Implementation of an FPGA Based Accelerator for Virtual Private Networks

O.Y.H. Cheung, P.H.W. Leong
{yhcheung,phwl}@cse.cuhk.edu.hk
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT Hong Kong

Abstract

Virtual Private Networks (VPN) are becoming increasingly popular network architectures for corporate networks. As VPNs are built on the Internet infrastructure, the data exchange among different local area networks will be passed through the Internet and thus can be easily eavesdropped, masqueraded, etc. Therefore, certain security measures must be used to deal with these privacy issues. The Internet Protocol Security (IPSec) by the Internet Engineering Task Force (IETF) addresses the abovementioned security issues and the Free Secure Wide Area Network (FreeS/WAN) is an open source software implementation of IPSec for Linux which uses triple-DES as the default encryption mode.

As shown in this paper, the performance of FreeS/WAN with IPSec is 50% of that without encryption. In order to improve its performance, a field programmable gate array (FPGA) based triple-DES accelerator was built on a reconfigurable computing development platform called Pilchard and achieved a throughput of more than 120 Mb/sec for triple-DES in cipher-block chaining mode, a speedup of 3 over a software implementation. Measurements show that an FPGA-accelerated FreeS/WAN offers a 30% speedup for the TCP protocol over the original software library.

1 Introduction

Virtual Private Networks (VPNs) are an architecture to realize connections among different private networks over a public network. For example, the Internet can be used as a convenient and low cost channel for a virtual private network. The Internet is a public channel and is not secure. Cryptographic algorithms provide a way to provide a secure communications channel between private networks over the insecure public network.

Field-Programmable Gate Arrays (FPGAs) are hardware devices which are reconfigurable, i.e. programming an FPGA can change its functionally. Implementations of cryptographic hardware using FPGAs offer higher performance than software implementations since higher degrees of parallelism can be achieved. Compared with traditional implementations using application specific integrated circuits (ASICs), FPGAs offer several advantages:

- With FPGAs, it is possible to reconfigure the chip for different encryption standards on demand. This means that unused encryption schemes need not reside on the FPGA, saving resources. In contrast, all supported encryption schemes must reside on an ASIC.
- It is possible to offer field upgrades for FPGA based systems to support bug fixes and new standards.
- FPGAs offer lower costs for small volumes, shorter development times and faster time to market over ASIC technology.
- The performance of FPGA accelerator can be improved by replacing an existing device with a faster one and does not involve any further engineering.

The main aim of this work was to develop an FPGA based accelerator for VPNs. The main aims of this work were as follows:

- Provide a high performance hardware accelerator for triple-DES in Cipher-Block Chaining mode using the Pilchard reconfigurable computing platform.

- Devise a hardware accelerator which is fully compatible with an existing software cryptographic library for use in other applications.
- Explore system issues associated with developing a hardware accelerator for a VPN which is integrated in a real network application.
- Measure the end-to-end performance of an FPGA accelerated VPN and compare it with a purely software implementation.

The rest of this paper is organized as follows. In Section 2, previous work on VPN implementations are reviewed. Section 3 describes the DES and triple-DES algorithms. The implementation of our triple-DES accelerator on the Pilchard FPGA platform is described in Section 4 and results are presented in Section 5. Finally, conclusions are drawn in Section 6.

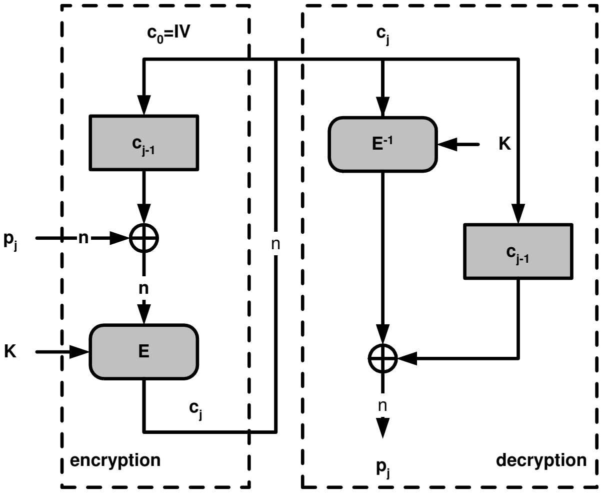

Figure 1: Cipher-block chaining (CBC) mode.

