

Encryption requirements are now found in almost every new SoC design. From digital rights management, through storage security and virtual private network (VPN) applications, security is becoming a mandatory feature. The throughput requirement in modern networks is also rising significantly and as such the processing required for encryption and decryption is substantial. This paper focuses on symmetric offload in a packet processing system for IPsec but the concepts apply equally well to SSL, SRTP and link security. The assumption is that the keys have already been derived through an administrative process or key exchange through asymmetric cryptography in software and therefore the SoC designer is focusing on the bulk encryption and hashing of packets in a virtual private networking (VPN) enabled gateway design.

A typical SoC architecture for such a gateway is shown in Figure 1.

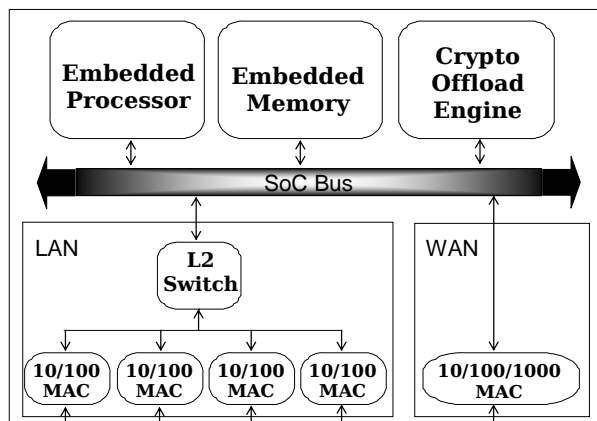


Figure 1 Gateway SoC Architecture

This approach to cryptographic offload is sometimes referred to as look-aside security offload in contrast to a flow through engine which captures VPN traffic directly from the MAC and processes it in line without significant processor interaction. The crypto

engines presented in this paper are optimal in gate count and throughput for applications in gateways, security appliances and handheld devices. The engines scale well in throughput from 1 Mbps up to 1 Gbps. Flow-through engines are best suited for ultra-high performance applications at 1 to 10 Gbps rates in high end security appliances.

To better understand the operation of the SoC system design, it worth reviewing the routing and transformation that a packet is subjected to. A packet arriving at the gateway through an IPsec VPN tunnel active on the WAN port undergoes the following transformations:

1. The inbound WAN MAC DMA's the packet into embedded memory
2. NAT software determines that the packet requires an IPsec transform
3. Software matches the packet to a security policy and security association
4. The packet transformation and decryption is done in software and/or hardware.
5. Software routes the packet to the L2 Switch
6. The L2 switch forwards packet to the destination LAN port

The designer therefore must decide how to implement the packet transformation and cryptographic operations. Will the embedded CPU and associated software be able to handle the load? Or will some form of offload engine be required to reach the overall system performance goal? This paper explores the offload options available to designers and provides guidance on the class of engine employed for each level of throughput required.

The terms cryptographic, crypto, cipher are used interchangeably in the following text. Generally these terms may be applied to hashing operations as well.

Processor Based Security

The first option is to perform all security processing on the existing embedded processor. Elliptic has done extensive analysis of the cryptographic load on common embedded processors such as ARM and MIPS processors and has derived a load factor of 30 MIPS per Mbps for the three pass version of the Data Encryption Standard common referred to as triple-DES or 3DES and 10 MIPS per Mbps for the Advanced Encryption Standard (AES). 3DES remains relevant to current cryptographic design as it is a popular cipher option for IPsec used in virtual private networking. AES is the preferred cipher recommended by the U.S. National Institute of Science and Technology (NIST) and is slowly replacing 3DES but the migration is gradual with many legacy devices still active in networks. If an IPsec virtual private network design were to target 10 Mbps for example and the traffic exhibited a mix of 50% AES and 50% 3DES, the overall load on the processor would be 20 MIPS – a reasonable load for today’s high speed embedded processors but a heavy burden for the low cost, low performance processors found in many handset devices. Consider what happens if the traffic is increased to 100 Mbps. The processor load jumps to 200 MIPS which means that a significant percentage of the capability of the embedded processor is used for symmetric cryptography leaving little capacity for other activities required for the gateway. Options often considered by designers is to use a larger processor, increase the clock rate or even add an additional processor, but these solutions are rarely optimal from a perspective of cost – e.g. incremental license fees, process migration expense, and larger gate counts.

