

Implementation and Benchmarking of Hardware Accelerators for Ciphering in LTE Terminals

Sebastian Hessel, David Szczesny, Nils Lohmann, Attila Bilgic

Institute for Integrated Systems, Ruhr-Universität Bochum

D-44780 Bochum, Germany

Email: {sebastian.hessel,david.szczesny,nils.lohmann,attila.bilgic}@is.rub.de

Josef Hausner

Infineon Technologies AG

D-85579 Neubiberg, Germany

Email: hausner@ieee.org

Abstract—In this paper we investigate hardware implementations of ciphering algorithms, SNOW 3G and the Advanced Encryption Standard (AES), for the acceleration of the protocol stack layer 2 in the 3G Long Term Evolution (LTE). This analysis is based on timing requirements from execution time measurements in a simulated mobile phone platform, where we apply data rates of 100 Mbit/s and above (200 and 300 Mbit/s) to account for LTE and beyond LTE investigations. Different architectures for both algorithms are explored in order to meet the performance requirements, while keeping the power and area budget at a reasonable level. Therefore, a hardware analysis is done using a standard cell library of Faraday's 90 nm CMOS technology. Finally, the cryptographic substitution box with one-hot encoding emerges as the best solution for both ciphering schemes. Additionally, the 128-bit data path in the AES is identified as the most suitable architecture for LTE terminals, whereas a dual-AES approach turns out to be a candidate for data rates far beyond LTE (like LTE-Advanced).

I. INTRODUCTION

In cellular mobile communication systems, sophisticated security architectures are inevitable to protect private contents and to allow for a secure access to services. Therefore, user data confidentiality and integrity must be ensured by the use of cryptographic algorithms. These algorithms are characterized by a high computational complexity, while they are loaded with continuously growing data rates in upcoming mobile networks. The increased demands, with data rates up to 100 Mbit/s for the 3G successor Long Term Evolution (LTE) or even beyond, require the employment of dedicated hardware accelerators even for higher layer functionalities in the protocol stack. On the other hand, mobile handsets as classic embedded devices have limited resources like chip area and battery lifetime. The development of cryptographic hardware engines is therefore affected by system performance requirements as well as reasonable power and area demands.

In the data plane processing of the LTE protocol layer 2 (L2), the execution of the ciphering function is already identified as a time critical part in the software stack [1]. Its acceleration by the design of an efficient ciphering module is a first step in order to find a suitable system architecture for LTE terminals. In this work, we therefore present results from hardware implementations of the SNOW 3G algorithm and the Advanced Encryption Standard (AES), respectively, that are defined for user confidentiality in LTE. For both schemes, different architectures are explored, where we focus

on the identification of suitable cryptographic substitution box (S-Box) implementations. Additionally, different data path widths and an approach with two AES engines in parallel are investigated for the AES based ciphering scheme. VHDL models are generated and analyzed with regard to their power consumption and area effort by standard cell synthesis using Faraday's 90 nm CMOS technology library. Finally, power-area products of all analyzed approaches are given in order to identify the best solution. All results thereby refer to timing requirements from an execution time analysis at data rates of 100 Mbit/s and beyond (200 and 300 Mbit/s). This is done with a virtual system prototype (VSP) of a mobile phone platform. The VSP comprises a model of the LTE L2 protocol stack that runs on top of an emulated, ARM based hardware platform.

This paper is organized as follows: Section II provides an introduction to the ciphering schemes in LTE, while section III focuses on implementation aspects for these algorithms. A description of the VSP as well as the profiling results are provided in section IV. Finally, results of the hardware implementations are given in section V, leading to the conclusion in section VI.

