Setting up secure VPN connections with cryptography offloaded to your Altera SoC FPGA

Roger May, System Architect, Altera
Sébastien Rabou, Product Manager, Barco Silex
Gregory Baudet, Marketing Manager, Barco Silex

IT applications that monitor and run industrial infrastructure are more and more connected to each other and to the cloud. Examples are the power grid, oil and gas infrastructure, supply chain and logistics… The Industry 4.0 is becoming embedded in a growing Internet of Things (IoT). If you are responsible for implementing and safeguarding the security of industrial applications, this forms a formidable challenge. The question is not whether cyberattacks on your infrastructure will happen; it is when they will happen.

A key element is securing all point-to-point connections in the network through the proper use of cryptography. But if you add these compute-intensive routines to your software stack, they may put a heavy burden on the performance of your applications, and still leave them vulnerable.

In this white paper, we'll explain the benefits of offloading cryptography routines to hardware. As an example platform, we consider the Cyclone® V SoC device, an Altera® FPGA. Key here is selecting the right IP blocks and installing the appropriate Linux drivers that drive the hardware and allow for an easy integration in your application. Next to being more secure, hardware cryptography is also much faster. A comparison of hardware and software security routines on the Cyclone V SoC shows a gain of 30X for typical Ethernet packets of 1.5 Kbytes.

Reference design and platform for secure communication

Barco Silex has developed a reference design to easily deploy a secure communication channel on an Altera SoC device. It combines a hardware part implemented within the FPGA, and a software part running on the integrated ARM processor of the SoC. The reference design can be used on any SoC device, including the Cyclone V SoC, Arria® V SoC, Arria 10 SoC, and even the high performance Stratix® 10 SoC.

The FPGA programmable part is used to implement specific IP cores dedicated to the execution of cryptographic algorithms, such as AES, RSA, ECC. These IP cores are very flexible, and can be tuned to address the required performance of the application and

keeping the logic size to the minimum. The FPGA also gives a great flexibility for future protocol and cipher updates.

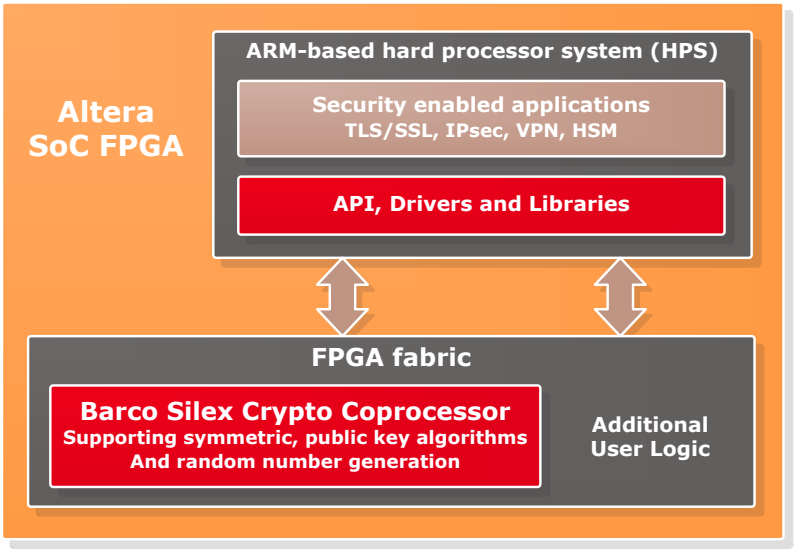

Figure 1 : Altera SoC FPGA with Barco Silex Cryptographic Coprocessor in FPGA and software drivers running on ARM processor

The idea behind this reference design is to enable a straightforward integration in a standard Linux environment. The Linux Kernel already implements cryptographic functionalities, and it is just a logical choice to interface the reference design to the Crypto API. Therefore, the reference design includes Linux drivers that make the bridge between the Crypto API of Linux and the acceleration cores in the FPGA fabric.

The interface between the hardware part and the software part also requires some care in order to achieve good performance with symmetric ciphers, such as AES. For this reason, we have integrated our dedicated DMA in the FPGA. The complete hardware module, also called cryptographic coprocessor, connects to the host processor with standard AXI interfaces.

