  Status: ACTIVE

outbound pcp sas:
```


The command `#sh crypto engine connections active` shows active connection and used algorithm.

```
R1#sh crypto engine connections active
Crypto Engine Connections

ID  Type   Algorithm      Encrypt  Decrypt  LastSeqN  IP-Address
25  IPsec  MD5            0        911694   0          192.168.1.1
26  IPsec  MD5            1010998  0        0          192.168.1.1
```

Test results are present in the Table 5. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	5,4	99	98	8070	5,3	99	98	7893
64	7,1	99	98	8704	6,5	99	98	7998
128	11,5	99	98	8640	10,5	99	98	7918
256	20,2	99	98	8584	18,7	99	98	7941
512	37,5	99	98	8512	34,4	99	98	7829
1024	70,9	99	98	8343	65,7	99	98	7736
Random	22,0	79	76		20,0	78	75	

Table 5. Throughput and CPU utilization. Test #5. IPsec with AH only.

Capture and analysis of packets has been made to learn the protocol overhead with different settings IPSEC.

Original captured frame is shown in Wireshark screenshot (Figure 7). There is written 78 bytes on wire. Wireshark shows frame length without 4 byte FCS checksum. In this example 6 bytes Source MAC + 6 bytes Destination MAC + 2bytes Ethertype field + 64 bytes L3 packet = 78 bytes. L3 Packet contains 20 bytes IP header + 44 bytes pattern 0xFF.

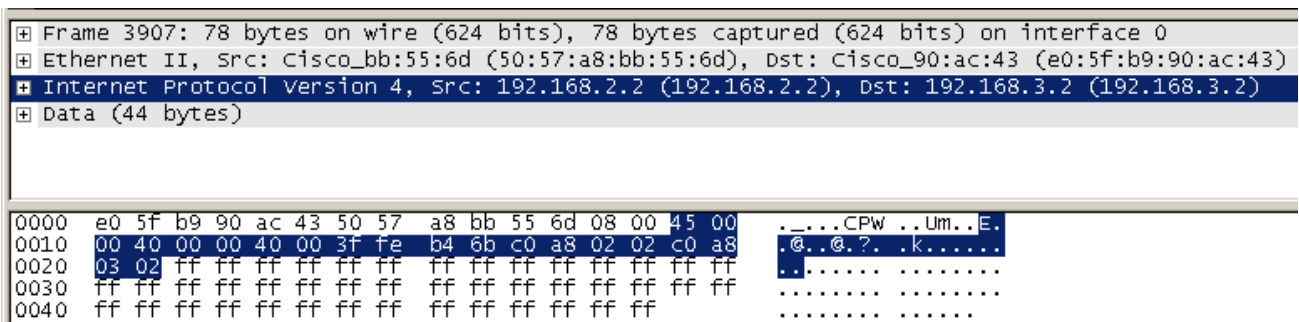


Figure 7. Original IP packet 64 byte length.

Structure of IPsec packet with AH authentication is shown in the Figure 8

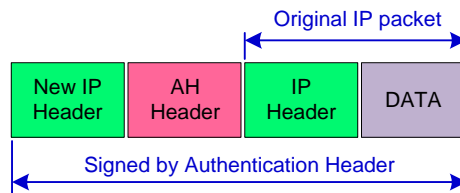


Figure 8. Format of packet with AH authentication

Captured IPsec packet is shown in the Figure 9. The AH header is 24bytes, plus new IP header 20 bytes. Total overhead for AH authentication is 44 bytes. Also you can see that now frame on wire is 122 bytes length. Original frame is 78 bytes length on wire. If we subtract L2 frame lengths, we also have $122 - 78 = 44$ bytes of overhead.

```

⊞ Frame 54509: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
⊞ Ethernet II, Src: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d), Dst: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43)
⊞ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
⊞ Authentication Header
  Next Header: IPIP (0x04)
  Length: 24
  AH SPI: 0x000003ea
  AH Sequence: 228246788
  AH ICV: d4ee01cf63f6df505235cfb9
⊞ Internet Protocol Version 4, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.3.2 (192.168.3.2)
⊞ Data (44 bytes)
  
```

0000	e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00 CPW . . Um . . E .
0010	00 6c e5 29 40 00 ff 33 12 e1 c0 a8 01 01 c0 a8	. 1 .) @ . 3
0020	01 02 04 04 00 00 00 00 03 ea 0d 9a c5 04 d4 ee
0030	01 cf 63 f6 df 50 52 35 cf b9 45 00 00 40 00 00	. . c . . PR5 . . E . . @ . .
0040	40 00 31 ff c2 6a c0 a8 02 02 c0 a8 03 02 ff ff	@ . 1 . . j
0050	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
0060	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
0070	ff ff ff ff ff ff ff ff

- Authentication header
- AH SPI 0x3EA = 1002 (it was manually configured)
- Not encrypted original IP header
- Not encrypted DATA

Figure 9. IPsec packet with AH authentication

IPsec tunnel works in tunnel mode. In this case you can see a new IP header with new IP addresses. They are IPsec peer's addresses: Source IP 192.168.1.1 Destination IP 192.168.1.2. Original IP packet is not encrypted (in this test) and follows after AH header. You can see the original IP header with the original IP addresses: Source IP 192.168.2.2 Destination IP 192.168.3.2.

Test #6

Cisco 892 switching performance in the case of IPSec with ESP-NULL encryption.

The NULL encryption algorithm with the IPSec Encapsulating Security Payload (ESP) does nothing to alter plaintext data. In fact, NULL, by itself, does nothing. NULL provides the means for ESP to provide authentication and integrity without confidentiality.

Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL
crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.2
  set session-key inbound esp 2001 authenticator 00
  set session-key outbound esp 1002 authenticator 10
  set transform-set ESP-NULL
match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL
crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.1
  set session-key inbound esp 1002 authenticator 10
  set session-key outbound esp 2001 authenticator 00
  set transform-set ESP-NULL
match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

Test results are present in the Table 6. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	6,3	97	93	9302	6,0	99	98	8896
64	7,6	97	93	9343	7,3	99	98	9003
128	13,2	99	98	9918	11,8	99	98	8900
256	23,1	99	98	9831	21,0	99	98	8944
512	43,0	99	98	9782	38,9	99	98	8845
1024	80,2	99	98	9436	73,7	99	98	8673
Random	24,0	80	75		22,0	79	76	

Table 6. Throughput and CPU utilization. Test #6. IPSec with ESP-NULL.

Structure of IPsec packet with ESP payload is shown in the Figure 10.

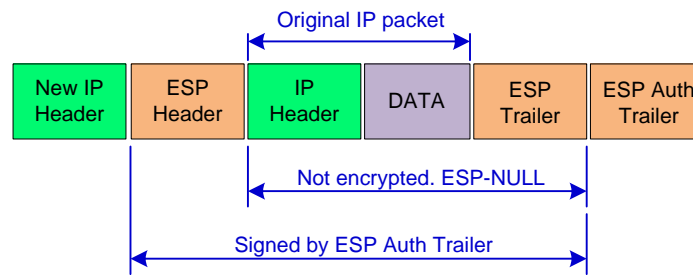


Figure 10. Format of packet with ESP.

Captured IPsec packet is shown in the Figure 11. Total Overhead is 44 bytes – new IP header 20 bytes + ESP header/trailer 24 bytes.

