A 155 Mbps Triple-DES Network Encryptor*

Herbert Leitold, Wolfgang Mayerwieser, Udo Payer, Karl Christian Posch, Reinhard Posch, and Johannes Wolkerstorfer

Institute for Applied Information Processing and Communications,
Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria
Johannes.Wolkerstorfer@iaik.at
http://www.iaik.at

Abstract. The presented Triple-DES encryptor is a single-chip solution to encrypt network communication. It is optimized for throughput and fast switching between virtual connections like found in ATM networks. A broad range of optimization techniques were applied to reach encryption rates above 155 Mbps even for Triple-DES encryption in outer CBC mode. A high-speed logic style and full-custom design methodology made first-time working silicon on a standard 0.6 μm CMOS process possible. Correct functionality of the prototype was verified up to a clock rate of 275 MHz.

Keywords. Network security, encryption, DES algorithm, Triple-DES, cipher block chaining, pipelining, true single-phase logic, full-custom design.

1 Introduction

Modern network technology offers transmission rates in the multi megabit range. In addition, Quality of Services (QoS) parameters like throughput and latency are guaranteed. These parameters make the transmission of voice and video in addition to normal LAN data possible. Whenever public accessible infrastructure is involved, mechanisms to secure confidential information are required. The Triple-DES algorithm in the Cipher Block Chaining (CBC) mode of operation meets these requirements [4], but demands considerable design effort to reach the desired throughput. Sustaining QoS-parameters even for short data packets requires an architecture where keys and encryption modes can be changed quickly. This task is far from trivial. Only dedicated hardware solutions can provide these properties by applying strong encryption.

This paper describes a single-chip context-agile encryption unit which is capable of encrypting (or decrypting) at rates of 155 Mbps using various DES-related algorithms. With respect to speed, the most demanding choice is Triple-DES in outer CBC mode. However, a network employing statistical multiplexing, such as the Asynchronous Transfer Mode (ATM), imposes another bottleneck: each

* The work described origins from the European Comission funded Project *Secure Communications in ATM Networks (SCAN)* established under contract AC0330 in the Advanced Communications Technologies and Services (ACTS) Program.

Ç.K. Koç and C. Paar (Eds.): CHES 2000, LNCS 1965, pp. 164–174, 2000.

user connection asks for its own encryption context consisting of keys and mode of operation. The time span between getting to know the identifier of a new user connection and the actual use of the related keys is too short. Moreover, two directions of data flow need to be served by a common key repository, and both directions may ask for accessing different keys at the same time.

The bottleneck of replacing keys rapidly arises due to the nature of ATM and is referred to as key-agile encryption [7]. ATM is a relaying technique operating on data units of a fixed size - called cells. ATM cells are relatively small units and consist of five byte header information and 48 byte payload. ATM is a connection oriented technology employing virtual connections (VCs) that are identified by a 24 bit value in the cell header. As ATM cells are statistically multiplexed between VCs, replacing the session key may be required for each ATM cell. Further requirements may even strive for assigning different encryption algorithms to each VC, such as DES and Triple-DES, which is referred to as algorithm-agile encryption [8]. Actually, the encryption unit presented in this paper allows to uniquely assign the encryption context including the operational mode to each connection which is called context-agile encryption [6].

The remainder of this paper sketches in section 2 general constraints arising from the application and their architectural impacts. Section 3 presents implementation details with emphasis on high-speed optimization techniques. Measuring results of the produced silicon and the prototype Network Interface Card (NIC) are presented in section 4. Finally, conclusions are drawn and future work is discussed.

2 Architecture

High-speed digital hardware can take advantage of exploiting parallelism. The more parts of a circuit work in parallel, the more data can be processed. For a network encryptor this is especially true with respect to the number of encryption modules [9]. Unfortunately, encryption in the CBC mode of operation requires the result of the previous encryption in order to process the current block. Thus, a parallelized architecture with more encyption modules would only speed up the electronic code book (ECB) mode and does not improve performance in general.

