Configurable Hardware Implementations of Bulk Encryption Units for Wireless Communications

Paris Kitsos and Odysseas Koufopavlou
Electrical and Computer Engineering Department, University of Patras, Greece

Abstract: *Hardware implementations of bulk encryption units for wireless communications are presented in this paper. These units are based on the Triple DES (TDES) block cipher. The hardware modules can be configured in order to implement either the TDES or the DES block cipher. Three different hardware implementations of TDES are proposed. The first two implementations are based on the pipeline design technique, while the third implementation uses the traditional feedback logic design technique (looping). In addition, the DES block cipher's S-BOXes have been implemented by Look Up Tables (LUTs) and/or ROM blocks. Comparing with the LUTs, the ROM blocks implementation approach provides higher performance. But, the LUTs implementation approach is used in cases where the ROM blocks are not available. For high-speed performance applications the loop unrolling architecture is selected. The proposed implementation of this architecture achieves 7.36 Gbps data throughput whilst the 16-stage pipeline 2.45 Gbps. The implementation data throughput which is based on the looping architecture is 121 Mbps, but is used significant less hardware resources.*

Keywords: *Triple-DES, DES, block cipher, S-Box, cryptography, VLSI implementation.*

Received April 17, 2003; accepted August 3, 2003

1. Introduction

The major goal of several mobile communication standards, included GSM [8], GPRS [5], and WAP [14], is to meet the requirements of wireless data communication. The needs for mobility and power savings demand efficient implementations, which reduce the required hardware resources. In addition, communication standards are being developed for local area network communications. The High Performance Radio Local Area Network, type 2 (HIPERLAN/2) [4] and Asynchronous Transfer Mode (ATM) [1] are two of them. These standards provide high-speed data rates, and especially HIPERLAN/2 provides high-speed communication access to different broadband core networks as well as mobile terminals. For this reason efficient cryptography algorithm implementations are required in order to satisfy the requirements of all these communication protocols. Usually the offered security level of a communication system depends on the services and applications. For example, although the HIPERLAN/2 uses the DES block cipher for encryption, in the cases with high security level the TDES block cipher is used.

Three different hardware implementations of the triple-DES block cipher are presented, in this paper. The first two are based on the pipeline design technique and are suitable for high-speed applications. The third is based on the consecutive iterations and is preferable in applications with limited area resources. For the implementation of the TDES S-BOXes two different design approaches were used, Look Up Tables (LUTs) and ROM blocks. Based on our research, the ROM blocks implementations of the S-BOXes provide higher performance than the LUTs implementations. But the LUTs are used in cases where ROM blocks are not available.

Lately, many designs have been proposed for the FPGA hardware implementation of the TDES [2, 9, 10, 12]. The proposed in this paper implementation achieves higher data throughput than the implementations in [2, 9] but slower than the implementation in [12]. In [10] an ASIC implementation was proposed. Of course this implementation achieves higher data throughput than the proposed in this paper implementation by a factor range from 0.3 to 0.4.

The paper is organized as follows. In sections 2, the TDES block cipher is briefly described. The proposed hardware implementations are presented and explained in details in section 3. Synthesis results for the FPGA implementations are shown in section 4, and finally in section 5 the paper conclusions are given.

2. Triple DES Block Cipher

The Data Encryption Standard (DES) was published by the National Bureau of Standards in 1977 [3] and reaffirmed in its final form by the Federal Information Processing Standards Publication (FIPS) in 1994 [6].

DES is a block cipher with Feistel [13] networks, which operates on data blocks of 64-bit with a key value support of 64-bit length.

Triple DES is built on three DES block cipher in order to support a higher security level. It operates on the Encryption-Decryption-Encryption (EDE) mode, which uses sequentially first DES encryption, then DES decryption and last DES encryption, with the support of three different keys. The total keys length therefore is 3x64= 192 bits. The EDE mode is illustrated in Figure 1. The decryption operation of the TDES is performed such as DED mode.