II. USER DATA CONFIDENTIALITY IN LTE

In LTE, encryption and decryption of user data is performed in the Packet Data Convergence Protocol (PDCP) sublayer [2]. Ciphering is thereby applied to the data part in the PDCP Protocol Data Unit (PDU). Two EPS (Evolved Packet System) Encryption Algorithms (EEA) are specified by the 3rd Generation Partnership Project (3GPP): SNOW 3G (128-EEA1) and AES (128-EEA2) [3]. The stream cipher SNOW 3G is already used in 3G systems as a second set of algorithms for the confidentiality algorithm f8 [4], whereas the AES now replaces the KASUMI block cipher. Both algorithms adopt a 128-bit key that is provided by the Radio Resource Control (RRC). The SNOW 3G and the AES generate keystream blocks of 32 bits and 128 bits, respectively, that can be used for ciphering of the plaintext as well as for the inverse operation when the ciphertext is decrypted.

To show the architectural demands of both algorithms, a detailed view from an implementation perspective is given in the following subsections.

A. The 128-EEA1 Algorithm with SNOW 3G

The structure of the SNOW 3G is depicted in Fig. 1. It mainly consists of three parts: the Linear Feedback Shift Register (LFSR), the Finite State Machine (FSM) and a feedback part [5]. The LFSR contains 16 32-bit registers S_0 to S_{15} . The registers S_0 , S_2 and S_{11} are used for the feedback operation, whereas the FSM needs the contents of S_5 and S_{15} . In the latter register, the new value from the feedback part is stored each clock cycle. In addition to five XOR and two shift operations in the feedback part, there are two functions, MUL_α and DIV_α , that map the lower byte (DIV_α) and the upper byte (MUL_α) of the respective register to a 32-bit output value. In the FSM, three 32-bit registers R_1 to R_3 are connected by S-Box substitutions. Here, the S-Box S_1 is based on four 8-bit Rijndael S-Boxes S_R , whereas S_2 uses four 8-bit S-Boxes S_Q . Furthermore, two XOR operations and two adds are employed in this part.

The SNOW 3G comprises two modes of operation: the initialization mode and the keystream mode. During the initialization mode, the LFSR and the FSM are clocked 32 times. The output of the FSM is thereby adopted in the feedback calculation. Afterwards, a 32-bit keystream block is produced each clock cycle by an XOR operation of the FSM output and the LFSR register S_0 , whereas the first FSM output word is discarded.

In the 128-EEA1 algorithm, the initial value of the LFSR is given by XOR combinations [5] of the cipher key, defined by the LTE parameter KEY, and another 128-bit variable [6] that is composed of the LTE parameters COUNT, BEARER and DIRECTION (see table I). After initialization, the SNOW 3G generates $j = \text{LENGTH}/32$ 32-bit keystream blocks.

B. The 128-EEA2 Algorithm with AES

The symmetric block cipher AES is specified by the National Institute of Standards and Technology (NIST) as a successor of the Data Encryption Standard (DES) [7]. Based on the Rijndael algorithm, the AES in general can be used with cipher key lengths of 128, 192 and 256 bits, respectively. In LTE, a key length of 128 bits is defined by the 3GPP. Fig. 2 illustrates the structure of the AES. The registers STATE and KEY contain 128-bit values organized in 4x4 byte arrays. After $N_r = 11$ round transformations, the final

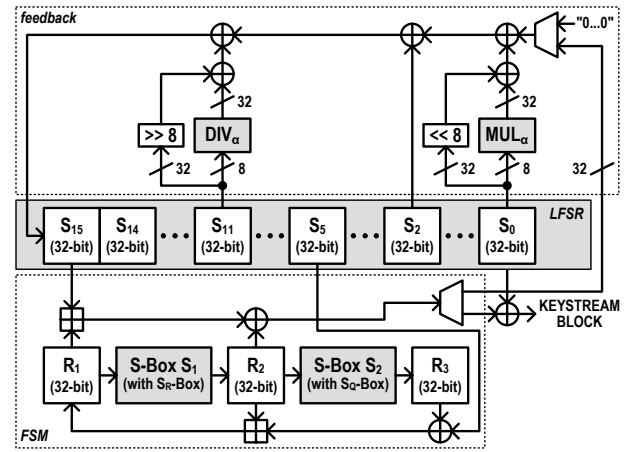


Fig. 1. Structure of the 32-bit SNOW 3G algorithm used in the 128-EEA1 scheme. Keystream generation is based on Rijndael's S_R -Box and S-Box S_Q .