```

⊕ Frame 4305: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
⊕ Ethernet II, Src: Cisco_bb:55:6d (50:57:a8:bb:55:6d), Dst: Cisco_90:ac:43 (e0:5f:b9:90:ac:43)
⊕ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
⊖ Encapsulating Security Payload
    ESP SPI: 0x521d2524 (1377641764)
    ESP Sequence: 782041
0000  e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  . . . . CPw ..Um..E.
0010  00 6c 58 28 40 00 ff 32 9f e3 c0 a8 01 01 c0 a8  . 7X(@.2 .....
0020  01 02 52 1d 25 24 00 0b ee d9 45 00 00 40 00 00  . . R.%$. . .E..@.
0030  40 00 3f fe b4 6b c0 a8 02 02 c0 a8 03 02 ff ff  @.?..k..
0040  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0050  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0060  ff ff ff ff ff ff ff ff ff ff 01 02 02 04 3f ee  .....?
0070  29 cb 89 6f 25 5c b5 a4 60 4d  )..o%\.. `M
  
```

- ESP SPI identifier
- ESP Sequence number
- Not encrypted original IP header
- Not encrypted DATA
- ESP trailer
- ESP Authentication

Figure 11. IPsec packet with ESP payload.

Please note

There is a point for potential errors when using the manual keying. Please see the configuration below. Here, the IPsec transform is configured with ESP authentication `crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac`, but crypto map uses manual keying with AH authentication `set session-key inbound ah 1001 10`.

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.2
  set session-key inbound ah 2001 00
  set session-key outbound ah 1002 10
  set transform-set ESP-NULL
match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.1
  set session-key inbound ah 1002 10
  set session-key outbound ah 2001 00
  set transform-set ESP-NULL
match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

This configuration does not work. Manual AH session key kills ESP and sets ESP SPI to zero. Figure 12 shows a captured packet.

```

⊞ Frame 85447: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
⊞ Ethernet II, Src: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d), Dst: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43)
⊞ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
⊞ Encapsulating Security Payload
    ESP SPI: 0x00000000 (0)
    ESP Sequence: 363977

```

```

0000 e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  ._.CPW ..Um..E.
0010 00 6c b6 24 40 00 ff 32 41 e7 c0 a8 01 01 c0 a8  .l.$@..2 A.....
0020 01 02 00 00 00 00 00 05 8d c9 45 00 00 40 00 00  .....E..@..
0030 40 00 3f fe b4 6b c0 a8 02 02 c0 a8 03 02 ff ff  @.?..k..
0040 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0050 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0060 ff ff ff ff ff ff ff ff ff ff 01 02 02 04 f1 91  .....
0070 cc d8 19 3b 68 96 8a 09 7f d6  ....h...

```

- ESP Payload
 - ESP SPI identifier
 - ESP Sequence number
 - Not encrypted original IP header
 - Not encrypted DATA
 - ESP trailer
 - ESP Authentication

Figure 12. ESP payload with zero SPI

You can see that ESP SPI is 0. Warning message is displayed on the console every 60 seconds.

```

*Sep 12 11:29:31.310: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
destaddr=192.168.1.2, prot=50, spi=0x0(0), srcaddr=192.168.1.1, input interface=GigabitEthernet0

```

```

*Sep 12 11:30:31.310: %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
destaddr=192.168.1.2, prot=50, spi=0x0(0), srcaddr=192.168.1.1, input interface=GigabitEthernet0

```

You can use some the **show** commands for checking this case:

```

R1#sh crypto session
Crypto session current status

Interface: GigabitEthernet0
Session status: UP-NO-IKE
Peer: 192.168.1.2 port 500
IPSEC FLOW: permit ip host 192.168.2.2 host 192.168.3.2
Active SAs: 2, origin: manual-keyed crypto map

```

You can see that SAs are active, but really traffic is not transmitted. You can check it with next show command.

```

R1#sh crypto engine connections active
Crypto Engine Connections

ID  Type   Algorithm      Encrypt  Decrypt  LastSeqN  IP-Address
25  IPsec  MD5            0        0        0  192.168.1.1
26  IPsec  MD5           1303903  0        0  192.168.1.1

```

```

R1#sh crypto ipsec sa

interface: GigabitEthernet0
Crypto map tag: TEST, local addr 192.168.1.1

***
#pkts encaps: 2823845, #pkts encrypt: 2823845, #pkts digest: 2823845
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
***

```

The router encrypts packets only in egress direction, but 0 ingress packets are decrypted. This occurs because router waits another ESP SPI identifier.

Change the configuration:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.2
  set session-key inbound esp 2001 authenticator 00
  set session-key outbound esp 1002 authenticator 10
  set transform-set ESP-NULL
  match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-manual
  set peer 192.168.1.1
  set session-key inbound esp 1002 authenticator 10
  set session-key outbound esp 2001 authenticator 00
  set transform-set ESP-NULL
  match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

After changing of configuration and resetting SA, you can see that traffic is transmitted via tunnel.

```
R1#sh crypto engine connections active
Crypto Engine Connections

  ID  Type    Algorithm      Encrypt  Decrypt  LastSeqN  IP-Address
  --  --      -
  29  IPsec   MD5             0        101489   0 192.168.1.1
  30  IPsec   MD5           361109   0         0 192.168.1.1
```

```
R1#sh crypto ipsec sa

interface: GigabitEthernet0
  Crypto map tag: TEST, local addr 192.168.1.1
***
  #pkts encaps: 5121844, #pkts encrypt: 5121844, #pkts digest: 5121844
  #pkts decaps: 1323489, #pkts decrypt: 1323489, #pkts verify: 1323489
***

  inbound esp sas:
    spi: 0x3EA(1002)  <- configured ID
***

  outbound esp sas:
    spi: 0x7D1(2001)  <- configured ID
```

Figures 13 and 14 show that ESP payload now has the right SPI number.

