

An Introduction To Security In IPsec

by Mike Borza and Al Hawtin

Early in the existence of the commercial Internet, the Internet Engineering Task Force (IETF) recognized that there was a need for a robust end-to-end security design for IP networks. This resulted in a number of RFCs (requests for comments) being authored to create an overlay security design in the network layer. IPsec remains the dominant virtual private networking technology and is widely deployed in handsets, gateways and carrier grade VPN service routers. Many of the interoperability issues encountered early in the IPsec lifecycle have been resolved and software solutions from Microsoft, in Linux distributions and in proprietary network products, support well-proven IPsec solutions. As more IPsec traffic traverses the Internet and wireless networks, designers are increasingly faced with the requirement to build hardware offload solutions to help process the security load presented by this class of traffic. This TechNote is the first of two that looks at the challenges of processing IPsec. Here we provide background on security definitions and the security model used in IPsec. The next TechNote will examine hardware offload solutions and how they are integrated into software stacks.

IPsec Matures And Delivers

There is no denying that any security design is difficult to administer and presents overhead in processor load and network traffic. In the past these immediate costs outweighed the long term benefit of securing sensitive corporate data against possible future compromise. However, many of the data security breaches in recent years demonstrate that it is essential to adopt a proactive security-in-depth posture. Single points of failure and weak links come under attack in proportion to the value of data they protect. For organizations accumulating millions of records of private personal data pertaining to health, financial, government services entitlements and banking public, handling these data in an irresponsible way is a serious liability.

These two cases illustrate the liability of sending unencrypted back-up tapes through couriers: [Bank of America today](#) is warning the holders of at least 1.2 million of its federal employee credit card accounts that a major security breach...

[The retail finance division of Citigroup](#) has admitted that a backup tape containing personal information on almost 4 million customers in the US has gone missing.

Clearly, there are two better models which could be employed. The first is to encrypt the back-up tapes using the new IEEE 1619 standard for tape. Alternatively, a continuous incremental back-up could be done between the servers and back-up location over the Internet using a VPN technology such as IPsec.

The most egregious breach in recent years was the compromise of 94 million customer records at [TJX](#). It is now clear that the initial compromise of the TJX credit card processing system was through a wireless network at a retail branch. This access point may have been set up with security turned off; perhaps a WPA key was comprised; or perhaps they were using the vulnerable WEP link security design. A robust, network wide security design such as IPsec could have prevented this breach if properly deployed and administered.

Background

Each security design is unique and must consider the threat model and economics which guide successful and resilient security architecture. The key elements in a security design are:

Confidentiality	Only intended recipients of data can correctly interpret it
Authenticity	The originator of the data is known to the recipient
Integrity	The data has not been changed without detection (either intentionally or accidentally)
Non-repudiation	The originator of the data cannot deny that it sent that data

In IPsec, the confidentiality and integrity are implemented through the Encapsulating Security Payload (ESP) and the Authentication Header (AH). IPsec is a flexible and extensible architecture that provides a standard set of cipher suites (eg AES for confidentiality and HMAC/SHA-1 for message authentication), and allows for local extensions to suit local requirements. Authenticity in IPsec is managed through a key exchange protocol such as IKE (Internet Key Exchange) combined with a Public Key Infrastructure (PKI) service. Non-repudiation is weak in IPsec as it was felt to be less important in terms of the network bias of IPsec and, should non-repudiation be required, it could be accommodated at the application layer.

Every encryption protocol such as IPsec relies on algorithms developed to ensure the confidentiality and integrity of the data protected by the protocol. In these two TechNotes, the following restricted definitions will be used:

- Cipher: an invertible algorithm to create a confidential representation of data
- Plaintext: also called cleartext, the representation of data in a format that is user meaningful
- Ciphertext: the encoded form of data such that its meaning is hidden (confidential)
- Encryption: the process of encoding plaintext using a cipher create a ciphertext
- Decryption: the process of decoding ciphertext using a cipher to re-create the plaintext

Comparing IPsec to other network security solutions, explains why it dominates this space. First, it is important to understand how IPsec has been integrated into the network stack and Fig. 1 illustrates how security at different layers integrates into both TCP/IP and the ISO model.