2 Previous Work

2.1 Commercial VPN solutions

There are several existing software and hardware based commercial VPN products. Although different products may have different built-in cryptographic algorithm options, triple-DES is available in all of the VPN solutions described below.

Cisco Systems Inc. have a range of VPN solutions with different specifications. Cisco model 3015 uses software encryption and hence has a relatively low throughput of 4 Mb/sec. In Cisco 5000 series VPN solutions, different numbers of encryption processors can be used. For the highest throughput VPN solution in this series, 760 Mb/sec triple-DES operation is achieved using eight encryption processors.

Intel Corp. provides two VPN solutions using software encryption with throughputs of 8 Mb/sec and 20 Mb/sec using triple-DES. Intel's 3125 VPN gateway uses a PCI encryption processor and has a throughput of 85 Mb/sec for triple-DES.

2.2 Implementations of DES and triple-DES

Software implementations of DES and triple-DES by Biham [1] achieved 46 Mb/second and 22 Mb/sec respectively on a 64-bit 300 MHz Alpha processor. A commonly used open source implementation of DES, LibDES [2] achieves 121.5 Mb/sec for DES ECB mode on an Intel Pentium III 866 MHz machine. LibDES also achieves 42.9 Mb/sec for triple-DES CBC mode on the same machine.

Hardware implementations offer much higher performance than software. In 1999, Free-DES [3], a 3656 Mb/sec implementation of DES algorithm on Xilinx Virtex XCV400-6 with 60 MHz clock rate was reported. A 1280 Mb/sec implementation of DES was reported in 1999 [4] by Wilcox et. al. Sandia National Laboratories developed an ASIC implementation of DES [4] which achieved 6700 Mb/sec. A Xilinx Virtex based DES implementation [5] was proposed by Patterson which achieves 10752 Mb/sec. This implementation operates at a 168 MHz clock rate and employs dynamic circuit specialization in an FPGA to achieve high performance. Trimberger et. al [6] employed even deeper pipelining to achieve a 12 Gb/sec performance on a Xilinx Virtex device.

It should be noted that previous high performance hardware implementations of DES maximize their throughput by unrolling and pipelining the design in ECB mode. However, for improved security, feedback modes such as cipher-block chaining (CBC) are employed. In the CBC mode of operation, every plaintext block is exclusive-ORed with the previous ciphertext block before being encrypted (see Figure 1). Since the input is dependent on the previous output, pipelining does not offer the same advantage as for ECB mode and so the performance of the CBC mode of operation is much lower than that of ECB mode.

3 The Data Encryption Standard Algorithm (DES)

The Data Encryption Standard (DES) [7, 8] algorithm is the most widely used secret key encryption algorithm. It was the first commercial cryptographic algorithm with fully specified implementation details. Although introduced in 1976, it has proven resistant to all forms of cryptanalysis. A major disadvantage of DES is that its 56-bit key is not large enough by today's standards. A DES key search engine called "Deep Crack" which can search 88 billion keys per second was able to solve the RSA laboratories DES-III challenge [9] (which involves finding the key of a DES encrypted message) in 22 hours.

DES is a block cipher as shown in Figure 2 which processes 64-bit plaintext blocks and produces 64-bit ciphertext blocks. The effective portion of the secret key is 56-bit out of 64-bit since although the key is 64-bit, 8-bits are used as parity bits.

DES encryption proceeds in 16 identical rounds. From the input key, sixteen 48-bit subkeys K_i (one for each round), called the key-schedule, are generated via a series of left shifts and permutations. Within each round, 8 fixed 6 to 4-bit substitution mappings known as S-Boxes are used.