```

Frame 4571: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
Ethernet II, Src: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d), Dst: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43)
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
Encapsulating Security Payload
  ESP SPI: 0x000003ea (1002)
  ESP Sequence: 2539287
0000  e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  ...CPW ..Um..E.
0010  00 6c 86 1f 40 00 ff 32 71 ec c0 a8 01 01 c0 a8  .l..@..2 q.....
0020  01 02 00 00 03 ea 00 26 bf 17 45 00 00 40 00 00  .....& ..E..@..
0030  40 00 3f fe b4 6b c0 a8 02 02 c0 a8 03 02 ff ff  @.?.k.. ..
0040  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0050  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0060  ff ff ff ff ff ff ff ff ff ff 01 02 02 04 51 a6  .....
0070  ed 82 ee 54 6b 10 c0 ce 6b ba  ...Tk... k.
  
```

Figure 13. ESP packet with ESP SPI: 1002

```

Frame 3214: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
Ethernet II, Src: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43), Dst: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d)
Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)
Encapsulating Security Payload
  ESP SPI: 0x000007d1 (2001)
  ESP Sequence: 3000352
0000  50 57 a8 bb 55 6d e0 5f b9 90 ac 43 08 00 45 00  Pw..Um. _ ...C..E.
0010  00 6c 0e 9d 40 00 ff 32 e9 6e c0 a8 01 02 c0 a8  .l..@..2 .n.....
0020  01 01 00 00 07 d1 00 2d c8 20 45 00 00 40 00 00  .....- .E..@..
0030  40 00 fe fe f5 6a c0 a8 03 02 c0 a8 02 02 ff ff  @....j.....
0040  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0050  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0060  ff ff ff ff ff ff ff ff ff ff 01 02 02 04 9c 88  .....
0070  bb 57 71 95 a9 e7 28 a3 7a 45  .wq... (. zE
  
```

Figure 14. ESP packet with ESP SPI: 2001

Please note

Also please note, if you change a transform set definition, the change is applied only to crypto map entries that reference the transform set. The change is not applied to existing security associations, but is used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the `clear crypto sa` command.

Please see the example below:

In this example ISAKMP (IKE) is used instead manual keying. R1 and R2 have identical crypto configuration:

```

R1#sh run | s crypto
crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set ESP-NULL esp-null esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set ESP-NULL
  match address R1_TO_R2_ACL
crypto map TEST
  
```



```
R1#sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
51	IPsec	MD5	0	63721	63721	192.168.1.1
52	IPsec	MD5	63737	0	0	192.168.1.1
2002	IKE	MD5+DES	0	0	0	192.168.1.1

You can see that MD5 algorithm is used. Now change the transform-set only in R1 to ESP-DES instead ESP-NULL.

```
R1(config)#crypto ipsec transform-set ESP-NULL esp-des esp-md5-hmac
```

After this change, R1 and R2 have configured different algorithms, but tunnel still active and traffic is transmitted. Old algorithm is used.

```
R1#sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
51	IPsec	MD5	0	2063913	2063913	192.168.1.1
52	IPsec	MD5	2063929	0	0	192.168.1.1
2002	IKE	MD5+DES	0	0	0	192.168.1.1

Tunnel is active until end of lifetime.

```
R1#sh crypto ipsec sa | i remaining key
sa timing: remaining key lifetime (k/sec): (4194061/2730)
sa timing: remaining key lifetime (k/sec): (4194060/2730)
```

If we clear the current SA manually, then the new settings to take effect immediately.

```
R1#clear crypto sa
```

After reset SA, tunnel is down and warning message is shown,

```
R1#
*Sep 12 13:10:28.951: %CRYPTO-4-RECV_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
destaddr=192.168.1.1, prot=50, spi=0x6E1BD2D9(1847317209), srcaddr=192.168.1.2, input
interface=GigabitEthernet0
```

Now the transform-set configuration is changed in R2 and SA is cleared. Tunnel is up and has the new algorithm.

```
R1# sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
57	IPsec	DES+MD5	0	46464	46464	192.168.1.1
58	IPsec	DES+MD5	46484	0	0	192.168.1.1
2002	IKE	MD5+DES	0	0	0	192.168.1.1

Test #7

Cisco 892 switching performance in the case of IPSec with ESP DES encryption.

The ISAKMP (IKE) is used instead manual setting in next tests. Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set DES+MD5 esp-des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set DES+MD5
  match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.1
crypto ipsec transform-set DES+MD5 esp-des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.1
  set transform-set DES+MD5
  match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

You can verify crypto algorithm as shown below.

```
R1# sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
61	IPsec	DES+MD5	0	356269	505993	192.168.1.1
62	IPsec	DES+MD5	510723	0	0	192.168.1.1
2002	IKE	MD5+DES	0	0	0	192.168.1.1

Test results are present in the Table 6. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	5,6	99	98	8349	5,5	99	98	8168
64	6,9	99	98	8490	6,7	99	98	8193
128	12,1	99	98	9088	11,0	99	98	8317
256	21,4	99	98	9088	19,8	99	98	8434
512	39,7	99	98	9020	36,0	99	98	8191
1024	74,1	99	98	8720	66,5	99	98	7831
Random	22,0	78	75		20,0	79	75	

Table 7. Throughput and CPU utilization. Test #7. IPSec with ESP-DES

Captured IPSec packet is shown in the Figure 15. Now you can see that payload is encrypted. Total overhead is 44 bytes – new IP header 20 bytes + ESP header/trailer 24 bytes.