It might be assumed that other operational modes, such as the counter mode (CM) do not have the drawback of the CBC mode and can use multiple encryption modules in parallel. Nevertheless, CBC has excellent properties regarding synchronization. When ATM networks drop cells in periods of congestion, cryptographic resynchronization is required. The CBC mode re-establishes synchronization within two blocks when multiples of the block size get lost - as in the case of lost ATM cells. In contrary, the CM requires an explicit mechanisim to re-establish synchronization. This turns out to be major advantage of CBC, even if it forms tougher constraints on the crypto hardware.

In a network application, only two encryption modules can work independently as depicted in Figure 1. The first module encrypts the Down-Stream,

where data is sent to the network. The second one decrypts the Up-Stream, which receives data from the network.

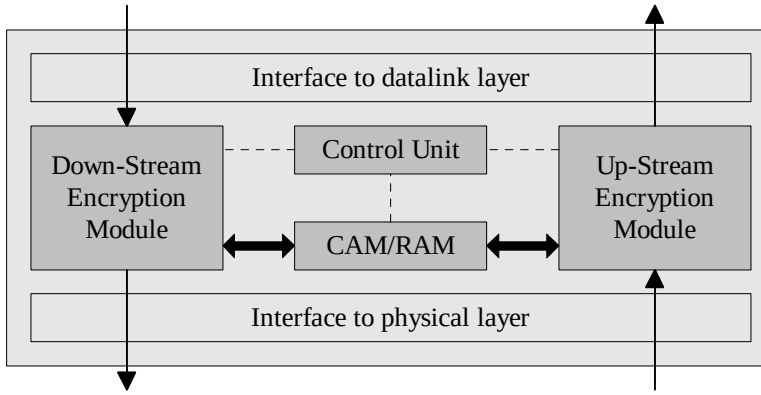

Fig. 1. Architecture of the network encryptor

2.1 Virtual Connections

When choosing an architecture with two encryption modules, connection parameters like session keys are loaded through the Down-Stream into the encryptor to avoid an extra interface. For every virtual connection, these data are stored in a CAM/RAM module for later retrieval. The CAM/RAM module is addressed with the 24-bit value identifying the virtual connection. Eight bits identify the virtual path, the remaining 16 bits identify the virtual channel. We do not distinguish between the virtual path identifier and the virtual channel identifier and denote all 24 bits as VCI. The VCI is part of the header information of a cell. It precedes the user data which is called payload. Encryption applies only to the payload. The header information is unaffected to preserve routing mechanisms. Each time a cell is processed, encryption parameters like type of algorithm, mode, session keys and initial vectors are retrieved from CAM/RAM. The worst case access rate for CAM/RAM is derived in Equation 1. The calculation is based on a 155 Mbps STM-1 signal used in ATM networks which offers a net bandwidth of 149.8 Mbps. It has a fixed cell size of 48 bytes payload and five byte header information.

$$f_{CAM} = \frac{bandwidth}{cellsize} = \frac{149.76\ Mbps}{(5+48)\,8\ Bit} = 353.2\ kHz = \frac{1}{2.83\ \mu s} \tag{1}$$

During these 2.83 μs the cell has to be identified by its VCI and the according connection parameters have to be copied from CAM/RAM into the encryption

module. In order to raise parallelism, this is done during encryption of the previous cell. Data is retrieved sequentially from CAM/RAM and stored in intermediate buffers for keys A and B and the initial vector (KEYA, KEYB, IV) as depicted in Figure 2. When encryption starts, data and encryption parameters are loaded concurrently into the DES module.

2.2 Encryption Module

The encryption module (Figure 2) is instantiated twice in the network encryptor: once for Down-Stream encryption and a second time for Up-Stream decryption. Each can perform both DES encryption and decryption, because the Triple-DES algorithm with two keys in the encryption-decryption-encryption (EDE) scheme demands both. The 16 rounds of a DES operation are executed sequentially. In our approach each round consumes two clock cycles which yields small logic functions that are convenient for a high-speed circuit. A version consuming one clock cycle per round would spend relatively more time for loading than for encryption, would have bigger logical functions and would demand a more complicated sub-key generator. Speed optimization would still be necessary and would not be simpler as for the two-cycle-variant. The overall-effort would be even higher. A pipelined version of the DES-round hardware makes no sense, because in the CBC mode no DES-blocks can be processed concurrently. Loading a 64-bit data block takes 10 clock cycles and occurs concurrently to unloading the previously processed block.