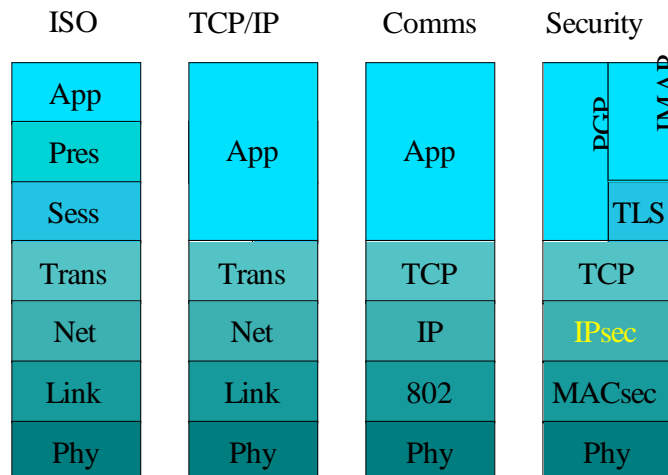


Fig. 1: Network Security Model

IPsec is integrated into the IP layer (network layer in ISO terms) of the TCP/IP network stack. Layer 2 security is found at the datalink (MAC) layer, while transport and application layer security are found above the network stack in TCP/IP networks, and generally span the Transport, Session, Presentation and Application layers of the ISO model.

Layer 2 (datalink) VPN technologies such as MACsec (IEEE 802.1AE) and L2TP provide protection to individual LAN or WAN subnets, allowing only authorized devices to access the network and its resources. When confidentiality is enabled on subnets, network data (including network layer headers that may disclose network architecture information to an attacker) is protected from eavesdropping by unauthorized entities. (Note, however, that this protection is lost at the edges of the subnet.) Layer 2 VPNs are generally restricted to a single subnet: as such, they are not routable. Layer 2 VPNs are able to carry non-IP network protocols such as NetBEUI.

This view of Layer 2 VPNs is more restrictive than some common industry uses of the term as it has a rather broad set of definitions in the industry. From a purist perspective a Layer 2 VPN should be considered a networking solution which implements privacy protection and, possibly, authenticity through the use of encryption technologies. However, many network equipment suppliers and leased network providers classify label switching and virtual LAN (VLAN) technologies as Layer 2 VPNs, arguing that the obfuscation of traffic traversing the network is sufficient for many applications. The single largest advantage of L2VPN technology is the raw bandwidth available to implementers. It is now feasible to offer solutions which scale to 10 Gbit/s and beyond with reasonable size offload engines.

By contrast, IPsec VPNs operate at Layer 3 (network), which means that IPsec data is routable on the global Internet and other IP-based networks. This is enormously powerful, since IPsec data is simply a different kind of IP data in the Internet. Routers and other network equipment in the access and core networks do not need to do anything special to handle IPsec data, and require no configuration in advance of creating the VPN. Only the end parties to communication need to perform any configuration and management to operate an IPsec VPN. Once established, an IPsec VPN protects network data from end-to-end while allowing that data to be routed over the global Internet. IPsec protects all data within the VPN, regardless of higher layer protocol. This makes its use transparent to higher layer applications and protocols: in essence these applications and protocols gain security for free.

SSL and its successor TLS provides secure connectivity between hosts at the transport layer. Strictly speaking, SSL/TLS exists between the TCP (transport) layer and the applications that use it. Like IPsec, it is routable and end-to-end. This is suitable for specific applications that need to assure security, such as web-based transaction processing. Unlike IPsec, applications need to be designed specifically to take advantage of security at this layer. By design SSL/TLS is able to quickly create an authenticated, confidentiality-protected connection between hosts that do not necessarily trust (or even know) each other. So it is more suitable than IPsec for transient connections where the management overhead of creating and provisioning IPsec policies and security parameters would be feasible.

