



# IPsec Simplified — Part 2

**Peter J. Welcher**

---

## Introduction

I'm writing this article in warm Sao Paulo, Brazil. I'm here for a week to teach the internal DCIINS class for Cisco. I'm enjoying my visit so far. Next week I should be in Mexico City next week doing the same, barring surprises.

Upon looking at my list of articles, I'm surprised to see that this is article number 70. And I'm still enjoying writing them!

Last month's article covered some basic terminology and concepts relating to IPsec. See <http://www.netcraftsmen.net/welcher/papers/ipsec1.html>. This month we'll continue with IPsec. My goal is to talk through a very basic configuration for IPsec. Unless you really have a complex situation, this may be all you need to turn on IPsec in your network.

## IPsec Security Association Choices

The previous article did not have space to cover some basic choices you have to make when deciding how to run IPsec.

One of the choices is whether you wish to use Authentication Header (AH) or Encapsulating Security Payload (ESP). Each of these is an IP protocol, just as TCP and UDP are. The protocol codes are 51 and 50, for AH and ESP respectively. Thus IPsec packets will normally have 50 or 51 in the IP protocol field, and there will be an AH or ESP header between the IP header and the payload data.



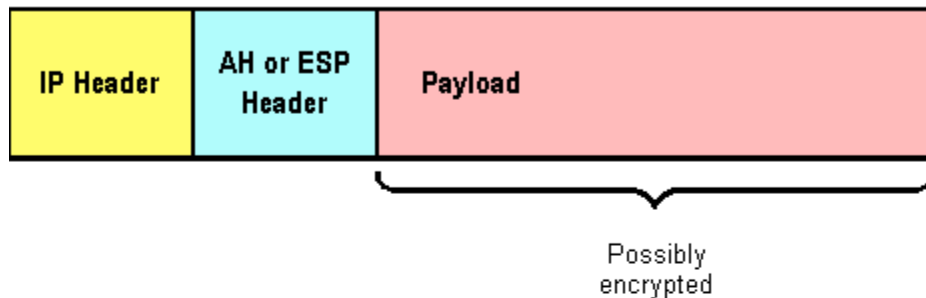
With AH, an HMAC keyed hash is transported in the AH header. That provides data integrity and source authentication: the data must come from an authentic source, one that knows the hash key. And

the data cannot be changed by anyone "in the middle", since the keyed hash allows detection of any changes. Note however that AH does not provide for encryption of the actual data. So you would only use AH for things like placing small orders across the Internet, assuming that does not need to be done confidentially.

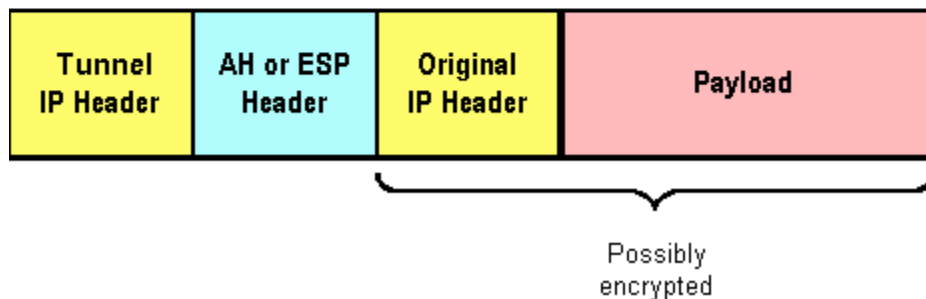
With ESP, you get the HMAC keyed hash plus encryption of the data. Generally, this is what we want when we run IPsec.

The other important choice as to how we do IPsec is tunnel versus transport mode. With **transport** mode, the original IP header sources and destinations are kept in the IPsec IP packet header. The AH or ESP header is slipped in between the IP header and the payload. With **tunnel** mode, a new IP header is applied, with source and destination being the routers or other devices that form the encrypted tunnel endpoints. The original IP header follows the AH or ESP header, and is encrypted if ESP is being used.

## Transport Mode



## Tunnel Mode



Normally, you might run transport mode between two computers with IPsec (however, they might also use tunnel mode, particularly if one end of the IPsec conversation is a router). Tunnel mode is what you'd use between two routers, encrypting all traffic between the LAN's they connect.