Encryption data is loaded from a First In First Out (FIFO) buffer which collects data bytewise from an asynchronous interface. The output FIFO buffers an encrypted block for asynchronous output. A complete DES encryption - loading included - takes 42 clock cycles, a Triple-DES encryption 108 cycles and plaintext loading 12 cycles. The CBC mode requires no additional clock cycles. Its XOR operation is done during loading for encryption and during unloading for decryption. For the sake of simplicity, a detailed descripton of the CBC dataflow is omitted, but it should be mentioned that the need for updating initial vectors (IV) in CAM/RAM and the need to treat the first block of a cell differently from subsequent ones raises the complexity of a hardware solution significantly.

2.3 Troughput

Triple-DES encryption with or without block chaining is the worst case scenario for throughput considerations. The required clock speed can be derived from encrypting a complete ATM cell in this mode as shown in Equation 2.

$$f_{clk} = f_{CAM} \times cycles = 353.2\ kHz \times (6 \times 108 + 12) = 233.1\ MHz \quad (2)$$

In practice, a clock speed close to 250 MHz is needed because retrieving data from CAM/RAM takes longer than encrypting the previous block. The resulting idle time of the DES module is compensated by a higher clock speed.

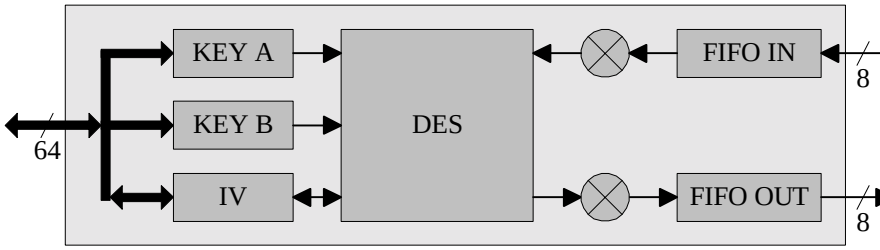

Fig. 2. Architecture of the encryption module

3 Circuit Implementation

The throughput calculation given above makes clear that a conventional chip design methodology like a standard-cell approach cannot cope with a clock speed of 250 MHz. This is especially true when a widely available CMOS process is to be used. We selected a standard 0.6 μm process from AMS International that offers a single polysilicon layer, two metal layers, and an option for a third metal layer. The nominal supply voltage is 5.0 Volts for the core and IO.

Those parts of the circuit which have to run at 250 MHz were designed using a full-custom design methodology. They exploit the true single-phase logic style that exhibits appropriate parameters for high speed applications. Due to the high design effort for a full-custom approach, we partitioned the circuit into a high-speed block (250 MHz) and a low-speed block. The latter operates with one quarter of the clock frequency (62.5 MHz). The low-speed block is assembled by auto-routed standard-cells synthesized from a VHDL description.

3.1 Standard-Cell Circuit

The standard-cell circuit handles the cell-level and block-level control of the network encryptor. It monitors input data of the Up-Stream and Down-Stream that are collected in the input FIFOs of the encryption modules. There it detects cell borders on basis of a start of cell (SOC) bit that is stored in addition to data. From the header information the cell type is identified. Roughly spoken, three types of cells are distinguished: key-download cells, signalling cells, and user cells. Key-download cells are used to update connection parameters of a virtual connection in CAM/RAM. They are only accepted in the Down-Stream to prevent malicious manipulation via the network. Signalling cells are passed to the output without any alteration to maintain the routing mechanism. When a user cell is identified, the connection parameters according to its VCI are retrieved from CAM/RAM. The cell's header information is passed unaltered to the output, and the payload is encrypted with the encryption algorithm, the encryption mode, the keys, and the initial vectors as stated in the CAM/RAM entry. If no entry was found in CAM/RAM, the cell will be output unaltered if

the static external signal "feedthough" is set to high. Otherwise the complete cell will be dropped. The cell-level control of Up-Stream and Down-Stream is completly independent, but their CAM/RAM access is shared. An arbitration unit schedules the access and grants higher priority to the Up-Stream to avoid data congestion and cell loss.