```

⊕ Frame 4007: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
⊖ Ethernet II, Src: Cisco_bb:55:6d (50:57:a8:bb:55:6d), Dst: Cisco_90:ac:43 (e0:5f:b9:90:ac:43)
  ⊕ Destination: Cisco_90:ac:43 (e0:5f:b9:90:ac:43)
  ⊕ Source: Cisco_bb:55:6d (50:57:a8:bb:55:6d)
  Type: IP (0x0800)
⊕ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
⊖ Encapsulating Security Payload
  ESP SPI: 0xcb19d3be (3407467454)
  ESP Sequence: 230122
  
```

```

0000  e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  . . . . CPW . . Um . . E.
0010  00 78 6b a1 40 00 ff 32 8c 5e c0 a8 01 01 c0 a8  . . . . . . . . . . /w
0020  01 02 |cb 19 d3 be| 00 03 82 ea dd 88 07 d3 2f 57  . . . . . . . . . .
0030  38 33 a2 f0 1d 8c 77 45 6d 18 4c 04 35 d1 5d 23  83 . . . . wE m. L. 5. ]#
0040  b0 88 2f 9d da 9a c4 a9 86 f8 67 49 90 4d e6 23  . . / . . . . . . gI. M. #
0050  ff e8 d5 c5 b5 d1 75 85 54 1d 71 ae 92 61 21 5e  . . . . . . u. T. q. . a! ^
0060  29 0b 3b 4d e1 e7 f7 d6 d5 8a 33 ee 19 0f ba 59  ). ; M. . . . . 3. . . . Y
0070  dc f7 18 8a 7c b9 a6 1e 1f 2b 31 19 5e 9b c3 de  . . . | . . . . +1. ^ . .
0080  11 07 26 fa d3 ae . . . . . . . . . . & . .
  
```

- New IP header
- EPS Payload
- ESP SPI identifier
- ESP Sequence number

Figure 15. IPSec packet with ESP-DES

Test #8

Cisco 892 switching performance in the case of IPSec with AH MD5 authentication and ESP DES encryption.

Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set MD5+DES ah-md5-hmac esp-des
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set MD5+DES
  match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.1
crypto ipsec transform-set MD5+DES ah-md5-hmac esp-des
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.1
  set transform-set MD5+DES
  match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

You can verify crypto algorithm as shown below.

```
R1#sh crypto engine connections active
Crypto Engine Connections

  ID  Type   Algorithm      Encrypt  Decrypt  LastSeqN  IP-Address
  ---  ---    -
   3  IPsec  MD5+DES        0        7057503  7057503  192.168.1.1
   4  IPsec  MD5+DES      7121355         0         0  192.168.1.1
 2001  IKE    MD5+DES         0         0         0  192.168.1.1
```

Test results are present in the Table 8. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	5,1	99	98	7572	4,6	99	98	6786
64	5,8	98	97	7140	5,7	97	96	6947
128	10,3	99	98	7728	9,3	99	98	6999
256	18,2	99	98	7729	16,5	99	98	7020
512	33,7	99	98	7666	31,0	99	98	7053
1024	63,6	99	98	7490	57,5	99	98	6772
Random	20,0	78	75		19,0	80	76	

Table 8. Throughput and CPU utilization. Test #8. IPsec with AH authentication and ESP DES payload.

Structure of IPsec packet with AH authentication and ESP payload is shown in the Figure 16.

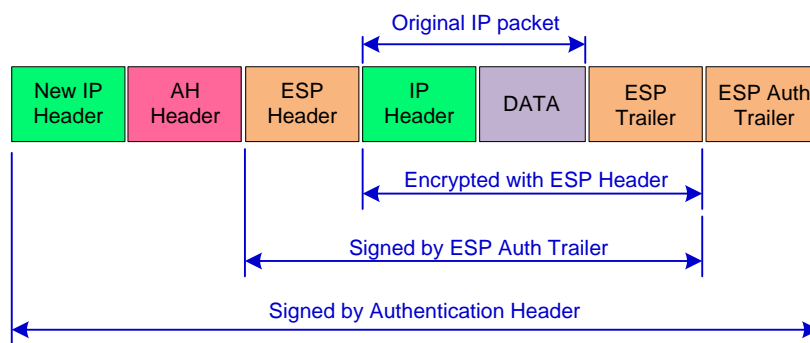


Figure 16. Format of packet with AH + ESP

Captured IPsec packet is shown in the Figure 11. Total Overhead 68 bytes – new IP header 20 bytes + AH header 24 bytes + ESP fields 24 bytes.

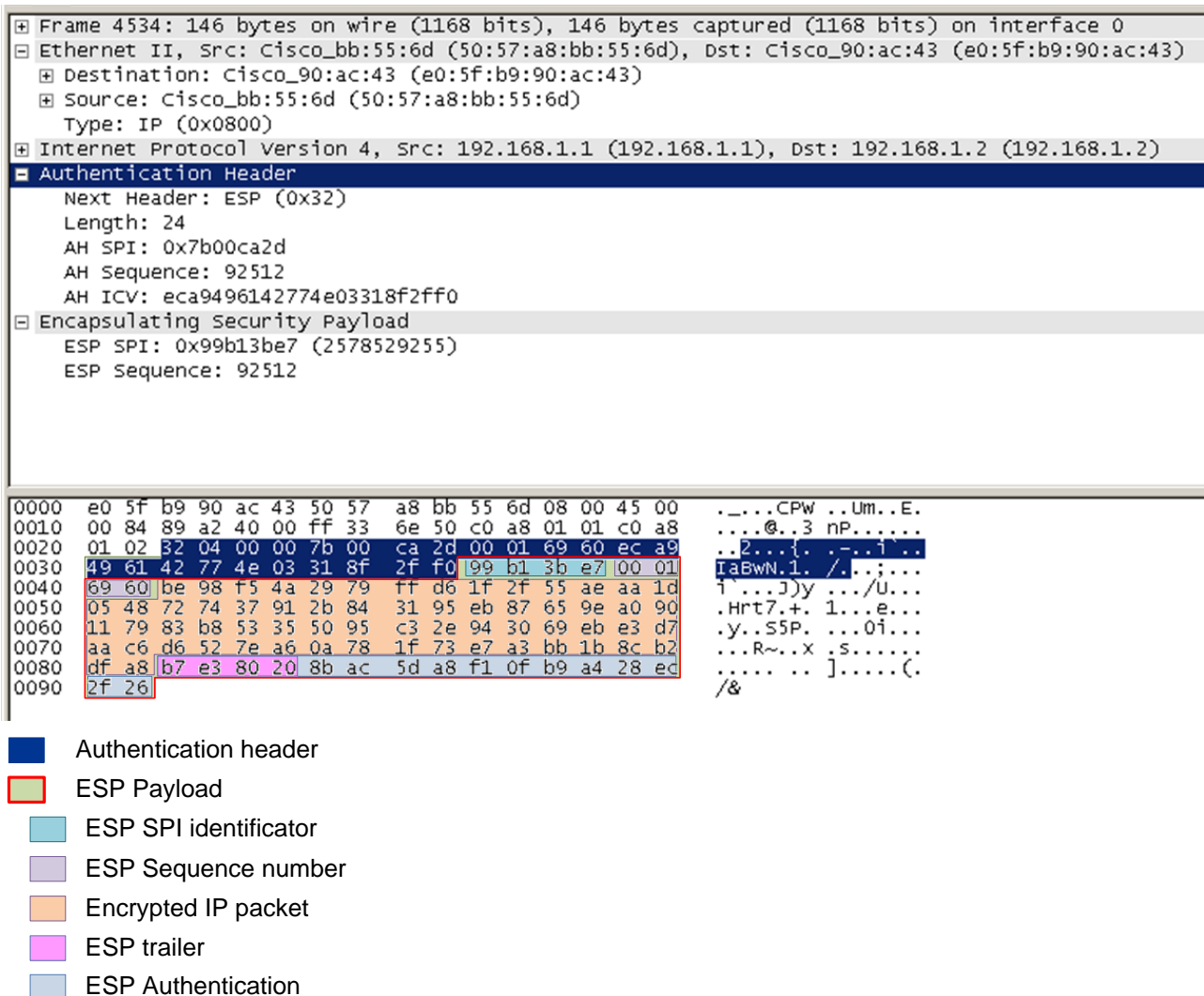


Figure 17. IPsec packet with AH and ESP.

Test #9

Cisco 892 switching performance in the case of IPSec with AH MD5 authentication + ESP payload with DES encryption and MD5 authentication.

Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set MD5+DES+MD5 ah-md5-hmac esp-des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set MD5+DES+MD5
  match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2# sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.1
crypto ipsec transform-set MD5+DES+MD5 ah-md5-hmac esp-des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.1
  set transform-set MD5+DES+MD5
  match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

You can verify crypto algorithm as shown below.