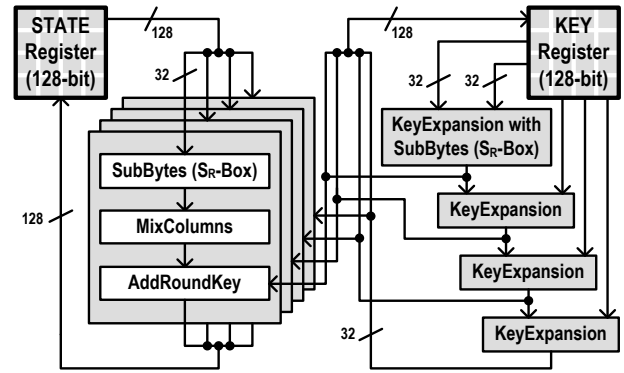


Fig. 2. Structure of the 128-bit AES algorithm used in the 128-EEA2 scheme. It is based on Rijndael's S_R -Box. The *ShiftRows* function can be directly realized in the read and write accesses of the STATE register and is therefore not depicted.

data is available in the STATE register. In order to perform the round transformations, the functions *SubBytes*, *ShiftRows*, *MixColumns* and *AddRoundKey* are adopted. Since *ShiftRows* denotes a periodic shift of the STATE register rows, it can be directly realized in the read and write accesses of the STATE register and is therefore not depicted in Fig. 2. The function *SubBytes* executes nonlinear substitutions with the Rijndael S-Box S_R , while the columns of the STATE register are modulo multiplied in Rijndael's Galois field by a given matrix in *MixColumns*. Finally, the result of one round transformation is given by an XOR operation with the output of the *KeyExpansion* block, that also utilizes the Rijndael S-Box S_R , in the *AddRoundKey* function. It should be noted that in the first round only the *AddRoundKey* function is carried out, whereas in the last round *MixColumns* is not included.

The structure of the 128-EEA2 algorithm is shown in Fig. 3. In LTE, the Counter (CTR) mode of operation, recommended by NIST [8], is defined for the AES. In the CTR mode, the initial data for the STATE register is determined by a 128-bit counter block T_k (with $k = 1, 2, \dots, i$), where T_k is calculated from counter block T_{k-1} applying the standard

Parameter	Size (bits)	Description
KEY	128	Cipher key
COUNT	32	Counter value*
BEARER	5	Radio bearer identity
DIRECTION	1	Transmission direction (0 for uplink, 1 for downlink)
LENGTH	16	Number of bits to be processed

* composed of the hyper frame number (HFN) and the sequence number (SN)

TABLE I
PARAMETERS USED FOR CIPHERING AND DECIPHERING IN LTE [2].

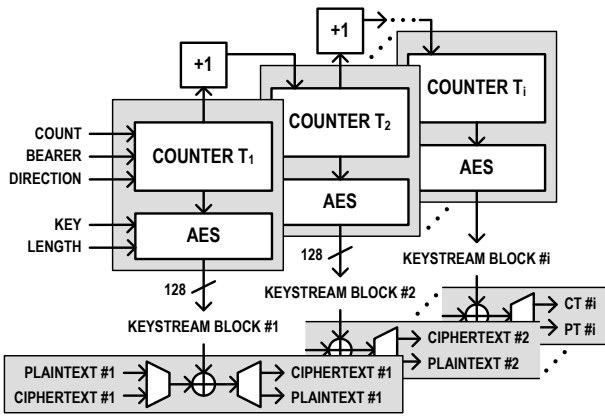


Fig. 3. Structure of the 128-EEA2 ciphering scheme applying the 128-bit AES algorithm in Counter (CTR) mode with incremented counter blocks T_k .

incrementing function defined in [8]. In the initial counter block T_1 the most significant bits consist of the parameters COUNT, BEARER and DIRECTION, whereas the other bits are set to zero. The AES generates $i = \text{LENGTH}/128$ 128-bit keystream blocks that are taken from the final STATE register content after the round transformations.