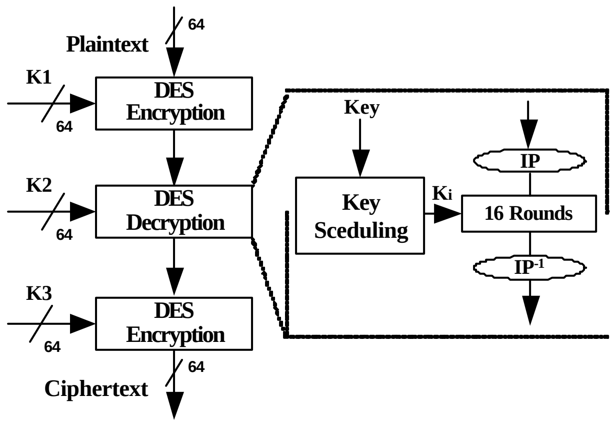

Figure 1. The triple DES block cipher architecture.

3. Proposed TDES Implementation

In Figure 2, the proposed loop unrolling pipeline TDES architecture is illustrated. It consists of 48 pipeline stages. The data buffering between the pipeline stages is achieved by using pipeline registers between the stages. The DES architecture supports encryption and decryption modes, because the Triple-DES algorithm in the encryption-decryption-encryption (EDE) scheme demands both modes. DES decryption uses the same algorithm as encryption. The only difference is the subkeys generation. The decryption subkeys have to be generated in a reverse order of the encryption subkeys. So, the encryption subkeys are cyclically shifted left and the decryption subkeys are cyclically shifted right. Each DES execution starts with the initial permutation IP and ends with the inverse initial permutation IP^{-1}. These two permutations are inverse operations. When three DES are concatenated, the initial permutation of the previous DES follows the inverse initial permutation of the current DES. There is no reason to do either permutation, since the result is no permutation at all [7]. After that, any IP-IP^{-1} pairs can be factored out of the algorithm and only the first and the last IP^{-1} need to be done. As a result, the initial permutations of the second and third DES, and the final permutations of the first and the second DES were not included in the proposed architecture. With this technique a significant performance gain is achieved.

The proposed architecture is able to process simultaneously 48 independent data blocks, so the data throughput is drastically increased. The DES key scheduling can be performed on the fly. The sub-keys are generated by the three key scheduler units as shown in Figure 2. Each, key scheduler generator consists of 16 rounds. The 64-bit input key is initially permuted, then is shifted by an appropriate hardwired shifter and finally is passed through a second round permutation. The key scheduler generators are implemented with pipeline stages in order to balance the pipelining in each TDES round. A 64-bit register is placed in the input of each of the three key scheduler generators. This register stores the user 64-bit key in order to force the generation of the appropriate sub-key on the appropriate time. On the 1st clock cycle the first 64-bit encryption key is applied on the key scheduler, whils on the 17th clock cycle the decryption key is applied. Finally, on the 33rd cycle the second encryption key is forced and DES operates in the encryption mode.