Today’s SoC designs can leverage one of five offload options which are explored in this paper:

1. Implement Processor Instruction Set Enhancements
2. Integrate Slave Cryptographic IP Cores
3. Integrate Cryptographic Cores with Linear Mastering Capability
4. Integrate Cryptographic Cores with Scatter/Gather Mastering Capability
5. Integrate Packet Transformation Engines

Option 1: Enhanced Instruction Set

Some embedded processor architectures such as the Tensilica Xtensa and ARC processor cores allow the designer to add custom instructions. Cryptographic functions typically have many fine grained bit manipulation operations and as a result these operations take a large number of instructions to perform in software. Adding enhanced instructions can help by decreasing the total number of instructions required to perform a cryptographic operation. Some examples are:

- Barrel shift/rotate instructions.
- Wide register manipulation. e.g. AES works on 128 bit words.
- Specific bit manipulation instructions: e.g. an instruction to perform the AES/DES S-box manipulation on a wide register.
- Galois Field arithmetic

This approach has a significant drawback in that the enhanced instruction set must be supported in the tool chain. If this issue can be overcome, the designer can expect an improvement of up to 50% in throughput which may meet the system performance requirement. If not, the addition of a separate cryptographic engine as an IP core will be the right path to follow.

Option 2: Pure Slave Cryptographic Cores

The next level of offload is the addition of a pure slave crypto module. This brings cryptographic acceleration with a minimal gate footprint as the crypto module consists of only the circuitry required to implement the datapath algorithm. Software must sequence each block of data through the cipher engine which is simply memory mapped as a set of control and data registers. This greatly accelerates the cryptographic algorithm, but will require substantial host processor involvement to feed the cipher datapath. Figure 2 illustrate this approach.

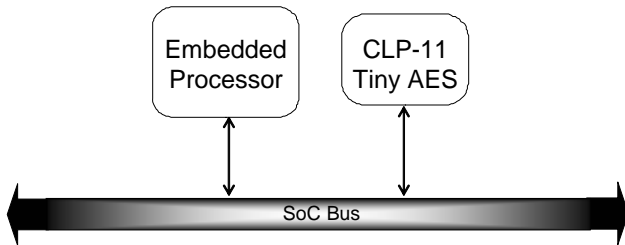


Figure 2 Slave cryptographic IP cores

This offload design greatly accelerates the cryptographic operation; but there is still substantial processor involvement to feed the cipher core. Designers therefore generally apply this class of offload engine to IPsec requirements up to 10 Mbps in applications such as residential gateway and VDSL modems.

Option 3: Bulk Cryptographic Engine with Linear Master

The next level in performance is to provide a dedicated core data buffer into which software can load a larger amount of data into prior to starting the crypto engine. This can be sized all the way up to a complete Ethernet packet or even two packets to permit one packet to be in

transit into or out of the engine while another packet is undergoing an encryption, decryption or message authentication operation. The crypto engine reads data via a master memory interface for example then writes the ciphered data back to the buffer. This architecture is shown in Figure 3.

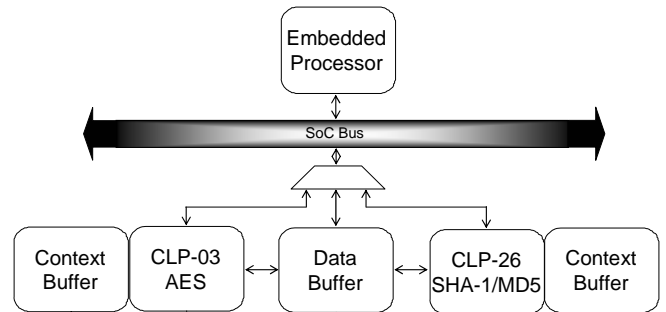


Figure 3 Bulk Cryptography Engine with Linear Master

On completion of the operation, an interrupt is generated back to the host. This style of engine can be further enhanced by storing multiple contexts in the crypto engine in a separate buffer to allow for ciphering several streams of data in an interleaved fashion. This approach will permit SoC designers to achieve performance capability up to 60 Mbps making it perfect for gateways, VDSL modems and security appliances.

Option 4: Bulk Cryptographic Engines with Scatter/Gather DMA

Packets and fragments of packets are generally stored in system memory and are scattered across several memory buffers. To further offload the host from feeding data to the cryptographic engines, a scatter/gather DMA engine may be used to collect data from multiple locations in system memory and write the data back upon completion of the cryptographic operation. To further enhance the

capacity of the engine, a sequencing module eliminates the host processor from requiring direct control of the cipher and DMA operations. The host simply writes a pointer to a descriptor table and a command register, which causes the engine to import all required data, cipher it, and write it back out to the host memory. This approach is illustrated in Figure 4.