The security point to tunnel mode is that it prevents some forms of traffic analysis. For example, if a hacker sees packets from many hosts on many subnets all going to a small number of addresses, the hacker may correctly assume the common destination(s) are servers of some kind. With tunneling, all packets might be from a router and to a router. The hacker can still see that all traffic is going to a common site, and reason that that site is your data center, but they probably knew that anyway.

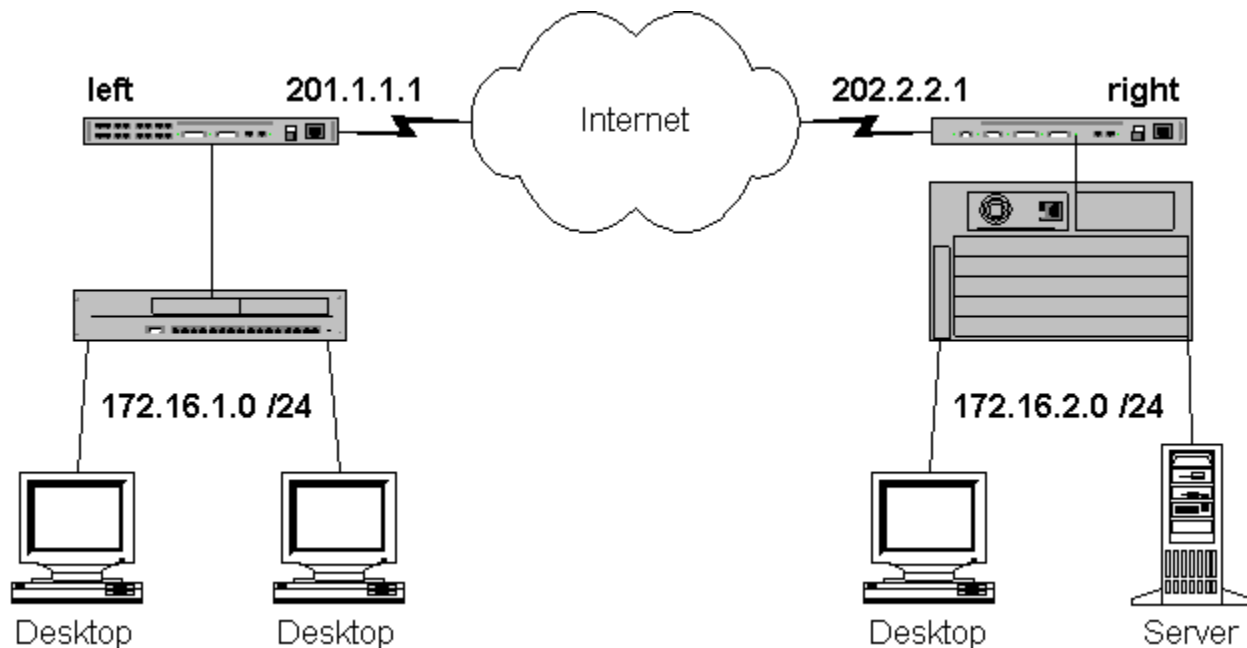
By the way, IPsec through a NAT point can cause problems, because under transport mode the source and destination IP addresses are protected by the keyed hash (and under tunnel mode, they are

encrypted). Thus the NAT point cannot just go "fix up" the addresses. There is currently discussion of a standard way to make this work. I would try to design the network so the IPsec can terminate outside or before the NAT point.

## Configuring IPsec

In this section we'll talk through a basic IPsec configuration. We're going to use a simple setting, where two sites are connected by IPsec. We'll tunnel between router "left" and router "right", hiding the private addressing for the host computers at the two sites.

Here is the diagram that goes with the configuration:



In the previous article, we looked at the major steps a router goes through in establishing an IPsec Security Association using IKE. (If you don't recognize some of those terms, please review the previous article!) The steps in configuring the router tell it which options to use for each step, so it is not too surprising that the steps you use for router configuration resemble the operational steps in the router. Of course, as always, there are many options you can also configure, but we're not going to go over all of those!