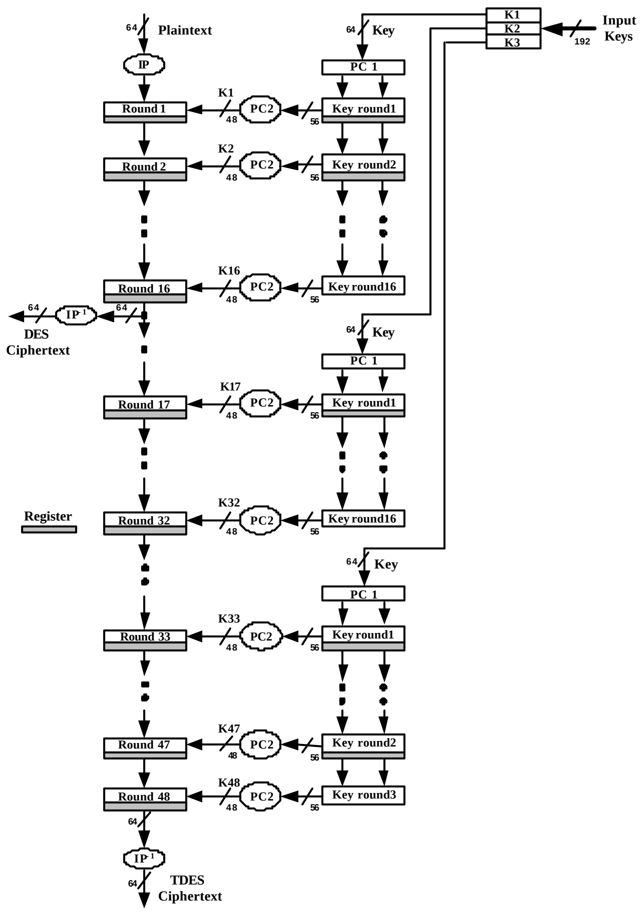

Figure 2. The 48 stage pipeline TDES implementation.

The second architecture consists of one DES with 16 pipeline stages as shown in Figure 3. The multiplexer after the IP subunit chooses between the new data and the output data of the previous DES execution. This architecture is able to process simultaneously 16 independent data blocks. The DES key scheduling can be performed on the fly. The subkeys generation comprises 16 rounds. The generation of the sub-keys is similar as in the previous first TDES architecture in Figure 2. The described architectures are suitable for high-speed applications that support Electronic Codebook (ECB) or ATM-Counter mode of operation.

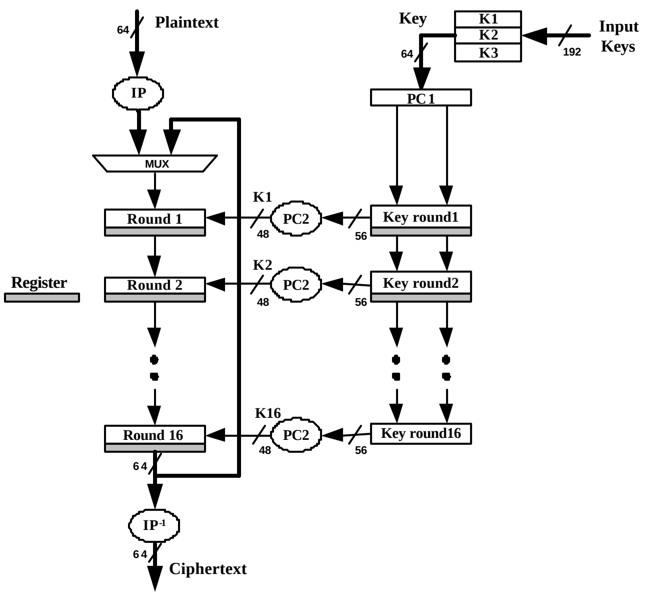

Figure 3. The 16 stage pipeline TDES implementation.

The third architecture as shown in Figure 4, consists of consecutive iterations. Only one stage is implemented in order to minimize the TDES implementation area resources. So, the required hardware allocated area resources of this architecture, comparing with the two, above described, pipeline architectures, are reduced by a factor equal to forty eighty and sixteen respectively.

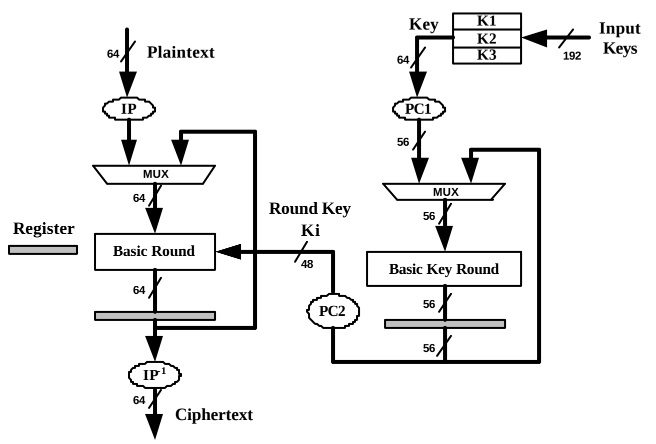

Figure 4. The consecutive iterations TDES implementation.