Finally, applications may choose to provide their own security. Perhaps the most common form of this is file encryption within various archive formats. Properly implemented, application security can provide protection for data both at rest on a storage system such as tape or hard disk, and while

in transit on a network. The most popular example is Pretty Good Privacy (PGP), which offers both file encryption and an end-to-end solution for email encryption. Application layer security is often seen as a good solution by implementers as it can be deployed for workgroups requiring the extra layer of security and full end-to-end security protection irrespective of network topology and reach. It is not uncommon for application security such as PGP to run on top of a VPN transparently.

The different attributes of each security option open to implementers are:

Attribute	Layer 2 VPN	IPsec	SSL	Application
Routable	No	Yes	Yes	Yes
Protects network internals	Within the subnet	Yes	No	No
End-to-end	No	Yes	Yes	Yes
Carry other transports (UDP, SRTP, etc.)	Yes	Yes	No	No
Carry non-IP traffic (NetBEUI, etc.)	Yes	No	No	No
Hop-by-hop	Yes	No	No	No

A few years ago there was a great deal of interest in SSL VPN technology as a rival to IPsec. The rationale presented by implementers of this technology was the ease of deployment of SSL VPN solutions, its ease of integration into web applications and robust and interoperable technology used in the design. The reality of SSL VPNs has been somewhat different. The most powerful way to use any VPN technology is to authenticate users or hosts using cryptographic identifiers that allow the use of a protocol to authenticate the parties to communication. In practice, public key cryptography such as RSA or Diffie-Hellman is used in combination with X.509 certificates. Thus, just as in the case of IPsec, VPN, administrators must be able to manage a public key infrastructure (PKI) to control access to the VPN. SSL VPNs have shared this characteristic with IPsec VPNs, largely eliminating much of the “ease-of-use” argument. Because they are built on SSL/TLS, SSL VPNs are an end-to-end security technology and are fully routable. They can leak information on the internal addressing of the LAN, which represents a moderate vulnerability. More importantly, as SSL operates above the TCP layer, only applications that use TCP connections may be carried on an SSL VPN. Thus SSL VPNs cannot protect all of the IP traffic between hosts. After a quick start, however, it appears that SSL VPN technology is being relegated to a niche technology as companies revert to the flexibility and interoperability of IPsec VPNs: universally adopted.

IPsec was initially developed as a software design and as such many cipher suite options were implemented. As network speeds have increased, hardware implementation of the cipher suites, packet transforms, and even the entire packet processing have become commonplace. Widespread availability of hardware and recognition that some cipher suites are more suitable than others for high-speed implementation has led implementers to narrow the range of cipher suites in general use. As the cipher suite is specified as part of the IPsec policy, and as implementations migrate from software to hardware-based VPNs, VPN deployments may be migrated from older suites to more modern ones.

IPsec Standards

Over the last four years, there has been an effort to overhaul IPsec, both in IPv4 and in IPv6. The result is an updated series of RFCs for IPsec that reflect more than a decade of experience. The new base architecture document for IPsec is now RFC4301, which replaces the original standard specified in RFC 2301. Some of the more important new RFCs that define IPsec are:

RFC 2367	PF_KEY Interface
RFC 2410	The NULL Encryption Algorithm and Its Use With IPsec
RFC 2411	IP Security Document Roadmap
RFC 2412	The OAKLEY Key Determination Protocol
RFC 2451	The ESP CBC-Mode Cipher Algorithms
RFC 2857	The Use of HMAC-RIPMD-160-96 within ESP and AH
RFC 3526	More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
RFC 3706	A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
RFC 3715	IPsec-Network Address Translation (NAT) Compatibility Requirements
RFC 3947	Negotiation of NAT-Traversal in the IKE
RFC 3948	UDP Encapsulation of IPsec ESP Packets
RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
RFC 4301	Security Architecture for the Internet Protocol
RFC 4302	IP Authentication Header
RFC 4303	IP Encapsulating Security Payload (ESP)
RFC 4304	Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)
RFC 4305	Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4306	Internet Key Exchange (IKEv2) Protocol
RFC 4307	Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
RFC 4308	Cryptographic Suites for IPsec
RFC 4309	Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)
RFC 4478	Repeated Authentication in Internet Key Exchange (IKEv2) Protocol
RFC 4543	The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
RFC 4555	IKEv2 Mobility and Multihoming Protocol (MOBIKE)
RFC 4621	Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol
RFC 4718	IKEv2 Clarifications and Implementation Guidelines
RFC 4806	Online Certificate Status Protocol (OCSP) Extensions to IKEv2
RFC 4809	Requirements for an IPsec Certificate Management Profile
RFC 4945	The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX

As a system designer working through the myriad RFCs the first and highly important distinction to make is that certain RFCs relate to key negotiation protocols (in yellow), while the remaining (in green) focus primarily on packet data transforms for IP headers and payload encryption for data transferred over an IPsec network. Of particular note are those pertaining to IKE as these form the framework for automatic secure establishment of IPsec security associations.