III. HARDWARE CONSIDERATIONS FOR THE LTE CIPHERING SCHEMES

The AES algorithm can be efficiently implemented in both hardware and software. Although AES optimizations e.g. for 32-bit processors exist [9], software solutions of course will never be as fast as its hardware counterparts. Furthermore, from a security point of view, a hardware module is much better isolated from attacks compared to a software implementation. For these reasons a large number of FPGA and ASIC implementations for the AES algorithm have been published. These implementations focus either on low-power, low-cost devices [10] for systems with lower data rates or on high data throughputs. A good overview to high performance solutions can be found in [11].

From a power perspective, the substitutions with Rijndael's S_R -Box are identified to consume 75% of the overall energy budget [12]. Generally, the table look-up in the S_R -Box is realized either as memory or based on combinational logic. While memories are expensive in terms of power consumption and area effort, S_R -Box architectures using combinational logic offer power and area efficient solutions. A comprehensive study on hardware characteristics for different S_R -Box approaches can be found in [13]. In that work, two architectures show promising properties: the solution from Canright [14] with the lowest gate count at a moderate power level and the approach by Bertoni [15] with the lowest power budget at a clearly increased area effort. While Canright proposes a decomposition of the Galois field inversion into subfields, Bertoni uses a one-hot encoder and decoder with a suitable switching block in between.

Regarding the data throughput, a straightforward way is a variation of the data path width [16]. Intuitively, the AES can

be implemented with a data path width of 32, 64 or 128 bits (see Fig. 2), since the *MixColumns* function processes one 32-bit column a time. Furthermore, solutions with an 8-bit data path exist [10] that are characterized by a very compact architecture. On the other hand, these implementations can hardly deliver data throughputs as needed in (beyond) LTE systems. Therefore, 8-bit architectures are not considered in this work. Applying a 32-bit data path, $4 \cdot N_r = 44$ rounds (and thus clock cycles) have to be carried out in order to calculate a 128-bit keystream block, while four 8-bit S_R -Boxes are needed for both, the *SubBytes* and the *KeyExpansion* function. With a 64-bit and 128-bit data path, the processing time is reduced to $2 \cdot N_r = 22$ and $N_r = 11$ rounds, respectively. As a consequence, the hardware effort is clearly increased. Especially the number of S_R -Boxes is doubled (64-bit) or even quadrupled (128-bit) for the *SubBytes* function. On the other hand, the number of S_R -Boxes for the *KeyExpansion* block is unchanged, because they are only used in the processing of the first KEY register column.

Another implementation is given by the CTR mode of the AES in the 128-EEA2. It allows for the use of more than one AES core, since the input of the AES only depends on the counter block values that can be calculated in advance (see section II-B). With two AES engines in parallel, the data throughput can be doubled compared to a single core solution or the clock frequency can be halved, while the data throughput remains the same. On the other hand, the area effort is also doubled. In general, even more than two AES cores can be adopted in parallel to further increase the data throughput in (beyond) LTE terminals, whereas this work focuses on a dual-AES architecture.

Compared to the AES, only few realizations are known for the SNOW 3G. In [17], a high performance ASIC implementation is presented, where both S -Boxes, S_R and S_Q , are synthesized to combinational logic from look-up tables. On the other hand, it is obvious that for a power and area efficient design, the approaches regarding the S_R -Box with AES can also be adopted for the S_R -Box of the SNOW 3G. Furthermore, the solution by Bertoni is a general approach for cryptographic S -Boxes and can therefore be utilized for the S_Q -Box as well.

The mapping blocks MUL_α and DIV_α (see Fig. 1) can be either implemented as a logic function or synthesized to combinational logic from look-up tables given in [5]. The logic function consists of four recursive calculations for each mapping block with a maximum loop range of 245. Due to this high computational effort, better synthesis results and thus more compact circuits are obtained from look-up tables.