The plaintext has an initial bit permutation (IP) and is then divided into left L_0 and right halves R_0, each 32-bits in size. Each round takes 32-bit inputs L_{i-1} and R_{i-1} from previous rounds and produces 32-bit outputs L_i and R_i for $1 \leq i \leq 16$, as follows:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \qquad (1) \end{aligned}$$

where $f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$, E is a fixed expansion permutation mapping R_{i-1} from 32-bits to 48-bits and P is a fixed permutation on 32-bits. The right half of each round goes through an expansion permutation from 32-bits to 48-bits and is then exclusive-ored with the subkey of that round. The temporary result is passed through an S-Box and forms the new 32-bit product of the right half. For each round, right half and left half are exchanged. Finally both halves are combined together in the 16th round and permuted by the inverse of the initial bit permutation to form the ciphertext.

Decryption uses the same key and algorithm, however, the subkeys in internal rounds are applied in reverse order. For encryption, the key schedule order is $K_1, K_2, K_3, \ldots, K_{16}$. For decryption, the decryption key schedule is $K_{16}, K_{15}, K_{14}, \ldots, K_1$.

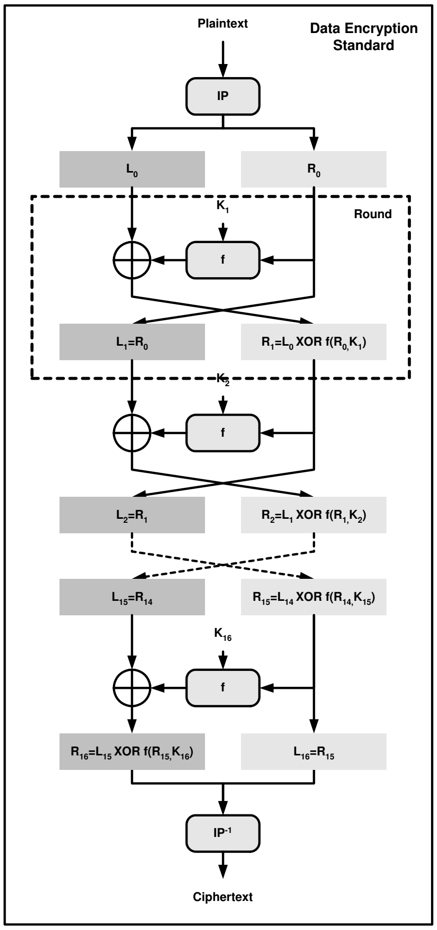

Figure 2: Data Encryption Standard algorithm

3.1 The Triple-DES Algorithm (3DES)

The triple-DES algorithm [10] was introduced to increase the key size of DES while maintaining compatibility with legacy DES software and hardware systems. For encryption, the plaintext is passed through three cascaded DES cores as shown in Figure 3. Note that the first and the last DES cores are in encryption mode and the second one is in decryption mode. If the same key is used for K_1 *and* K_2, triple-DES is the same as DES with key K_3. For decryption, the modes are inverted so that the first and the last DES cores are in decryption mode and the middle one is in encryption mode. The triple-DES algorithm increases the key size by a factor of three compared to DES, i.e. from 56-bits to 168-bits. However, the processing time is increased by the same factor.