The standard-cell circuit also controls the block level. It starts the encryption module and cares for pipelined dataflow from the input FIFO to the DES module and from there to the output FIFO. Header blocks have to be treated differently because they have only five bytes instead of eight bytes. Another pecularity is the encryption of the first payload block in the CBC mode. It requires an initial vector from CAM/RAM, whereas subsequent blocks take the previous result as IV. After encrypting a cell's last block, the IV entry in CAM/RAM has to be updated. When no further data is available in the input FIFO, the block has to be moved into the output FIFO without pipelined loading of new data.

3.2 Full-Custom Encryption Module

The encryption module is a full-custom circuit for processing 64-bit data blocks. Besides plaintext operation, it supports two different algorithms: DES and two-key Triple-DES in EDE scheme. As modes of operation, ECB and CBC are supported for all algorithms without affecting throughput. Pipelined loading and unloading of the DES module is possible, because input FIFO and output FIFO act as buffers. Each FIFO is able to hold one and a half data blocks and has an asynchronous interface for off-chip communication. This interface conforms to the defacto standard UTOPIA [2], which connects the ATM layer with the physical layer in ATM components.

Datapath Optimization. The highest clock frequency at which a digital circuit has correct functionality depends on its crictical path. The crictical path is the part of circuit that needs most time for evaluating its logic function. In case that the evaluation time exceeds the cycle time of the clock, erroneous output will result. High-speed optimization basically tracks down critical paths repeatedly until the desired clock rate including a safety margin is reached.

In case of a hardware DES implementation, the critical path surely lies in the S-boxes which are used to substitute 6-bit values by 4-bit values. By granting a S-Box operation two clock cycles, the attainable clock frequency was nearly doubled. As an architectural consequence, every round of a DES encryption takes two cock cycles which in turn allows a simplified subkey generator design. The subkey generator has to rotate two 28-bit values up to two positions per DES round. Having to rotate only one position per clock cycle reduced the subkey generators functionality and improved its regular structure.

Control Logic. The control logic of the encryption module has to be clocked with the same frequency as the datapath. As in the datapath, a standard cell approach cannot be applied. Hence, a full-custom methodology was used to meet

the performance requirements. In contrary to the datapath, control logic lacks of regular structures. The design effort concentrates therefore on (hierarchical) decomposition. This strategy produces small subcircuits which can generate control signals adjacent to their controlled elements. Further, it is easier to generate layout for small subcircuits and to interconnect them to a complete circuit. Hierarchical decomposition also helps to cope with the functional complexity of controlling sequences. The control logic of the encryption module is split into two control machines. The first machine generates the control sequences for the 16 rounds of DES encryption and decryption. The second machine is able to perform Triple-DES encryption by starting the DES-controller three times. In addition, it controls the pipelined loading mechanism.

True Single-Phase Logic. As stated before, a semi-custom design methodology like a standard-cell approach does not reach the desired clock frequency of 250 MHz. Hence, a full-custom design methodolgy was applied which offers higher flexibility at the cost of additional design effort. Using this approach, a logic style was selected that has several benefits for speed optimized circuits: true single-phase logic (TSPL) [10].

TSPL is a dynamic logic style that combines combinational functionality with storage behaviour and thus offers low transistor count. It requires just a single clock signal which simplifies clock generation and clock distribution. Besides a complementary version of TSPL, like depicted in Figure 3, precharged N-latches can be built, where the P-logic block is replaced by a P-clock-transistor. Precharged N-latches just occur in ROM circuitry, which executes the S-box substitution of the DES algorithm. They speed up NOR structures significantly, but dissipate more power than a complementary version. This technique decreased the delay of the 256-bit S-box-ROMs to an acceptable level and defused the critical path. The logical function of complementary TSPL gates are kept simple to preserve their high speed. Especially, the number of P-transistors connected in series is kept low. Only 40 gate structures fulfilled these requirements, which made electrical and layout optimization a rewarding task due to their high reusability.