Here's the list of major steps in configuring IPsec:

**Step 1:** Ensure that any packet filters are compatible with IKE and IPsec.

**Step 2:** Define crypto access lists for what traffic to encrypt/decrypt (one such access list for each different Security Association).

**Step 3:** Specify IKE options, especially authentication method.

**Step 4:** Define transform sets, for the IPsec encryption, hashing, and other choices.

**Step 5:** Define a crypto map, tying peers to SA options and crypto access lists.

**Step 6:** Apply the crypto map to an interface.

Let's talk about these in turn, complete with configuration commands.

**Step 1.** I've included Step 1 because it is a very real "gotcha" here: you're probably running IPsec on a router that's connected to the Internet (for VPN purposes, or VPDN access). Any such router probably has a security packet filter access list (ACL) on the interface connecting to the Internet. You do need to make sure that packet filter ACL allows IKE and IPsec packets into the router, or the router won't be able to begin to do IPsec. In order to do this, you need to know that IKE uses UDP port 500. The IPSec ESP and AH protocols use IP protocol numbers 50 and 51, respectively. Extended access lists allow the use of the "esp" and "ahp" keywords (like "tcp" or "udp"), so you don't really have to remember the numbers 50 and 51. (But they're obviously factoids that might appear as questions on any quizzes or tests about this.)

Sample configuration portion from router right:

```
interface Serial0
 ip address 202.2.2.1 255.255.255.0
 ip access-group 111 in
 no ip directed-broadcast

! Before the firewall will allow traffic initiated on the
! outside into this site, that traffic must satisfy this list
access-list 111 permit udp host 201.1.1.1 host 202.2.2.1 eq 500
access-list 111 permit esp host 201.1.1.1 host 202.2.2.1
access-list 111 permit ahp host 201.1.1.1 host 202.2.2.1
access-list 111 permit ip 172.16.1.0 0.0.0.255 172.16.2.0 0.0.0.255
```

The left router would need to be configured similarly.

Remember that this is a minimal example. We might want to allow SNMP, syslog, and TFTP traffic, from the left router to a management station. That traffic should also be encrypted. But that would increase the scope and complexity of our example. Let's do **simple** first!

**Step 2.** We need at least one access-list defining "interesting traffic". The statements with permit describe the traffic that is to be encrypted by a specific Security Association. Deny means do not encrypt, at least not with the current SA.

The access list also specifies what traffic gets encrypted according to which rules, as in "traffic to A uses the Security Association for A, and traffic to B uses the SA for B". In our case, the rule is simple: traffic from the right subnet to the left subnet gets encrypted going out interface serial 0 (and traffic from left to right decrypted, based on the same access list).

```
access-list 120 permit ip 172.16.2.0 0.0.0.255 172.16.1.0 0.0.0.255
```

We only need access list 120, since there will only be one Security Association here.

**Step 3.** Our configuration also needs to tell the router how we wish to do IKE: what IKE choices are acceptable, and the order we prefer them in. And how to authenticate with each peer. This is our IKE policy.

```
! Define IKE Policies. All will be offered to the Peer
! and the most secure match will be used. Bigger preference
! number is "most secure".
```

```
crypto isakmp policy 1
  hash md5
  authentication pre-share

! If the peer can accept the following policy, then it will be used as
! it is more secure than Policy 1
crypto isakmp policy 2
  authentication pre-share
  group 2
  lifetime 360

! Define the Pre-Shared Authentication Key of the Peers
crypto isakmp key itsnotverysecret address 201.1.1.1
```

**Step 4.** We then have to tell the router what IPsec encryption and hashing techniques to use, and whether to use IPsec tunnel or transport mode. Different peers or SAs can use different encryption and other choices. Each set of choices is a transform set. Below we define two, named **mydessha** and **mydesmd5**:

```
! IPsec policies are defined here. These include your AH
! and ESP choices as well as the mode of operation.
crypto ipsec transform-set mydessha esp-des esp-sha-hmac
crypto ipsec transform-set mydesmd5 esp-des esp-md5-hmac
```

We have chosen tunnel mode here (the default). Add "mode transport" to these lines to configure transport mode.