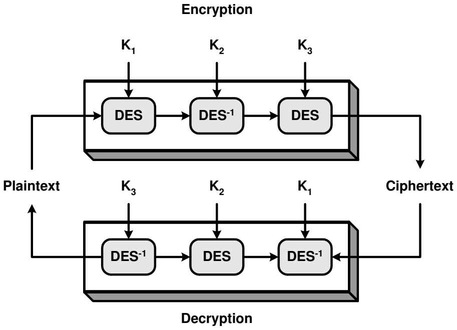

Figure 3: Triple-DES algorithm

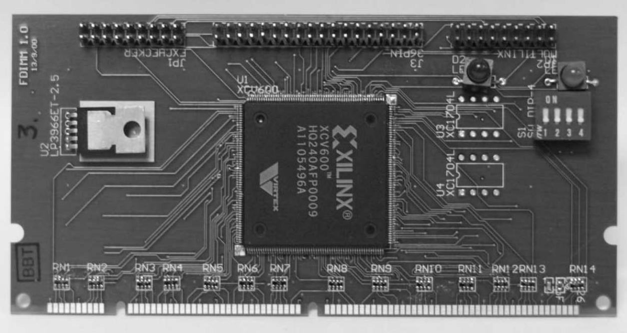

Figure 4: Picture of Pilchard

4 Implementation

4.0.1 Pilchard Platform

The FPGA platform used was a Pilchard FPGA card (Figure 4) [11] populated with a Xilinx Virtex XCV1000E-6 FPGA. Pilchard uses a 64-bit SDRAM memory bus interface instead of the conventional PCI bus and has much improved latency and bandwidth over the standard PCI bus.

4.1 Triple-DES in CBC mode

4.1.1 Hardware

The triple-DES core (Figure 5) was formed by cascading three combinational DES cores. Although the triple-DES core is combinational, an external finite state machine was used to determine the readiness of input and output. The triple-DES core operates at 2.135 MHz and the external finite state machine works at 50 MHz which is the system clock (100 MHz) divided by two.

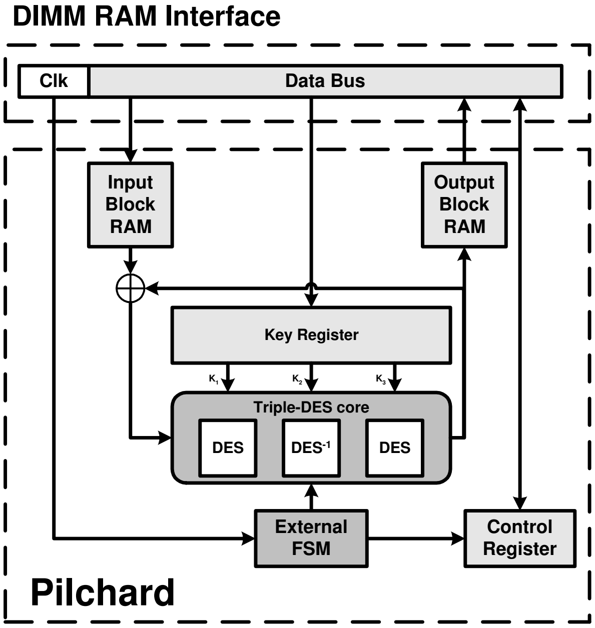

Figure 5: System architecture of triple-DES accelerator

4.1.2 Software Interface

A polled interface was used for data transfer between the host computer and the Pilchard board. For efficiency reasons, transfers are made to the Pilchard board in blocks of 248 64-bit words. In order to perform an encryption or decryption, the data are first transferred to the board, the core is then asked to encrypt/decrypt the data. The host polls the core for a completion signal after which it can transfer the block of processed data back to the host. Note that the completion signal is produced before all the data have been processed. The completion signal is timed so that if the host starts reading at that time, computation of the unprocessed data will finish before the host reads it. Note that this scheme is only able to partially overlap communications and computation. However, as will be shown in Section 5, the main system bottleneck is in the triple-DES core.

In order to integrate the hardware accelerated Pilchard triple-DES core into the FreeS/WAN VPN software, the LibDES library was first modified to call the hardware accelerated Pilchard triple-DES core in-

Table 1: Configuration of benchmark machine.

CPU	P-III 866
RAM	128 MB
Motherboard	Asus CUSL2 (Intel 815EP chipset)
Network card	3COM 590 (100 Mb/sec)
OS	Mandrake v7.2 with kernel 2.2.16

stead of the software version. Since FreeS/WAN manipulates IPSec packets directly in the Linux kernel, both user mode and kernel mode versions of LibDES were required. The user and kernel mode functions differ in the representation of memory address mapping for Pilchard. In user mode, a virtual address is used in interface for Pilchard, however, direct access to a physical address is used in kernel mode.

The architecture of the VPN and IPSec protocols in FreeS/WAN was unchanged, therefore no major modifications to FreeS/WAN were required. One minor issue was that in FreeS/WAN, the triple-DES keys are first processed and stored in the form of a key-schedule. In contrast, in the triple-DES core, the key should be a raw-key.