```
R1#sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
7	IPsec	MD5+DES+MD5	0	8614	32873	192.168.1.1
8	IPsec	MD5+DES+MD5	34049	0	0	192.168.1.1
2001	IKE	MD5+DES	0	0	0	192.168.1.1

Test results are present in the Table 9. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	4,2	99	98	6289	4,0	99	98	6014
64	5,2	98	97	6423	5,0	99	98	6067
128	8,4	99	98	6358	8,1	99	98	6073
256	14,9	99	98	6329	14,2	99	98	6057
512	27,6	99	98	6274	26,1	99	98	5929
1024	51,7	99	98	6080	49,2	99	98	5795
Random	17,0	78	75		16,0	79	76	

Table 9. Throughput and CPU utilization. Test #9. IPSec with AH + DES + MD5.

Captured IPSec packet is shown in the Figure 18. Total Overhead 80 bytes → new IP header 20 bytes + AH header 24 bytes + ESP payload 100 bytes – 64 bytes of original packet.

```

⊞ Frame 3144: 158 bytes on wire (1264 bits), 158 bytes captured (1264 bits) on interface 0
⊞ Ethernet II, Src: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d), Dst: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43)
⊞ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
⊞ Authentication Header
  Next Header: Encap Security Payload (0x32)
  Length: 24
  AH SPI: 0x0f013f05
  AH Sequence: 737325
  AH ICV: d3aa96fe85e246526bd8bd5f
⊞ Encapsulating Security Payload
  ESP SPI: 0xc7c05999 (3351271833)
  ESP Sequence: 737325

0000  e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  ._.CPW ..Um..E.
0010  00 90 e1 7b 40 00 ff 33 16 6b c0 a8 01 01 c0 a8  ...{@.3 .k.....
0020  01 02 32 04 00 00 0f 01 3f 05 00 0b 40 2d d3 aa  ..2....?..@-..
0030  96 fe 85 e2 46 52 6b d8 bd 5f c7 c0 59 99 00 0b  ...FRk. .Y...
0040  40 2d 4b 5e 9d 60 4e 4b ad 59 53 09 41 0b 85 5f  @-K^ .NK .YS.A...
0050  33 70 7b 1f 26 35 ca bf 53 e9 68 6a c0 ec 22 9e  3p{.&5.. S.hj..".
0060  c7 6c 63 00 21 67 53 52 8e 0b de a0 ac 40 a8 57  .lc!gSR .....@.w
0070  c5 77 f6 cd be 85 7c b6 f7 6b d2 f3 f3 a6 08 95  .w...|. .k.....
0080  6c dd 93 53 a7 39 27 31 8f e6 fd c2 ab df fd f2  l..s.9'1 .....
0090  62 6f ad 35 64 b9 d9 9a 8a 2e ff 88 54 d6      bo.5d... ....T.
  
```

- AH header 24 bytes
- EPS Payload 100 bytes

Figure 18. IPSec packet with AH + ESP DES + MD5.

Test #10

Cisco 892 switching performance in the case of IPSec with AH MD5 authentication + ESP payload with 3DES encryption and MD5 authentication.

Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|GigabitEthernet0|R1_TO_R2_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  encr aes 256
  hash md5
  authentication pre-share
  group 19
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set MD5+3DES+MD5 ah-md5-hmac esp-3des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set MD5+3DES+MD5
  match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s crypto|GigabitEthernet0|R2_TO_R1_ACL

crypto pki token default removal timeout 0
crypto isakmp policy 10
  encr aes 256
  hash sha512
  authentication pre-share
  group 16
crypto isakmp key CISCO address 192.168.1.1
crypto ipsec transform-set MD5+3DES+MD5 ah-md5-hmac esp-3des esp-md5-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.1
  set transform-set MD5+3DES+MD5
  match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

Test results are present in the Table 10. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	4,2	99	98	6289	4,0	99	98	5996
64	5,2	98	98	6420	5,0	99	98	6082
128	8,5	99	98	6371	8,1	99	98	6079
256	15,0	99	98	6375	14,4	99	98	6101
512	27,8	99	98	6308	26,2	99	98	5949
1024	52,7	99	98	62,02	49,6	99	98	5840
Random	16,0	74	71		15,0	76	72	

Table 10. Throughput and CPU utilization. Test #9. IPSec with AH + DES + MD5.