The output of the basic round is buffered and one additional register is used for the input plaintext store. During initialization the multiplexer chooses the plaintext whilst during the data transformation chooses the output of the basic round. In this architecture the key scheduler consists of one basic round. Consecutive round sub-keys are computed by simple rotations and permutations. The key scheduler forces the TDES by one sub-key at every clock cycle. Totally, 48 clock cycles are needed for the TDES block cipher execution. This architecture is suitable for limited area devices and for the Cipher Block Chaining (CBC) or Output Feedback (OFB) mode of operation.

The three above described TDES architectures could operate either with the same three keys, or with two different keys. In the case that the three keys are the same the TDES has the same crypto strength as simple DES. The correct operation of the above architectures is controlled by a control unit (not included in the Figures). This control unit is responsible for the configuration of the system in order to operate as a TDES or as a single DES. If a single DES is selected, the control unit stops the operation after 16 clock cycles.

4. VLSI Implementation Results

The proposed TDES architectures were captured by using VHDL. All the system components were described with structural architecture. The whole design was synthesized, placed and routed for XILINX FPGA devices [15]. Then all implementations were simulated again for verification. The TDES correct operation is validated by using the know-answer test vector, provided by [11].

The TDES S-BOXes have been implemented with LUTs as well as with ROM Blocks. The synthesis results for all implementations are illustrated in Tables 1, 2 and 3. From these results it is apparent that ROM blocks approach have higher performance than the LUTs S-BOXes .

Table 1. The 48 stage pipeline (loop unrolling) TDES implementations synthesis results.

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V1600EBG560		Xilinx V1600EB G560	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	326	81 %	326	81 %
Fun. Generators	28510	92 %	28380	91 %
CLB Slices	14240	91 %	14142	91 %
Dffs or Latches	10400	31 %	10400	31 %
F (MHz)	108		115	
Throughput (Gbps)	6.9		7.36	

Table 2. The 16 stage pipeline TDES implementations synthesis results.

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V400EFG676		Xilinx V400EFG676	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	326	73 %	326	73 %
Fun. Generators	9504	99 %	9462	98.8 %
CLB Slices	4752	99 %	4715	99 %
Dffs or Latches	3472	32 %	3472	32 %
F (MHz)	108		115	
Throughput (Gbps)	2.3		2.45	

Table 3. The one-round TDES implementations synthesis results.

Implementation	LUT		ROM	
FPGA DEVICE	Xilinx V200EBG352		Xilinx V200EBG352	
AREA ALLOCATION	Used	Util.	Used	Util.
I/Os	198	76 %	198	76 %
Fun. Generators	862	18 %	835	18 %
CLB Slices	431	18 %	405	17.5 %
Dffs or Latches	400	7 %	400	7 %
F (MHz)	86		91	
Throughput (Mbps)	115		121	

Comparisons between the proposed TDES implementations and other previous published designs are shown in Table 4. According to our knowledge, the 48-stage pipeline TDES implementation is the first published implementation. So, only the second and third proposed designs are compared with previous implementations.

Table 4. TDES implementations comparison.

Implementation	One Round		Full Pipeline	
TDES Architecture	F (MHz)	Data rate (Mb/s)	F (MHz)	Data rate (Gb/s)
TDES in [9]	69	83	-	-
TDES in [2]	91	116	91	1.5
TDES in [12]	-	-	207	13.3
TDES in [10]	250	155	-	-
Proposed_ROM	91	121	115	2.45
Proposed_LUT	86	115	108	2.3

In [9], two cascade rounds are implemented. The first is used for encryption and the second for decryption mode of operation. This has the impact that increases the critical path of the algorithm. In [2], two key scheduling round are implemented in order to execute the encryption and decryption mode of operation. In addition, one extra multiplexer in order to control the encryption or decryption key is used. So the critical path of the TDES is determined by the key scheduling circuit that is bigger than the proposed. In [12] a very high throughput implementation with 144 pipeline stages is proposed.

The proposed TDES implementations provides higher data throughput than the implementations in [2, 9], but lower than the implementation in [12]. In [10], the TDES is implemented in an ASIC device. The area resources of the proposed implementations are significantly less than the resources of the implementations in [2, 9].