The host interface of the triple-DES accelerator on Pilchard could be modified to accept a key schedule instead of the key. However, this approach would require 12 times more data to be transferred and is thus inefficient. The solution adopted was to modify LibDES to accept a raw key instead of a key-schedule.

5 Results

In this section, results are presented. Firstly, the testing environment is introduced. This is followed by performance results for the triple-DES accelerator. Finally, benchmark results obtained using the FPGA-accelerated version of FreeS/WAN are presented.

5.1 Benchmarking environment

Two computers with identical configuration were used for benchmarking and obtaining all the results. These two computers connect to a 100 Mbit network via a hub running FreeS/WAN version 1.5 with Linux Kernel 2.2.16 as shown in Table 1.

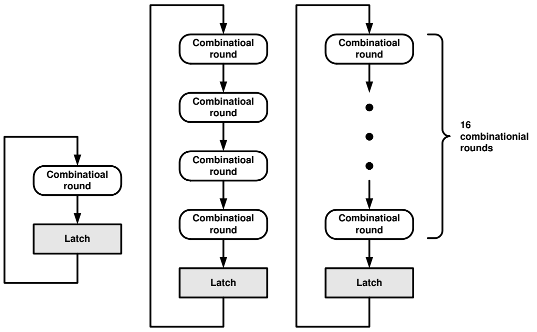

Figure 6: Architecture of the DES core with 1, 4 and 16 combinational rounds.

5.2 Performance of Triple-DES Core

The triple-DES processor on Pilchard was synthesized using Synopsys FPGA Express 3.5 and Xilinx Foundation Series 3.3i. The design was verified using the Synopsys VHDL Simulator and successfully implemented on Pilchard board. All implementations were tested using Pilchard cards populated with a Xilinx XCV1000E-6 device.

A study of area and speed tradeoffs for a single DES core with different degrees of unrolling (i.e. different numbers of rounds) was conducted (see Figure 6). Table 2 shows the performance of DES cores with different numbers of combinatorial rounds in ECB mode. As can be seen, the throughput is rather low if only 1 round is used due to overheads associated with registering the intermediate values. If 2 to 16 combinatorial rounds are used, the throughput is roughly constant at around 400 Mb/sec. A DES core design with 16 combinatorial rounds was selected for the triple-DES implementation since there were sufficient logic resources on an XCV1000 device, and a totally combinatorial DES core is slightly easier to incorporate in the triple-DES design.

From Table 2, the performance is similar among the DES cores. Therefore, a core with 16 rounds was chosen since it has a simpler control and host interface.

The triple-DES CBC core uses three combinational DES cores with 16 combinational rounds. It requires 5368 Virtex slices, which is 43.68% of the total 12288 slices in a Xilinx Virtex-E XCV100E device, and operates at 2.135 MHz, and thus has a maximum throughput of 2.135 MHz× 64-bit = 136.64 Mb/sec.

The triple-DES accelerator was tested on the ma-

Table 2: Area and Speed tradeoffs among DES cores with different numbers of rounds.

Number of combinational rounds	Area (slices)	Clock rate	Throughput (Mb/sec)
1	747	58.42	233.68
2	765	51.3	410.4
4	877	23.38	374.08
8	1121	12.32	394.24
16	1666	5.94	380.16

chine described in Table 1. The Linux kernel function do_gettimeofday() was used for timing. As shown in Figure 7, the performance of the triple-DES accelerator for small amounts of data is much lower than software. As the data size increases, the performance increases quickly and achieves a higher performance than software, achieving a maximum throughput of over 120 Mb/sec for the encryption of 7 KB of data. For even larger encryption blocks, the performance approaches the maximum performance of the triple-DES core which is 136 MB/sec. From these measurements, we can also conclude that the host to FPGA interface provided by Pilchard does not impose a bottleneck on the system.