**Step 5.** The next step is tying together all the IPsec SA options. This is done with a **crypto map**, which tells the Cisco router which peer or peers use which transform set, in other words what the IPsec options are for that peer or those peers, in order of preference. The crypto map also tells the router which crypto access list defines the interesting traffic to encrypt using that SA's encryption key.

So the crypto map ties together a list of router peers for an SA, a transform set, and a crypto access list. You can think of this as tying together **who** the peers are, **how** to encrypt, and **what** to encrypt. The crypto map can have several sub-rules, marked with sequence numbers. I think of these as blocks or chunks, where each chunk ties together peer or peers, crypto access list and transform sets.

```
crypto map theSimpleCryptoMap 10 ipsec-isakmp
  set peer 201.1.1.1
  set transform-set mydesmd5 mydessha
  match address 120
```

Note that we have specified two acceptable transform sets. These are attempted in order from left to right (so the SA with a peer may use the best match, where we specify what we mean by "best").

**Step 6.** The final step is to apply the crypto map to an interface: **where** to encrypt. (**When** and **why** don't really fit this story, do they? Well, you get **when** if you use a time-based access list, which would be rare, I suspect. And **why** goes in the documentation).

```
interface Serial0
  crypto map theSimpleCryptoMap
```

If you've been paying attention, you will have noticed that for this to provide a VPN between the two sites, we need routes for those private subnets. One way to get this would be to configure static routes with the peer router as next hop. Because the router will encrypt and tunnel actual data packets as they go out the serial interface, there won't be problems with private addresses that are not routable on the

Internet. Another possibility is to use the neighbor statement with your interior routing protocol, and IPsec tunnel the unicast traffic that results.

In case you're wondering, what about more peers? Well, you can use multiple "set peer ..." commands in a block of your crypto map. Of course, you need to pick up the appropriate source/destination pairs in your crypto access list. Here's what a more complex crypto map might look like:

```
crypto map theComplexCryptoMap 10 ipsec-isakmp
  set peer 201.1.1.1
  set peer 203.3.3.1
  set transform-set mydesmd5 mydessha
  match address 120
```

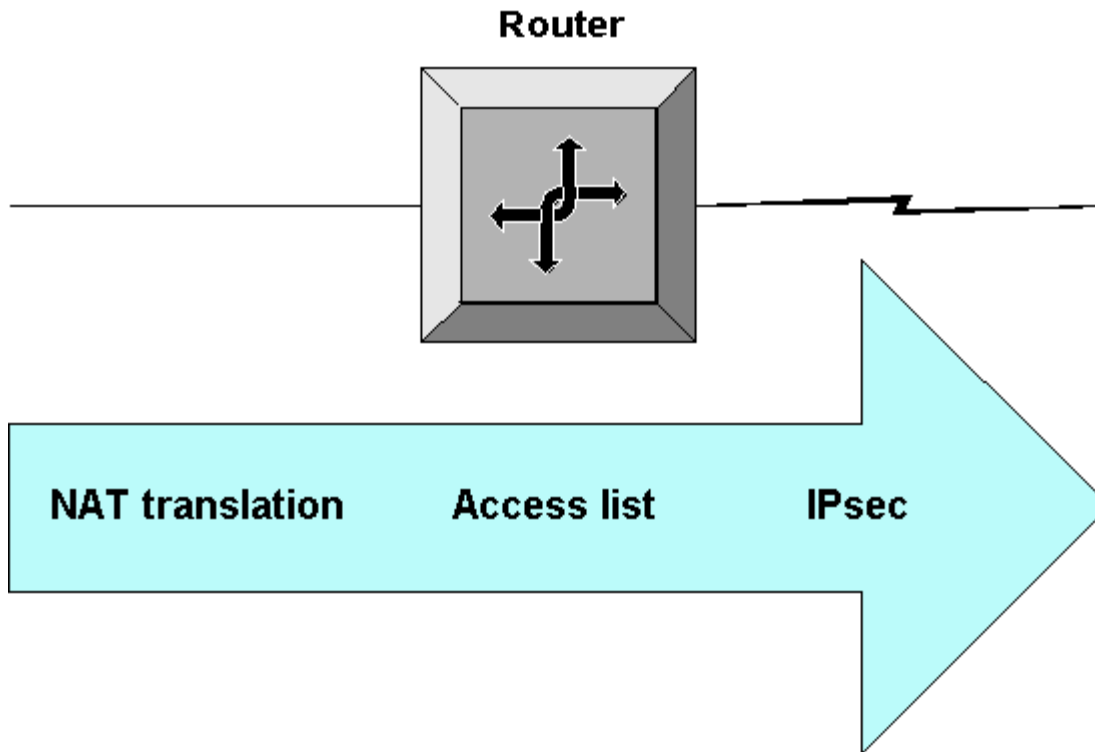
```
crypto map theComplexCryptoMap 20 ipsec-isakmp
  set peer 204.4.4.1
  set transform-set myotherrule
  match address 121
```