5. Conclusion

Configurable hardware implementations of bulk encryption units for wireless communications are presented in this paper. Three different triple DES hardware implementations are proposed. The algorithm's S-BOXes have been implemented either in Look Up Tables (LUT) and/or RAM blocks. The proposed designs provide high-speed performance and reduce the required area resources. They are more efficient in terms of area resources than many previous implementations. The different characteristics of the three implementations provide the ability of selection, according to the application requirements. The proposed designs was captured entirely in VHDL language and implemented in XILINX FPGA devices. Measurement results and comparisons between the proposed and previous hardware implementations are presented.

References

[1] ATM Forum, *ATM Security Specification Version 1.0*, ATM-SEC-01.0100, The ATM Forum, Security Working Group, 1999.

[2] Chodowiec P., Gaj K., Bellows P., and Schott B., "Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board," *in Proceedings of Information Security Conference*, Malaga, Spain, pp. 220-234, October 2001.

[3] Data Encryption Standard, Federal Information Processing Standard (FIPS) 46, National Bureau of Standards, 1977.

[4] ETSI TS 101 761-1 V1.2.1, Broadband Radio Access Networks (BRAN), HIPERLAN Type 2, Data Link Control (DLC) Layer, Part 1: Basic Data Transport Functions, 2000.

[5] ETSI TS 148 018, Digital Cellular Telecommunications System (Phase 2+), General Packet Radio Service (GPRS), Base Station System (BSS)-Serving GPRS Support Node (SGSN), BSS GPRS Protocol, May 2002, available at http://webapp.etsi.org/action/PU/20020611/ts_148018v050300p.pdf/.

[6] Federal Information Processing Standards Publication 140-1, "Security Requirements for Cryptographic Modules," *U. S. Department of Commerce/ NIST*, Springfield, VA: NIST, 1994.

[7] Feldmeier C. D. and Karn R. P., "UNIX Password Security-Ten Years Later," *CRYPTO'89*, Santa Barbara, California, USA, pp. 44-63, 1989.

[8] Global System for Mobile Communications, Specifications, available at http://www.etsi.org/.

[9] Kwon O., Seike H., Kajisaki H., and Kurokawa T., "Implementation of AES and Triple-DES Cryptography Using a PCI-based FPGA Board," *in Proceedings of the International Technical Conference on Circuits/ Systems*, Computers and Communications 2002, ITC-CSCC-2002, Phuket, Thailand, July 16-19, 2002.

[10] Leitold H., Mayerwieser W., Payer U., Posch C. K., Posch R., and Wolkerstorfer J., "A 155 Mbps Triple-DES Network Encryptor," *in Proceedings of Cryptographic Hardware and Embedded Systems (CHES' 2000)*, USA, August 2000.

[11] NIST Special Pubilication 800-20, "Modes of Operation Validation System for the Triple Data Encryption Algorithm," *National Institute of Standard and Technolog*y, 2000.

[12] Pasham V. and Trimberger S., "High Speed DES and Triple-DES Encryptor/ Decryptor," on line available in http://www.xilinx.com/xapp/ xapp270.pdf/, August 2001.

[13] Schneier B., *Applied Cryptography, Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, 1994.

[14] Wireless Application Protocol Forum, http://www wapforum.org/.

[15] Xilinx, *Virtex: 2.5 V Field Programmable Gate Arrays*, available at www.xilinx. com/, San Jose, California, USA, 2001.

Paris Kitsos received the BSc in physics from the University of Patras, Greece. He is currently pursuing his PhD in the Department of Electrical and Computer Engineering at the University of Patras. His research interests include VLSI design, hardware implementations of cryptography algorithms, security protocols for wireless communication systems, and Galois field arithmetic implementations. He has published many technical papers in the areas of his research.

Odysseas Koufopavlou received the Diploma of electrical engineering in 1983 and the PhD degree in electrical engineering in 1990, both from University of Patras, Greece. From 1990 to 1994 he was at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Currently, he is an associate professor with the ECE Department, University of Patras. His research interests include VLSI design, VLSI crypto systems, and high performance communication subsystems. Dr. Koufopavlou has published more than 90 technical papers and received patents and inventions in these areas. He served as general chairman for the IEEE ICECS'1999.