5.3 FreeS/WAN Benchmarks

In tests using FreeS/WAN, the encryption algorithm was chosen to be triple-DES in CBC mode and the authentication algorithm MD5-96. This configuration is referred as 3des-md5-96 in FreeS/WAN and is the default encryption and authentication mode suggested by FreeS/WAN.

ttcp [12, 13] was used to measure the throughput of the system and benchmarks were conducted for both TCP and UDP protocols. Different parameters for ttcp were selected and tested and it was found that they did not have major effect on the results. As a result, the benchmarks were conducted using the default settings of 8192 (source buffer) and 2048 (network buffer) bytes respectively. Another utility, iperf, was used to verify the results obtained by ttcp.

For every packet sent out in single way connection, an acknowledgment packet is received. The acknowledgment packet is small in size and does not favor the use of triple-DES accelerator. For TCP, we estimate that 50% of the total number of packets are small packets because there is an acknowledgement for all packets sent. Note also that during the benchmark, the only traffic on the network was that generated by our testing software so no collisions are likely to occur.

Thus for the encryption of small blocks of data, the software implementation in LibDES rather than the hardware accelerator was used. Unfortunately, this limits the speedup which can be obtained.

In Table 3 it can be seen that the UDP performance of FreeS/WAN without IPSec is close to the maximum bandwidth of a 100 Mbit network, and for TCP, 67 Mb/sec is achieved. When encryption is used, the performance of both TCP and UDP falls by approximately 50%. Thus it can be concluded that encryption limits the speed of the VPN. The large difference in throughput between TCP and UDP was unexpected. Although we are uncertain as to the cause of this inefficiency, it is consistent with other published results [14].

Table 4 shows the performance of FreeS/WAN using the triple-DES accelerator, measured with ttcp. The hardware accelerated version offers a 30% improvement for the TCP protocol over the original software implementation and a 16% improvement for the UDP protocol.

This result was somewhat disappointing since the FPGA based accelerator was three times faster than the software implementation. We attribute the poor performance to the following issues:

- The triple-DES encryption only accounts for 50% of the VPN's total computation time. Amdahl's law applies to parallel computing and states that if α is the fraction of the computation that cannot be parallelized and P are the number of parallel processors, then the maximum speedup S that can be achieved is given by $S = \frac{1}{\alpha+(1-\alpha)/P}$. Adapting this idea to the VPN accelerator case with $P = 3$ (the accelerator is 3× faster than software) and $\alpha = 0.5$, one can see that a maximum speedup of 50% can be achieved.

- The speedup of the accelerator over software (Figure 7) is not constant. For small packets, $P < 3$

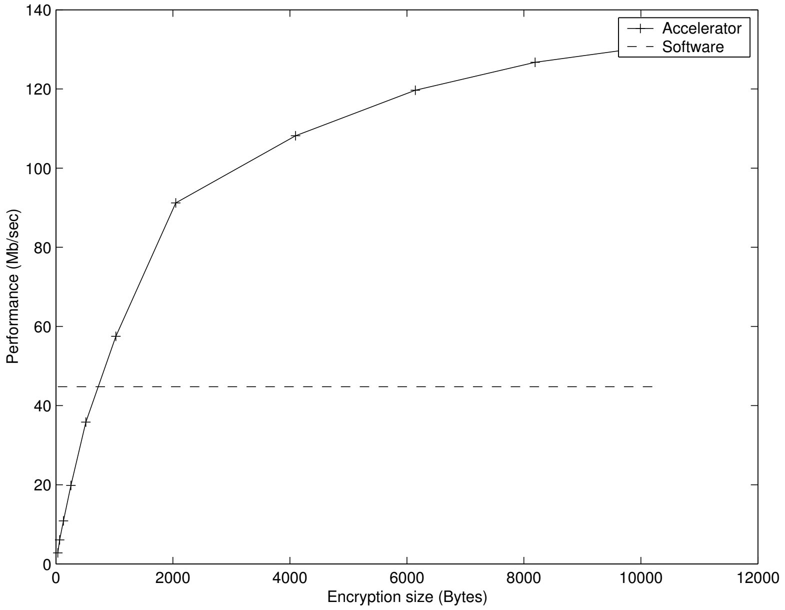

Figure 7: Performance of triple-DES accelerator for different encryption sizes.