IPsec Network Overview

IPsec offers two different network models – Transport Mode and Tunnel Mode.

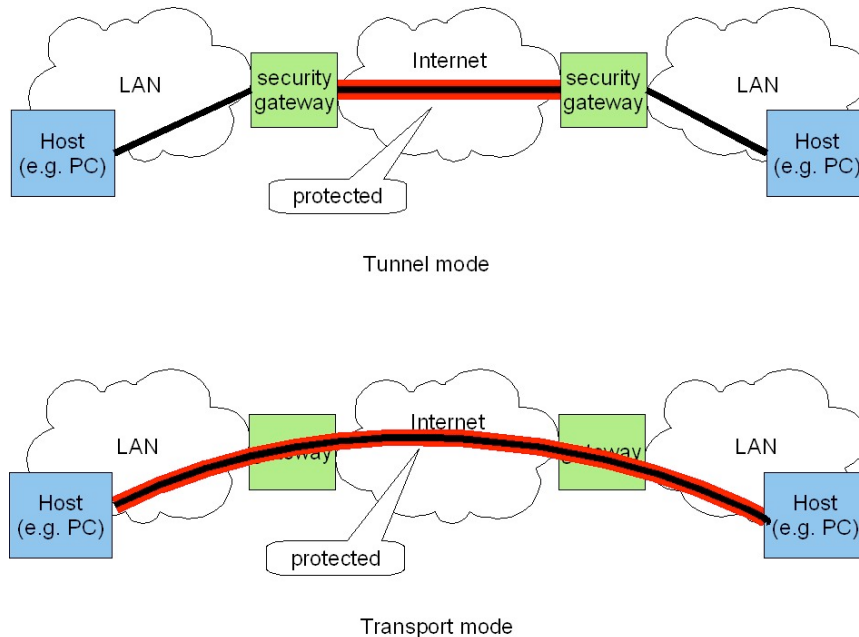


Fig. 2: IPsec Tunnel And Transport Modes

Fig. 2 illustrates these two modes. Tunnel mode offers security from gateway to gateway and effectively hides any details of the LAN networks protected by the security gateway from potential hackers.

This is one of the most common deployments of IPsec VPNs, used to securely connect two LANs in different locations.

On the other hand, transport mode has the characteristic of providing complete end-to-end security including traversal of the LANs at either end; however, both hosts must be reachable by the other over the WAN, which may undesirably leak LAN address information since the IP headers are not encrypted in this case.

Of course, hybrids and variations on these basic configurations are possible: for example, transport mode traffic tunnelled through the Internet via security gateways.