Captured IPSec packet is shown in the Figure 19. Total Overhead 80 bytes → new IP header 20 bytes + AH header 24 bytes + ESP payload 100 bytes – 64 bytes of original packet.

```

⊞ Frame 8916: 158 bytes on wire (1264 bits), 158 bytes captured (1264 bits) on interface 0
⊞ Ethernet II, Src: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43), Dst: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d)
⊞ Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)
  Authentication Header
    Next Header: Encap Security Payload (0x32)
    Length: 24
    AH SPI: 0xfc954919
    AH Sequence: 5000040
    AH ICV: f145dba5037f60d3e46c41b1
  Encapsulating Security Payload
    ESP SPI: 0xf3c0a6ac (4089489068)
    ESP Sequence: 5000040
0000  50 57 a8 bb 55 6d e0 5f b9 90 ac 43 08 00 45 00  Pw..Um._ ...C..E.
0010  00 90 22 f1 40 00 ff 33 d4 f5 c0 a8 01 02 c0 a8  ..".@..3 .....
0020  01 01 32 04 00 00 fc 95 49 19 00 4c 4b 68 f1 45  ..2....I..LKh.E
0030  cb a5 03 7f 60 d3 e4 6c 41 b1 f3 c0 a6 ac 00 4c  ....1A.....L
0040  4b 68 df f2 9b cb 88 dd 0b 72 e8 b0 12 72 62 21  Kh.....r...rb!
0050  8b 29 37 a2 e3 36 e0 a4 ca fc 4a 20 78 46 42 42  .)7..6...J xFBB
0060  53 ce 78 5a 12 ee 98 ca ec 9c 49 3c 89 4b 53 85  S,xZ....I<.KS.
0070  17 7d 9c f4 84 9c 4a a1 ae 54 93 e7 4f 8b ab e0  .}....J. .T..O...
0080  70 21 06 ff 7a 78 50 64 8b b9 84 00 dd 82 2d 6e  p!..zxPd .....-n
0090  9f 38 06 59 3b 22 ac 57 cb d0 1c e2 ae 57      .8.Y;"..w .....W
  
```

■ AH header 24 bytes
□ EPS Payload 100 bytes

Figure 19. IPSec packet with AH + ESP 3DES + MD5.

Test #11

*Cisco 892 switching performance test with strong encryption algorithm.
IPSec with AH SHA-512 authentication + ESP payload with AES encryption and SHA-512 authentication.*

Part of R1 and R2 configuration is presented below:

```
R1#sh run | s crypto|R1_TO_R2_ACL|GigabitEthernet0

crypto pki token default removal timeout 0
crypto isakmp policy 10
  encr aes 256
  hash sha512
  authentication pre-share
  group 16
crypto isakmp key CISCO address 192.168.1.2
crypto ipsec transform-set HIGH ah-sha512-hmac esp-aes esp-sha512-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.2
  set transform-set HIGH
match address R1_TO_R2_ACL
interface GigabitEthernet0
  ip address 192.168.3.1 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R1_TO_R2_ACL
  permit ip host 192.168.2.2 host 192.168.3.2
```

```
R2#sh run | s cry|R2_TO_R1_ACL|GigabitEthernet0

crypto pki token default removal timeout 0
crypto isakmp policy 10
  encr aes 256
  hash sha512
  authentication pre-share
  group 16
crypto isakmp key CISCO address 192.168.1.1
crypto ipsec transform-set HIGH ah-sha512-hmac esp-aes esp-sha512-hmac
crypto map TEST 10 ipsec-isakmp
  set peer 192.168.1.1
  set transform-set HIGH
match address R2_TO_R1_ACL
interface GigabitEthernet0
  ip address 192.168.1.2 255.255.255.0
  duplex auto
  speed auto
  crypto map TEST
ip access-list extended R2_TO_R1_ACL
  permit ip host 192.168.3.2 host 192.168.2.2
```

You can verify crypto algorithm as shown below.

```
R1#sh crypto engine connections active
Crypto Engine Connections
```

ID	Type	Algorithm	Encrypt	Decrypt	LastSeqN	IP-Address
1003	IPsec	SHA512+AES+SHA512	0	259186	259186	192.168.1.1
1004	IPsec	SHA512+AES+SHA512	259187	0	0	192.168.1.1
2002	IKE	SHA512+AES256	0	0	0	192.168.1.1

Test results are present in the Table 11. During the test with random packets, the bit rate was set such that the CPU load was approximately 75%.

Packet Length [Bytes]	1 st variant Fa8 – Gi0				2 nd variant VLAN1 - Gi0			
	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec	Measured L1 Rate [Mbps]	CPU total usage [%]	CPU usage caused by traffic [%]	Measured Rate packets/sec
40	0,9	99	26	1338	0,8	95	26	1250
64	1,0	99	25	1285	1,1	97	26	1288
128	1,6	99	24	1205	1,5	98	25	1165
256	2,5	99	22	1083	2,5	99	23	1058
512	4,0	99	19	904	3,9	99	19	888
1024	5,9	99	15	689	5,8	99	16	682
Random	3,4	75	17		3,3	75	17	

Table 11. Throughput and CPU utilization. Test #9. IPSec with AH + DES + MD5.

Strong encryption algorithms are big load for the CPE. You can see that total CPU usage is very high, but most of time resources are not used for data transmission. Switching performance is very low.

Captured IPSec packet is shown in the Figure 20.

```

Frame 44573: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
Ethernet II, Src: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d), Dst: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43)
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
Authentication Header
  Next Header: Encap Security Payload (0x32)
  Length: 44
  AH SPI: 0x56b65ce9
  AH Sequence: 44570
  AH ICV: b1bb151f087019f5f96fa8f13e09e08be1adf68e056a9f45...
Encapsulating Security Payload
  ESP SPI: 0x67cf64dd (1741645021)
  ESP Sequence: 44570
0000  e0 5f b9 90 ac 43 50 57 a8 bb 55 6d 08 00 45 00  . . . . CPw ..Um. .E.
0010  00 c8 a0 d6 40 00 ff 33 56 d8 c0 a8 01 01 c0 a8  . . . . @.3 V.....
0020  01 02 32 09 00 00 56 b6 5c e9 00 00 ae 1a b1 bb  ..Z...V. \.....
0030  15 1f 08 70 19 f5 f9 6f a8 f1 3e 09 e0 8b e1 ad  ..p...o .>.....
0040  f6 8e 05 6a 9f 45 05 47 f3 99 a8 f5 10 70 67 cf  ..].E.G .....pg.
0050  64 dd 00 00 ae 1a 62 fc b0 d2 af 9f 08 47 df 35  d.....b. ....G.5
0060  44 bb f4 fb 8f 4a a8 10 5c 00 4b 54 81 bc 59 09  D.....J.. \.KT..Y.
0070  60 34 23 62 50 9d a9 f8 20 a0 a4 20 89 8c 71 10  `4#bP... ..q.
0080  b4 d1 5e d3 21 cd 1d 2d f5 c7 98 c4 95 c7 35 2f  ..^!!-- .....5/
0090  57 84 15 15 90 51 84 5e e3 0b 15 36 7b e2 07 95  w....Q.^ ...6{...
00a0  b4 a9 8d fd 5c 96 4a f5 5b 16 f7 55 aa 5b 7e f1  ....\..J. [.U.[~.
00b0  54 ce 39 60 6e b3 6a 79 35 7c 2b e8 2c b6 1f 53  T.9`n.jy 5|+,...S
00c0  d9 15 98 b2 2b a7 53 2e 5a 37 49 dc 43 c2 d6 de  ....+.S. Z7I.C...
00d0  fd 94 cb 77 93 84  . . . . w..
  