Table 3: ttcp measured performance with and without FreeS/WAN

Protocol	Side	Throughput no encryption (in Mb/sec)	Throughput with encryption (in Mb/sec)	Performance degradation (%)
TCP	sender	67.024	35.448	47.72
TCP	receiver	66.968	35.360	47.19
UDP	sender	93.848	45.560	51.45
UDP	receiver	93.536	45.536	51.32

Table 4: Benchmark of ttcp with FreeS/WAN using Pilchard based accelerator

Protocol	Side	Throughput Mb/sec	Performance Improvement (%)
TCP	sender	45.788	29.1
TCP	receiver	45.660	29.1
UDP	sender	53.021	16.4
UDP	receiver	52.882	16.1

and our computed α is in fact greater than 0.5. Thus the maximum speedup obtainable is actually less than 50% for our accelerator since in practice, there are a large number of small packets to be encrypted.

6 Conclusion

The objective of this work was to develop an FPGA-based accelerator for virtual private network and explore systems issues in its integration. A hardware implementation of a triple-DES accelerator in CBC mode implemented on the Pilchard platform achieved a maximum throughput of more than 120 Mb/sec which was 3× faster than that of the highly optimized LibDES software implementation on an 866 MHz Pentium III machine. This FPGA-accelerated core was integrated with the FreeS/WAN VPN implementation. The resulting system offered a 1.3× and 1.16× improvement in end-to-end throughput for the TCP and UDP protocols respectively, the performance being limited by the latency of our bus interface, limiting speedups to be achievable only for large packets. Faster hardware accelerators with better performance on small packets could lead to further gains, with improvements up to a doubling in throughput being possible.

References

[1] E. Biham, "A fast new DES implementation in software," *Lecture Notes in Computer Science*, vol. 1267, pp. 260–272, 1997.

[2] "ftp://ftp.psy.uq.oz.au:/pub/crypto/des/libdes-x.xx.tar.gz."

[3] "http://www.free-ip.com/des/index.html."

[4] D. C. Wilcox, L. G. Pierson, P. J. Robertson, E. L. Witzke, and K. Gass, "A DES ASIC suitable for network encryption at 10gbps and beyond," in *Proceedings of first International Workshop on Cryptographic Hardware and Embedded Systems (CHES'99)*, pp. 37–48, 1999.

[5] C. Patterson, "High performance DES encryption in Virtex FPGAs using JBits," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 113–121, April 2000.

[6] S. Trimberger, R. Pang, and A. Singh, "A 12Gbps DES Encryptor/Decryptor core in an FPGA," in *Proceedings of the Cryptographic Hardware and Embedded Systems Workshop (CHES)*, pp. 156–163, Springer, 2000.

[7] National Institute of Standards and Technology (U. S.), "Data Encryption Standard (DES)," Federal information processing standards publication 46-2, National Institute for Standards and Technology, Gaithersburg, MD, USA, 1994. Supersedes FIPS PUB 46-1-1988 January 22. Category: computer security, subcategory: cryptography. Shipping list no.: 94-0171-P. Reaffirmed December 30, 1993.

[8] United States. National Bureau of Standards, *Data Encryption Standard*, vol. 46 of *Federal Information Processing Standards publication*. Gaithersburg, MD, USA: U.S. National Bureau of Standards, 1977.

[9] RSA Labs, "DES III Challenge," 1999.

[10] National Institute of Standards and Technology (NIST), "Data Encryption Standard (DES)." Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3), Oct. 1999.

[11] P.H.W. Leong, M.P. Leong, O.Y.H. Cheung, T. Tung, C.M. Kwok, M.Y. Wong, and K.H. Lee, "Pilchard – a reconfigurable computing platform with memory slot interface," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2001 (to appear).

[12] "http://www.dtic.mil/ttcp/."

[13] "http://www.cisco.com/warp/public/471/-ttcp.html."

[14] "http://www.dl.ac.uk/TCSC/disco/Beowulf/pc32-450/comms.html," 2002.