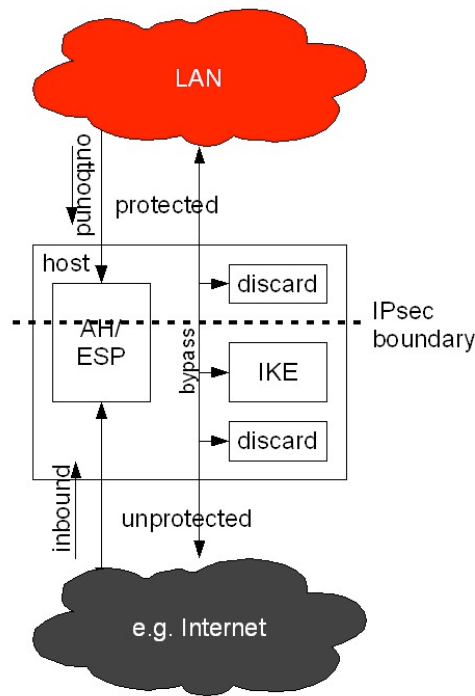


Fig. 3: IPsec Packet Processing

Fig. 3 provides a closer view of the relationship of the various subsystems in an IPsec gateway implementation. Classification of packets from either network interface (and generated internally within the host) takes place in the network stack routing logic. IPsec packets are forwarded to the ESP/AH transform processor for cryptographic processing. According to the local policy defined by the VPN administrator, traffic which does not have IPsec applied may be filtered according to content, protocol or source/destination address. This traffic may be passed to the appropriate interface, dropped if disallowed by the policy, or forwarded to a different application such as IKE for further processing within the host.

Using tunnel mode as an example, traffic originating in the LAN is forwarded to the security gateway, where the following sequence of events will typically occur:

1. Classification: based on the IP source/destination address and protocol, does this packet require an IPsec transform?
 Yes? route to the AH/ESP subsystem in Step 2
 No? route to the bypass subsystem to discard or pass to the IKE stack if key exchange traffic
2. ESP/AH Processing: determine security policy and security association for the IPsec tunnel
 Security policy database: maps IP flows to security parameters for the specific association
 Security association database: maps packets to keys, initialization vectors, packet counters and other state data
3. Encryption transform processing: Encrypt and calculate message authentication code (MAC) for the packet based on the SPD and SAD for that packet/flow. The specific algorithms applied are determined by the policy
4. Packet editing: Implement IP header transform with new IP source/destination addresses encapsulating originating LAN source/destination IP addresses which are now encrypted and authenticated

Fig. 4 illustrates the transform applied to encapsulate a tunnel mode packet. As discussed earlier, the transform encrypts the LAN IP header and adds a new IP header with the source and destination addresses being the local and remote gateway IP addresses. A more detailed view of the IPsec packet construction is shown in Fig. 5 along with the fields which are encrypted and those which are authenticated.

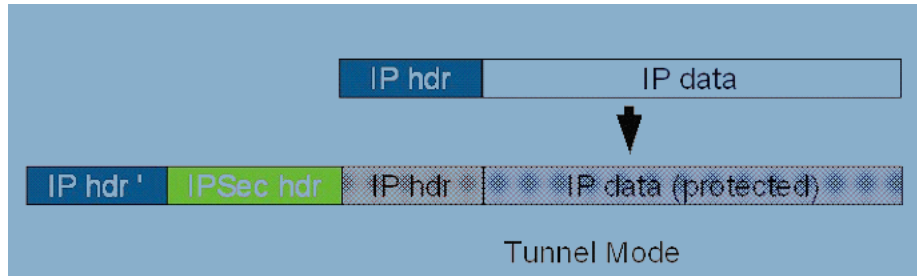


Fig. 4: Tunnel Mode Packet Transform

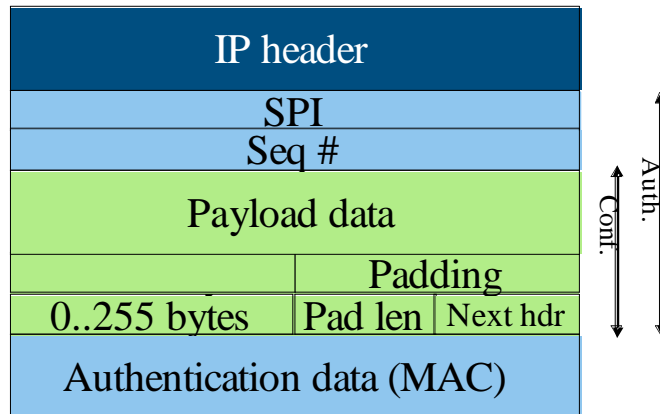


Fig. 5: IPsec Packet After Transformation

As is illustrated in this Figure 5, the IPsec packet now consists of:

1. The new IP (WAN) header
2. The security parameter index which is used to associate the packet to an SPD entry
3. The sequence number which is used for anti-replay protection
4. The original IP packet (including header)
5. Padding to round out the payload to the block size of the cipher
6. The pad length
7. A pointer to the next header to properly reconstruct the original IP packet at the receiver
8. The message authentication code derived from an algorithm such as HMAC-SHA-1

Encryption is applied to the payload, pad and next header pointer while message authentication is done on these fields plus the IPsec header.