```

- AH header 44 bytes
- EPS Payload 136 bytes

Figure 20. IPSec packet with SHA512+AES+SHA512

Total overhead for original 64 byte packet:
 IPSec frame on wire 214 bytes– original frame on wire 78 bytes= 136bytes.

Total overhead for original 1000 byte packet:
 IPSec frame on wire 1142 bytes - original frame on wire 1014 bytes = 128 bytes.

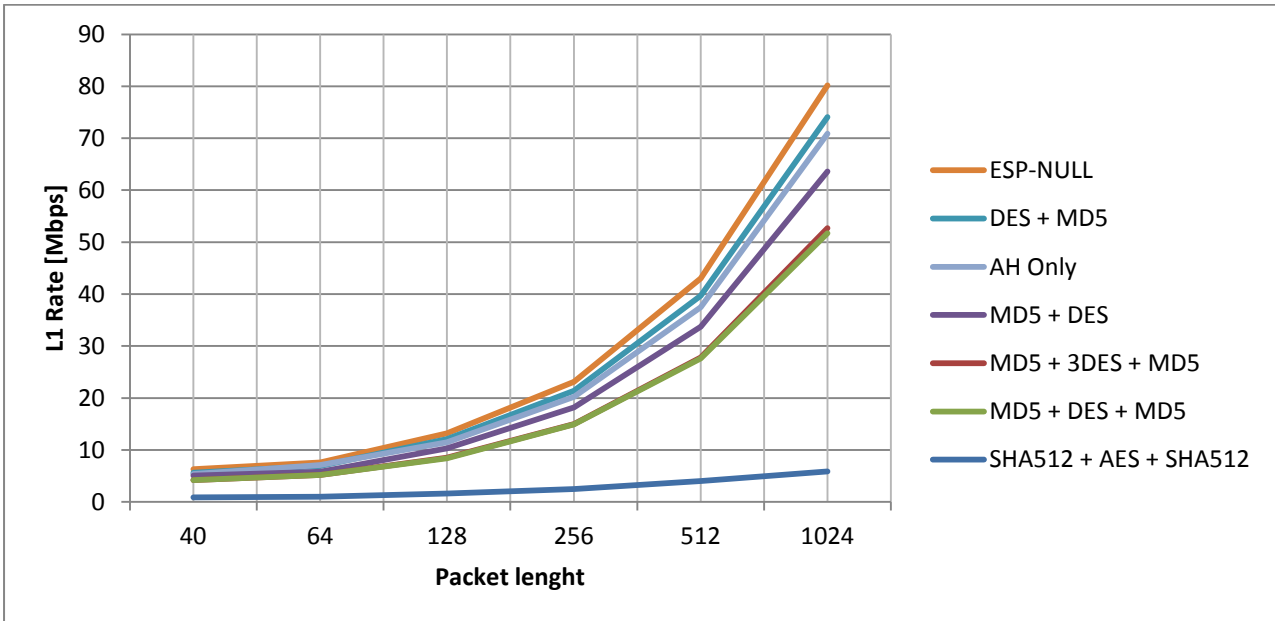
You can see that strong encryption algorithm gives up to 213% overhead for short packets.

IPSec test summary

Router performance depends from encryption algorithm and length of packet. Router has different throughput between different ports, due to the schematic features of router. Difference is not so big but it can be up to 10 Mbps with light encryption and long packets.

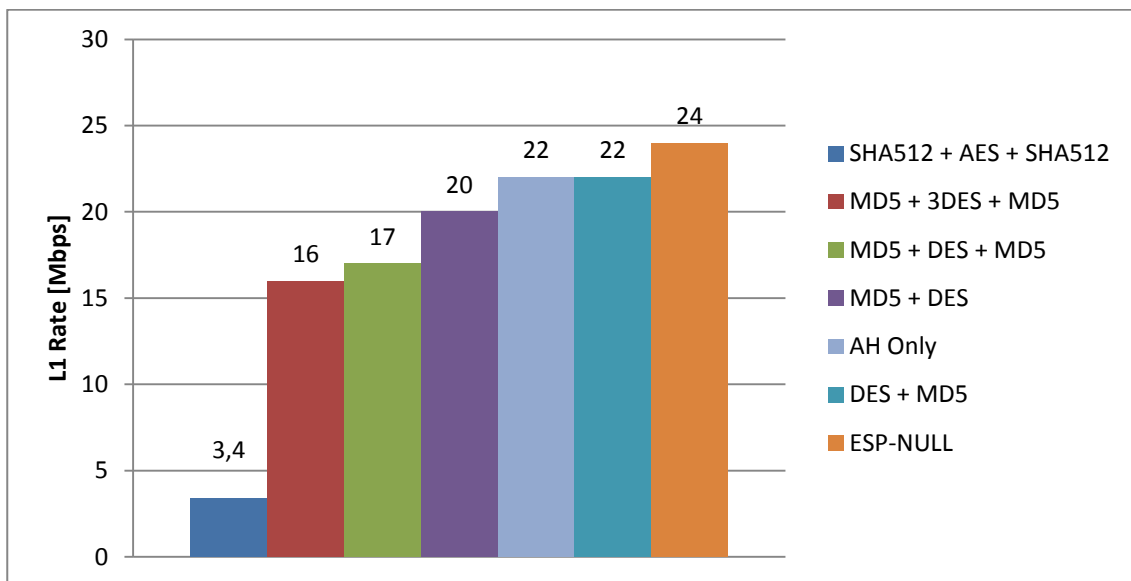
Only a few encryption options were tested. However, this provides sufficient vision of performance. On the graph are results of the easiest and most difficult encryption algorithm. All other variations are between them.

Summary throughput results for IPSec RFC2544 test are present in the Graph 5.



Graph 5. IPSec tunnel throughput. Summary results.

Graph 6 shows IPSec performance for random packets. This gives more close result to a real throughput.



Graph 6. IPSec tunnel throughput for random packets.

Appendix

40 byte packets

Minimum payload length for Ethernet frame is 46 bytes. If necessary, the data field should be padded (with octets of zero) to meet the Ethernet minimum frame size. This padding is not part of the IP packet and is not included in the total length field of the IP header.