Here peer 204.4.4.1 is another vendor's router or a router that for some reason needs a different transform set. We also build a different crypto access list, since the traffic to and from that router is being encrypted using different rules.

## Order of Operations

There's one little thing we sort of assumed in doing that configuration. Apologies if it was bothering you; I was trying to keep things simple. I built the assumption in when we set up access list 111.

You really need to know the order in which operations are applied by a router, when it is doing access lists and IPsec. We'll add in NAT address translation, since that is often going to part of what you do when you use IPsec for Internet-based connectivity.



The key thing here is that both security and IPsec access lists need to use the outside (translated) addresses, if NAT is present. As packets enter the router (from the right, in the drawing), first IPsec decryption is done, then any inbound access list applied, and finally NAT translation takes place.

See the following link for a more detailed version of the order of operations. (There's one difference from what I've just said: can you spot it? Better yet, test it and let me know what you find out).

<http://www.cisco.com/warp/public/556/5.html>

## In Conclusion

I didn't make a big deal about it above, but SA's are one way, so generally two IPsec parties form two SA's, one in each direction. That doesn't change the above description of how it works or how to configure it, but it is a technically more accurate way to describe what's going on.

If you look again at **Step 2** above, this is where it really helps to have summarizable blocks of subnets. If you don't know what subnets are where, forget about doing IPsec. If you have messy lists of subnets, you can configure IPsec but the access lists may get a little nasty. (Access List Manager in CiscoWorks 2000 might help). If you have single subnets or summarizable blocks of subnets, then a very short access list may do what is needed. Bonus: it might be easier to understand and troubleshoot, too!

There's a lot more of IPsec we could take a look at.

If you need to configure IPsec, your Cisco SE may be able to provide you with the IPsec Design Guide. The version I have is 444 pages, mostly examples. There is an online version accessible via CCO partner login:

[http://www.cisco.com/warp/partner/synchronicd/cc/so/neso/sqso/eqso/iptoc\\_dg.htm](http://www.cisco.com/warp/partner/synchronicd/cc/so/neso/sqso/eqso/iptoc_dg.htm)

By the way, there is a Cisco VPN (IPsec) training course available to Training Partners: CSVPN. Mentor Technologies does not currently (Feb. 2001) offer it (but check our web site for more recent information). It is documented at 4 days in length and apparently covers IPsec for routers, PIX, and Cisco 3000 (former Altiga) devices. If you're interested in taking this course (or have taken it) please drop me an email note.

Your comments, preferences and ideas and suggestions for topics are always more than welcome! I enjoy hearing from you!

---

Dr. Peter J. Welcher (CCIE #1773, CCSI #94014) is a Senior Consultant with Chesapeake NetCraftsmen. NetCraftsmen is a high-end consulting firm and Cisco Premier Partner dedicated to quality consulting and knowledge transfer. NetCraftsmen has nine CCIE's, with expertise including large network high-availability routing/switching and design, VoIP, QoS, MPLS, network management, security, IP multicast, and other areas. See <http://www.netcraftsmen.net> for more information about NetCraftsmen. Pete's links start at <http://www.netcraftsmen.net/welcher> . New articles will be posted under the Articles link. Questions, suggestions for articles, etc. can be sent to [pjw@netcraftsmen.net](mailto:pjw@netcraftsmen.net) .

---

2/5/2001

Copyright (C) 2001, Peter J. Welcher