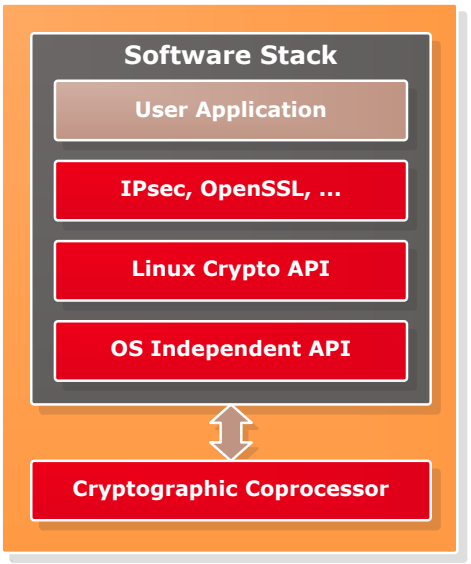

Figure 2 : Linux-based software stack for cryptography operation offloading to the hardware coprocessor

Setup a secured VPN with IPsec

IPsec is a standard which is often used to setup a Virtual Private Network (VPN). IPsec involves multiple cryptographic algorithms in order to ensure the security of the communication. The setup of the connection requires asymmetric cryptography while

the actual data transfer involves symmetric cryptography. This paper essentially focuses on the symmetric cryptography, i.e. the authentication and encryption of the data, but this reference design can also be used to handle the setup of the connection. Processing the CPU-intensive asymmetric operations during the setup of the connection often becomes a challenge as-well when many connections need to be established every second.

Benchmarking IPsec on Cyclone V SoC

To illustrate and compare the performance of cryptography running in software and hardware, we have set up a number of tests using the Altera Cyclone V SoC Development Kit[1], featuring a Cyclone V SoC device. The reference design is easily integrated on the platform. The coprocessor IP core is implemented in the FPGA and interfaced to the AXI interconnect bus. The coprocessor uses about 3K ALM which is only 7% of the FPGA device resources.

In the kernel, the IPsec calls are handled by the crypto API. The software drivers, part of the reference design, are compiled and loaded in the Linux Kernel Crypto API as an alternative implementation to the default software cryptographic algorithms.

The Cyclone V SoC Development Kit from Altera is connected to a remote standard PC. An IPsec connection over a 100 Mbps Ethernet link is established between them. The IPsec VPN is setup using the AES-GCM cipher mode with 256-bit key. With this cipher mode, the data is encrypted with a counter mode and authenticated on the fly using finite field (Galois) multipliers.

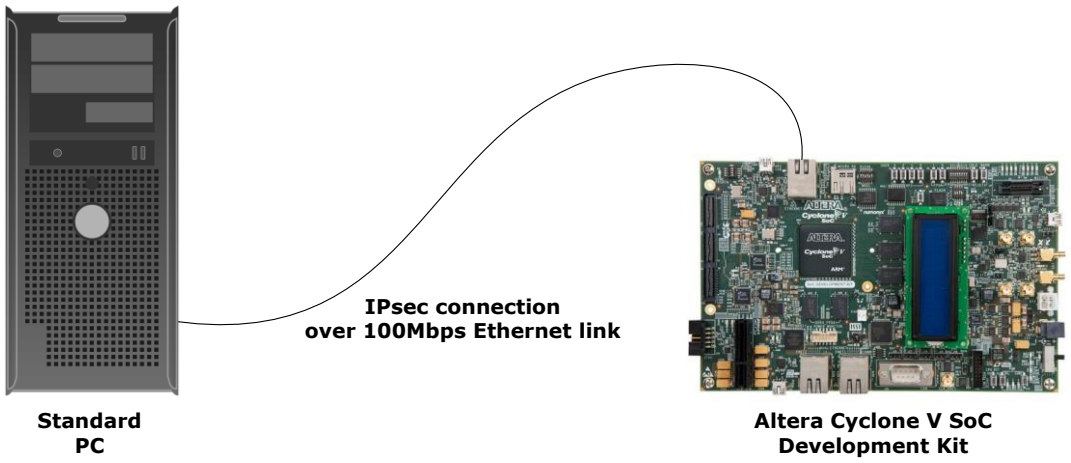

Figure 3 : Benchmark setup with IPsec connection between PC and SoC FPGA

Note that the same reference design could also be implemented on other Altera SoC platforms, notably also the mid-range Arria 10.