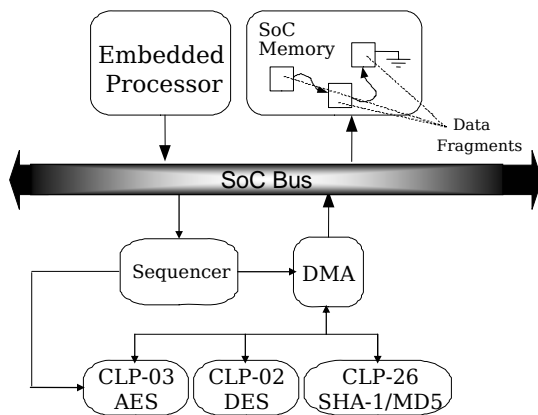


Figure 4 Bulk Cryptographic Engine with Scatter/Gather DMA

In this scheme, designers usually implement a suite of cryptographic modules behind the DMA and sequencer modules. The sequencing module may allow for chained cipher and hash operations, requiring the data to traverse the bus a single time for both cipher and hash operations.

Using this technique, designers can reach up to 100 Mbps of IPsec traffic with a slightly larger engine making this solution suitable to gateways, appliances and base station applications.

Option 5: Packet Transformation Engines

For high performance gateway and security appliance applications, Elliptic has developed protocol-aware cryptographic offload in for the CLP-25 and CLP-30. This permits crypto acceleration to reach up to the Gbps range in a reasonable engine size.

Security protocols such as IPsec involve the use of a cipher-suite, which involves both an encryption operation for confidentiality and a hash operation for authentication. Additionally, security protocols require insertion of block cipher padding, security headers and trailers, and provide mechanisms to prevent replay attacks. This class of security offload is a packet transformation engine. An IPsec packet processor such as the CLP-25 will apply the full ESP and AH transforms to an IP packet. The block diagram for this engine is shown in Figure 5.

This protocol requires that the cryptographic state must be maintained for each session. This is done in the form of a Security Association Database (SAD). The information in the SAD is initialized by the host processor via the key exchange mechanisms. Once data begins flowing on the connection, the entries in the SAD are managed by the packet transformation engine. Examples of information managed by the hardware and stored in the SAD are; cryptographic parameters (algorithms, keys, IVs), anti-replay lists, connection lifetime counters etc

In addition to the full transformation logic, the engine may provide methods to access the raw cryptographic resources directly. This allows

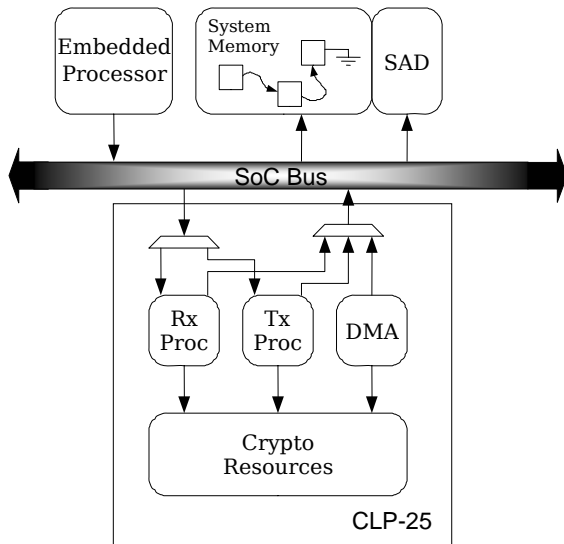


Figure 5 Packet Transformation Engine

protocols that are not implemented by the hardware to take advantage of the cryptographic acceleration in either the "Bulk Cryptographic Engine with Linear Master" or the "Bulk Cryptographic Engines with Scatter/Gather DMA" forms. For example an IPsec packet transformation engine which only implements the transforms for IPv4, could still be used to accelerate IPsec with IPv6 traffic by doing the packet transforms in software and using the bulk cryptographic acceleration for encryption and hashing.

Summary

Elliptic has developed several classes of cryptographic offload engines. The table below illustrates the engine capabilities and size in ASIC gate count.

Class of Engine	Gate Count	Performance
Slave IP Core	10,000	1-10 Mbps
Packet Transformation	190,000	40-200 Mbps
Packet Transformation	300,000	200-500 Mbps

Elliptic has developed a complete portfolio of cryptographic offload engines spanning markets such as DRM, VPN, Storage and MACsec. By implementing configurable engines from the ground up, Elliptic is capable of adapting its designs to precisely meet the performance requirements of its customers while preserving the economics in gate count required by the end market cost goals of the SoC designer.

For more information on Elliptic products and services, please contact:

Elliptic Technologies Inc.
62 Steacie Dr., Suite 201
Kanata, ON
Canada, K2K 2A9
info@elliptictech.com
Phone: +1 613 254-5456