IV. TIMING REQUIREMENTS FROM SYSTEM INVESTIGATIONS

The development of dedicated hardware for LTE protocol stack acceleration is based on results from system platform investigations. Fig. 4 shows the VSP of a (beyond) LTE mobile phone platform [18]. It is generated and analyzed with tools from VaST Systems Technology Corporation [19] that allow for cycle accurate simulations.

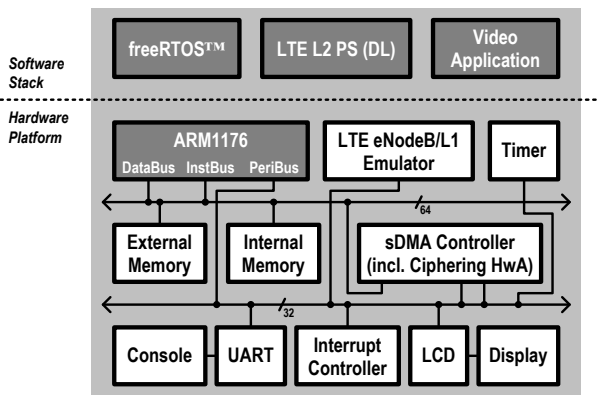


Fig. 4. Virtual system prototype of the LTE mobile phone platform running an LTE L2 protocol stack model on top of emulated, ARM based hardware.

The hardware platform and the software stack that runs on top, as well as profiling results for the timing requirements are described in the following subsections.

A. Hardware Platform

The hardware platform contains an ARM1176 processor that provides 32-bit and 64-bit interfaces to its peripherals. The internal memory is used for platform initialization, whereas the external memory is adopted during software stack execution. It has higher read and write latencies to be conform with state-of-the-art mobile phone platforms. The LTE eNodeB/L1 peripheral is a customizable emulator for the base station and layer 1 (L1) processing that generates transport blocks from a video file. Furthermore, settings for transmission conditions like the data rate are user-defined and can therefore be configured for investigations even beyond LTE. The smart Direct Memory Access (sDMA) controller is a C based peripheral model that enables on-the-fly hardware acceleration for the header processing in the MAC (Media Access Control), the RLC (Radio Link Control) and the PDCP sublayer as well as for the deciphering of user data. With this approach, a performance speedup of almost 50% is achieved compared to platforms with a conventional DMA engine and separate hardware accelerators [18]. The processing time of the decryption unit inside the sDMA controller is determined by the processing time per byte multiplied by the data length. Both values can be set by parameters. This concept allows for platform investigations that are independent of the implemented ciphering algorithm. After data processing in the protocol stack, the video frames are displayed using the LCD controller.

B. Software Stack

The software stack model implements the most time critical part for the execution of the L2 protocol stack for the LTE data plane in downlink (DL) direction [18], compliant to the 3GPP Rel.8 specifications. It consists of the MAC, the RLC and the PDCP sublayer. Mechanisms for the interaction with the on-the-fly hardware accelerator units inside the sDMA controller as well as the initial setup of their parameters are

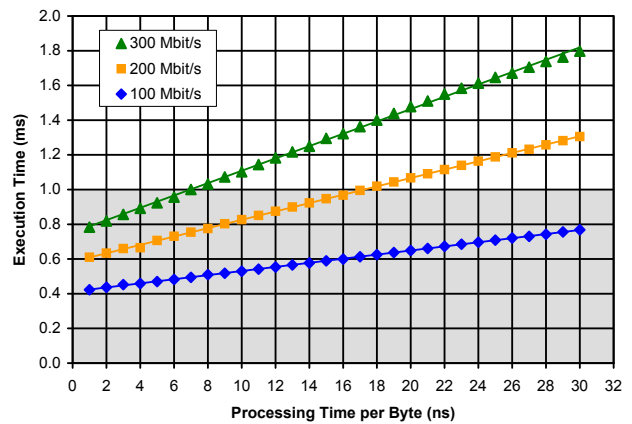


Fig. 5. VSP profiling results of the LTE L2 protocol stack model showing the ciphering timing requirements at different data rates. The execution time of the software stack must be smaller than 1 ms (grey shaded region).

realized in the model. Furthermore, the protocol stack model is partitioned into threads with different priorities using the open source real-time operating system freeRTOS™, while the video application processes the video frames for displaying.