[1] https://www.altera.com/products/boards_and_kits/dev-kits/altera/kit-cyclone-v-soc.html

Benchmark results: performance and CPU offloading

In order to benchmark the solution, we used the *iperf* tool, which enables us to measure the maximum bandwidth that can be transmitted over the secured IPsec connection. We also monitored the CPU load on the ARM processor to verify the correct offloading of the CPU.

The first test consists in measuring the maximum bandwidth that we can achieve with pure software implementation, so without our reference design. In this case we see that the ARM processor can support only up to 24 Mbps of total throughput loading one of the CPU cores to 100%. We also verified that a non-encrypted connection was able to reach the maximum throughput of the link (almost 100 Mbps TX+RX), which was the case.

The second test is similar to the first one, but activating the cryptographic offloading from the Linux Kernel to the FPGA crypto coprocessor. The offloading is simply activated by loading the right kernel module. This second test enables us to achieve 91 Mbps TX and RX, for a total of 182 Mbps of total throughput. This is quite a stunning performance considering the low-end SoC FPGA device which is used.

Moreover, during the test at full speed, the CPU load was reduced to only 20%. This remaining CPU usage is due to the Linux Kernel IPsec packet processing. The other 80% CPU load (and also the second CPU core!) can then be freely allocated to the actual application running on the embedded processor.

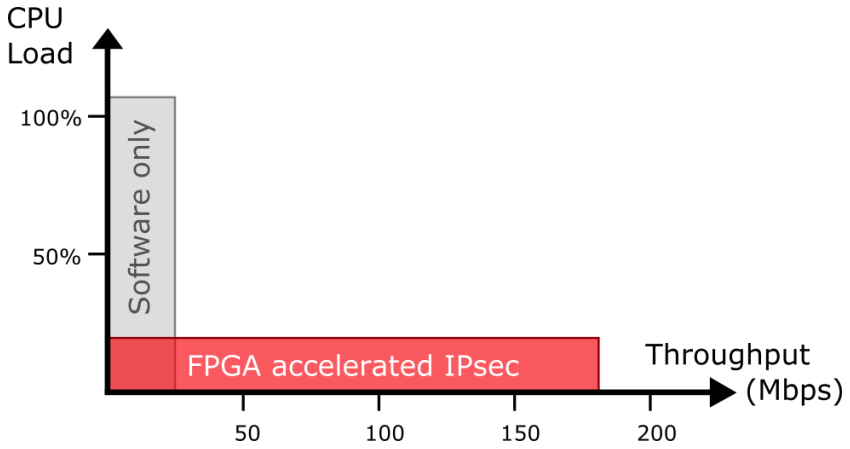

Figure 4 : Throughput and CPU usage comparison between software implementation and FPGA acceleration.

Considering the augmented IPsec bandwidth and the reduction of CPU usage by 4, the reference design can bring an improvement up to 30 times to your secured communication.

Improve security with a True Random Number Generator

The true random number generation (TRNG) is a key concept in cryptography and also important for IPsec. The TRNG is needed to generate keys that must not be predictable or repeatable, even to the most sophisticated attackers. With random or pseudo-random numbers that are generated by software routines, there is a risk that hackers deduce and mimic the algorithms. They may then predict the keys, and thus compromise the application.

Our Coprocessor also features an optional True Random Number Generator to enhance the security of your application. The dedicated true random generator is compliant with the NIST 800-90B and AIS-31 standards, and enables your application to be certified according to usual security standards (FIPS, PCI, …). The random generator also includes Linux drivers to transparently map to /dev/random.

Conclusion

In this white paper, we've talked about the need to secure industrial applications, with VPN being an established option to harden connections. But VPN running in software has some serious drawbacks, especially on the level of performance but also for CPU load. A hardware implementation takes away these drawbacks, resulting in an application that is both safer and faster. Such a cryptography coprocessor can be very easily implemented with high-quality IP blocks, and with drivers added to your OS to interface the hardware.

About Barco Silex

Barco Silex is a recognized leader and provider of hardware security solutions and services. We have developed a suite of IP blocks covering all complex cryptographic routines needed to run secure connections. These include true random number generation and algorithms for authentication and symmetric/asymmetric cryptographic operations. Our IP cores were built to maximally offload all operations, with dedicated techniques to refrain from having to read/write to memory.

Barco Silex is an Altera Design Solutions Network® partner, recognizing the high standards of our services and IP products.

For additional information and contact: www.barco-silex.com

BarcoSilex