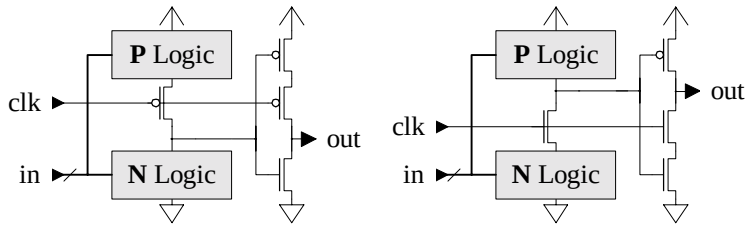

Fig. 3. Complementary true single-phase logic: P- and N-Latch

Clock Generation and Clock Distribution. When a clock signal for a synchronous high-speed circuit is distributed across a chip, delays caused by the distributed RC-effect of wires have to be considered. Especially long wires delay the signals by their parasitic sheet-resistance and sheet-capacitance that form a low-pass filter. The effective delay will vary across the chip area depending on the distance to the clock generation module (wire length) and the capacitive load. This delay variation is called clock skew. It can cause pipelining errors and data loss. Intersecting the clock net by inserting clock buffers solves this problem. A disadvantage of distributed clock drivers is the costly layout generation. For our circuit we chose a hybrid solution: two 'central' clocktrees for each encryption module shorten the average length of clock wires and their induced skew to an acceptable level.

The clock signal itself can either be fed via a pad into the circuit, when operating at clock frequencies up to 50 MHz, or it can be generated on-chip by a voltage controlled oscillator (VCO) module that is controlled by an analog pad. The VCO's frequency ranges from 25 MHz beyond 400 MHz. The clock signal is divided by four to clock standard-cell based modules. Their clock tree was automatically generated with place-and-route tools.

A problem similar to clock-distribution arises in the distribution of control signals. A control signal may have to drive many gates. Cascaded inverters are used to amplify the signal to an appropriate strength. In a high speed circuit, these inverters may cause a transport delay in the magnitude of a clock cycle. This delay can lead to erroneous behaviour. Our approach to avoid this situation was a limitation of the number of driven gates to 64. For higher gate counts, the control logic was doubled. Keeping the strength of the cascaded inverters moderate led to a reduction of the transport delay by slightly decreasing the steepness of the signal.

Layout. Layout and schematics of the encryption module were generated with Mentor GDT software. Regular structures - like the matrixces of the S-box-ROMs - were programmed by writing generators. Generators are used to instantiate interactively captured layout fractions and assemble them by wiring. The number of interactively captured layout cells was tried to be kept at a minimum which resultet in a library of highly reused leaf cells.

Special attention was paid to the floorplan. As mentioned above, the non-ideal behaviour of wires makes it necessary to keep routing distances as small as possible. The floorplan was optimized to avoid very long wires at cost of medium length wires. In the full-custom circuit, wires do not exceed the length of one millimeter. Wire length is also of interest when using TSPL cells. Their output signal should only be used near the cell, because it is not driven in all situations. In such a situation, the logic value is only held by the parasitc capacitance of the output node. If non-local interconnections are required, the insertion of static inverters overcomes this disadvantage.

Another floorplanning issue are the permutations of the DES algorithm. A straight forward implementation would require considerable routing area for per-

mutations. By exploiting regular structures of these permutations, routing area can be fairly reduced [5]. Figure 4 shows the complete layout of the chip. On the left side, two instances of the full-custom encryption module can be identified. Each has an area of 1.8 mm^2 and contains 32,000 transistors. The standard-cell circuit is located in the right half. The circuit's total die size is 23.7 mm^2 and counts nearly 120,000 transistors.

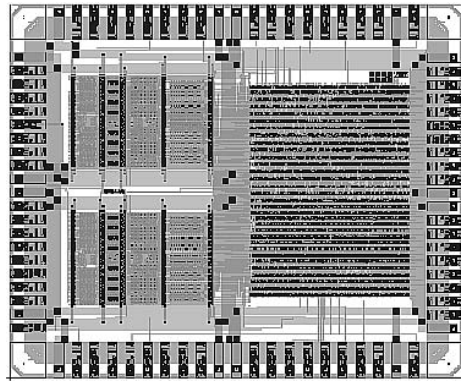

Fig. 4. Layout of the network encryptor

4 Measured Results

The functionality of the prototype chip samples (Figure 5) was verified on a chip tester. At a supply voltage of 5.0 Volts, correct functionality was verified up to a clock frequency of 275 MHz. Some samples even reached 290 MHz. These measurements match exactly circuit level simulations with extracted parasitics. A safety margin of more than 25 MHz ensures error-free behaviour at the target frequency of 250 MHz. When both encryption modules are held busy at this frequency, the circuit consumes 230 mA. At a supply voltage of 3.3 Volts, correct functionality is given up to 160 MHz.