IPsec is very flexible, but with this flexibility comes complexity. There are many different configuration options, and modes and transforms may be mixed together. It is possible to combine ESP and AH modes, and this sometimes makes sense. For example, in a transport IPsec implementation, it makes sense to use ESP to protect data confidentiality, then apply AH to provide authentication to both the encrypted data and the IP header, thereby providing data origin assurance. Similarly, it is also possible to put transport mode IPsec data into tunnels to transmit that data between sites. It is even possible to wrap tunnels within tunnels, allowing construction of more complex network architectures. Fortunately for developers, popular IPsec stacks support these modes as they can become very complex.

Implementation Of IPsec

There are three common IPsec architectures. Each has its own attributes and drawbacks.

Bump-in-the-Stack (BITS): implement a separate layer in the network protocol stack: requires duplication of network layer features such as fragmentation and routing tables or a re-entrant stack

Bump-in-the-Wire (BITW): an external, dedicated IPsec device that provides the externally visible IP connectivity

Stack Integrated: a distributed implementation in the network layer, eg Linux kernel IPsec

The majority of implementations today fall into the category of either the Bump-in-the-Stack implementation or the Stack Integrated approach. The 2.6 series of Linux kernels provides a complete kernel implementation of IPsec and is a very popular solution for server based and embedded developers. It is also more challenging design to implement hardware offload in and as such is used as the reference architecture for the second article in this series.

Bump-in-the-Wire solutions remain very commonplace in VPN service routers implemented in carrier network solutions. They offer the advantage of implementing IPsec in a dedicated, high-performance hardware engine capable of multi-gigabit per second performance. They also create a potential vulnerability in that unencrypted traffic must be routed to the VPN router in plaintext.

IPsec Roadmap

With the recent overhaul of IPsec by the IETF, great improvement to interoperability among major vendors and the inclusion of fast, new ciphers such as GCM-AES in the standards will ensure that this VPN technology remains the de facto choice among network administrators. Elliptic for its part will continue to offer the industry's broadest portfolio of IPsec offload core scaling from solutions that scale from several Mbit/s up to 10 Gbit/s. Working closely with customers, Elliptic will continue to offer software integration tools to ensure that system designers can enjoy the benefit offered by hardware offload solutions from an overall performance perspective.

References

Time *A New Cyber-Security Breach*, by Timothy J Burger, February 25, 2005
The Register *Citibank admits: we've lost the backup tape*, by Andrew Orlowski, June 7, 2005
The Boston Globe *Details emerge on TJX breach*, by Ross Kerber, October 25, 2007
Internet Engineering Task Force (IETF): RFCs are found at <http://www.ietf.org>
Symmetric Cryptography Offload Options for SoC Designers
<http://www.ellipticsemi.com/library-whitepapers-new.php>

About The Authors

Mike Borza is Chief Technology Officer responsible for strategic direction at Elliptic. Mike was a founder and CTO of Startle Networks, a manufacturer of Internet security appliances for SSL VPN and e-commerce applications. At Chrysalis-ITS (now Safenet), Mike worked as Director of IC Applications Engineering of the Luna series of network security processors. In 1995 Mike co-founded American Biometrics, acquired by ActivCard. Mike's early career was in the Toronto area, where he was in safety-critical systems engineering at Alcatel Canada, and held a variety of design and management positions at optoelectronics manufacturer Antel Optronics. In addition to his roles at Elliptic, Mike is a founding director of business networking forum The Ottawa Network.
mborza@ellipticsemi.com



Al Hawtin, VP Marketing and Business Development, is an accomplished marketing and product management professional with more than 30 years of experience in the global high-tech industry spanning both the Semiconductor and Networking markets. He spent four years with Intel in a sales capacity, ten years with Mitel Semiconductor in senior marketing and product management roles, seven years with Newbridge Networks as Vice-President and General Manager and two years as VP Marketing and Sales at ATMOS Corporation: a memory SIP company acquired by MoSys.
ahawtin@ellipticsemi.com