Router processes the data only with length which specified in the Total Length field of IP header. New padding is added for each short packet.

Please see below result of the small test, how router rewrites padding. Schematic diagram is present on Figure 21, Wireshark captured result are present in the Figure 22 - 24. Tester 1 is configured for sending of 40 byte packets. Packet is padded with 0xFF pattern (please see Figure 22).

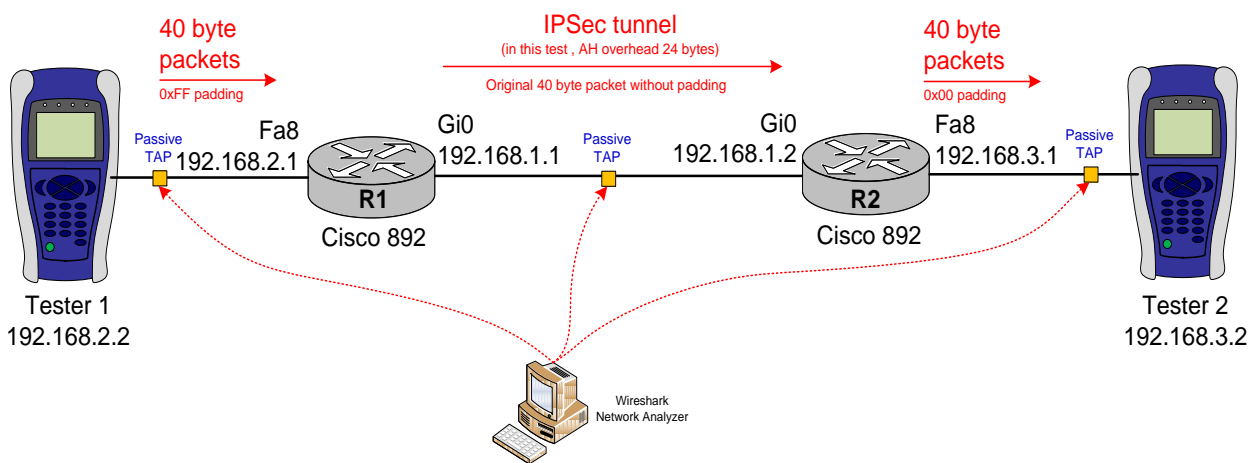


Figure 21. Schematic diagram for packet padding test

```
Frame 4755: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: 00:00:00_00:00:11 (00:00:00:00:00:11), Dst: 50:57:a8:bb:55:55 (50:57:a8:bb:55:55)
Internet Protocol Version 4, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.3.2 (192.168.3.2)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
  Total Length: 40
  Identification: 0x0000 (0)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: Unknown (254)
  Header checksum: 0xb383 [validation disabled]
  Source: 192.168.2.2 (192.168.2.2)
  Destination: 192.168.3.2 (192.168.3.2)
  [Source GeoIP: unknown]
  [Destination GeoIP: Unknown]
Data (20 bytes)
  Data: ffffffff
  [Length: 20]
0000  50 57 a8 bb 55 55 00 00 00 00 00 11 08 00 45 00  Pw..UU.. .....E.
0010  00 28 00 00 40 00 40 fe b3 83 c0 a8 02 02 c0 a8  (...@.@, .....
0020  03 02 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
0030  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
```

Figure 22. Frame was transmitted by Tester 1

Packet padding is shown in the orange frame.

IPSec tunnel is up between R1 and R2. AH authentication without ESP encryption is used in this test. AH protocol overhead adds 24 bytes. New L2 payload is 64 bytes, 24 byte AH header + 40 byte IP packet. In this case router does not add padding to packet. Please see the Figure 23

```

Frame 6731: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: e0:5f:b9:90:ac:43 (e0:5f:b9:90:ac:43), Dst: 50:57:a8:bb:55:6d (50:57:a8:bb:55:6d)
Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)
Authentication Header
Internet Protocol Version 4, Src: 192.168.3.2 (192.168.3.2), Dst: 192.168.2.2 (192.168.2.2)
Data (20 bytes)
Data: ffffffffffffffffffffffffffffffffffffffffffffffff
[Length: 20]

0000  50 57 a8 bb 55 6d e0 5f  b9 90 ac 43 08 00 45 00  Pw..Um._ ...C..E.
0010  00 54 b9 56 40 00 ff 33  3e cc c0 a8 01 02 c0 a8  .T.V@..3 >.....
0020  01 01 04 04 00 00 00 00  07 d1 02 f1 a5 4c 0e a9  ..L..
0030  11 38 3c a2 ec 65 d8 a0  2a 29 45 00 00 28 00 00  8<..e..*)E..(..
0040  40 00 fe fe f5 82 c0 a8  03 02 c0 a8 02 02 ff ff  @.....
0050  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  .....
0060  ff ff

```

Figure 23. Frame was transmitted between R1 and R2

Orange frame shows original DATA (20 byte) without padding. Blue fill shows AH header (24 bytes).

After packets decryption, R2 forwards packets to Tester 2, but in this case payload again is shorter than 46 bytes and router adds the padding 0x00 (please see the Figure 24).

```

Frame 4034: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: e0:5f:b9:90:ac:2b (e0:5f:b9:90:ac:2b), Dst: 00:80:16:8c:06:78 (00:80:16:8c:06:78)
Destination: 00:80:16:8c:06:78 (00:80:16:8c:06:78)
Source: e0:5f:b9:90:ac:2b (e0:5f:b9:90:ac:2b)
Type: IP (0x0800)
Padding: 00000000000000
Internet Protocol Version 4, Src: 192.168.2.2 (192.168.2.2), Dst: 192.168.3.2 (192.168.3.2)
Version: 4
Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 40
Identification: 0x0000 (0)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 62
Protocol: Unknown (254)
Header checksum: 0xb583 [validation disabled]
Source: 192.168.2.2 (192.168.2.2)
Destination: 192.168.3.2 (192.168.3.2)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Data (20 bytes)
Data: ffffffffffffffffffffffffffffffffffffffffffffffff
[Length: 20]

0000  00 80 16 8c 06 78 e0 5f  b9 90 ac 2b 08 00 45 00  ....x._ ...+.E.
0010  00 28 00 00 40 00 3e fe  b5 83 c0 a8 02 02 c0 a8  .(.@.>.....
0020  03 02 ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  .....
0030  ff ff ff ff ff ff 00 00  00 00 00 00 00 00 00 00  .....

```

Figure 24. Frame was transmitted from R2 to Tester 2