The network encryptor chip was also verified in a real application. A conventional ATM network interface card was modified, that the encryptor chip could intercept communication between ATM layer and physical layer at the UTOPIA interface. The card has a PCI interface and software drivers for Windows 2000. It passed several basic tests.

5 Conclusion

Expertise in various domains like networking, security, hardware design, and high-speed optimization was necessary in order to implement a full functioning

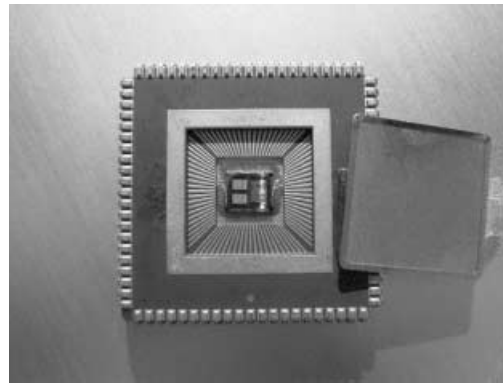

Fig. 5. Chip sample of the network encryptor

single chip for network encryption. The encryptor can concurrently encrypt and decrypt two 155 Mbps data streams with the Triple-DES algorithm in outer CBC mode. It operates at a clock frequency of 250 MHz. This combination of throughput and cryptographic strength was unattainable up to now. In addition, a sophisticated architecture allows encryption of several multiplexed virtual connections with different encryption algorithms, modes of operation, and keys without deteriorating Quality of Servce parameters.

Future work will include an expansion of the on-chip CAM/RAM memory size in order to support more virtual connections. This will improve the applicability in network interface cards. For use in network switches, an interface to external CAM/RAM will raise the number of virtual connections to a level that will satisfy even the needs of large networks. The prototype's interface for off-chip communication obeys the UTOPIA standard. Future versions will additionally support a microcontroller interface. This will open the scope of this chip for a broad range of applications, where high-speed encryption is asked for.

References

1. ATM Forum, ATM Security Specification Version 1.0, ATM-SEC-01.0100, The ATM Forum, Security Working Group, 1999.
2. ATM Forum, Utopia Level 2 - Version 1, af-phy-039.000, The ATM Forum, Technical Comittee, 1995.
3. ANSI, American Standard for Data Encryption Algorithm (DEA), ANSI 3.92, American National Standards Institute, 1983.
4. ANSI, American Standard for Information Systems Data Encryption Algorithm - Modes of Operation, ANSI 3.106, American National Standards Institute, 1983.
5. Hoornaert, F., Goubert, J. and Desmedt, Y., Efficient Hardware Implementation of the DES, Advances in Cryptology: Proceedings of CRYPTO '84, p. 147, Springer-Verlag, Berlin, 1985.

6. Lyndon G. Pierson, Edward L. Witzke, Mark O. Bean, Gerry J. Trombley, Context-Agile Encryption for High Speed Communication Networks, ACM SIGCOMM Computer Communication Review, vol. 29, no. 1., p. 35, 1999.
7. Daniel Stevenson, Nathan Hillery, Greg Byrd, Fengmin Gong, Dan Winkelstein, Design of a Key Agile Cryptographic System for OC12c Rate ATM, Internet Society Symposium on Network and Distributed Systems Security, San Diego, CA, 1995.
8. Thomas D. Tarman, Robert L. Hutchinson, Lyndon G. Pierson, Peter E. Sholander, Edward L. Witzke, Algorithm-Agile Encryption in ATM Networks, IEEE Computer, vol. 31, no. 9., p. 57, 1998.
9. D. Craig Wilcox, Lyndon G. Pierson, Perry J. Robertson, Edward L. Witzke, Karl Gass, A DES ASIC Suitable for Network Encryption at 10 Gbps and Beyond, Cryptographic Hardware and Embedded Systems: Proceedings of CHES '99, p. 37, Springer-Verlag, Berlin, 1999.
10. Jiren Yuan, Christer Svensson, High Speed Circuit Technique, IEEE J. Solid-State Circuits, vol. 24, p. 62, 1989.