C. Profiling Results

In order to get the timing requirements for the 128-EEA1 and 128-EEA2 hardware implementations, the execution time of the protocol stack model is measured for different processing times per byte of the deciphering unit (see Fig. 5). The hardware platform is thereby configured with a CPU clock frequency of 450 MHz and bus clock frequency of 200 MHz. The cache sizes are set to 32 kB for data and instructions, respectively, according to the results in [18]. To account for an LTE DL data rate of 100 Mbit/s, transport blocks of 100 kbits are generated by the eNodeB/L1 emulator peripheral within the transmission time interval (TTI) of 1 ms. For investigations beyond LTE, the data rate is further increased to 200 and 300 Mbit/s. Fig. 5 shows the increase of the execution time with higher processing times per byte and higher data rates. It should be noted that the execution time has to be clearly below the TTI of 1 ms, since the demand for L2 uplink and higher layer processing is not considered in the platform.

V. RESULTS OF HARDWARE IMPLEMENTATIONS

The hardware characteristics for both ciphering algorithms are obtained from VHDL model synthesis with Synopsys' DesignVision™ Version B-2008.09. Therefore, we use a standard cell library of Faraday's 90 nm CMOS technology with a core power supply of 1.0 V. Furthermore, activity data from netlist simulations are adopted for power analysis in order to get realistic power values. A randomly generated ciphertext of 1000 bytes is therefore decrypted within an absolute time interval of 80 μs during simulation. In the illustrations of the area efforts, the values are given in thousands of gate equivalents (kGE). The number of gate equivalents is calculated from the total area divided by the size of a 2-input AND (5 μm²) that is taken from the technology library. For the purpose of

Algorithm	Data Path Width	Keystream Block Size	Cycles per Keystream	Cycles per Byte
SNOW 3G	32-bit	32 bits	1*	0.28*
AES	32-bit	128 bits	45**	2.81
	64-bit	128 bits	23**	1.44
	128-bit	128 bits	12**	0.75

* additional 32 clock cycles needed for the initialization phase in SNOW 3G
** including one additional clock cycle to initialize the STATE and KEY registers

TABLE II
PROCESSING TIMES (IN CLOCK CYCLES) OF THE INVESTIGATED ARCHITECTURES FOR 128-EEA1 (SNOW 3G) AND 128-EEA2 (AES).

comparison, the results are depicted over the processing time per byte, whereas the clock frequencies are derived from this parameter according to the different architectures (see table II).

Fig. 6 and Fig. 7 show the power consumption and the area effort, respectively, for the 128-EEA1 algorithm with SNOW 3G. Different implementations for the S_R -Boxes and S_Q (see section III) are thereby compared to each other, whereas MUL_α and DIV_α (see Fig. 1) are synthesized to combinational logic from look-up tables. As expected, Bertoni's approach leads to the most power efficient solution, while architectures with Canright's S_R -Boxes have the smallest area. In general, all power values are decreasing with a higher process-

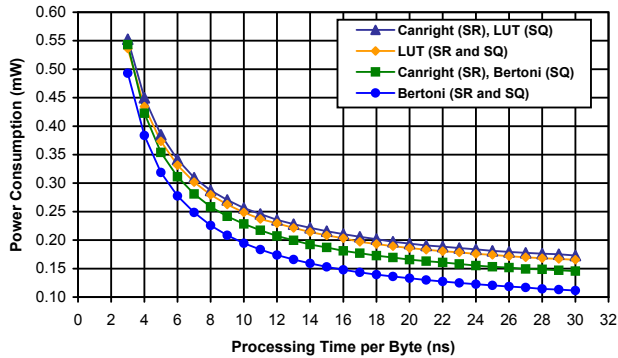


Fig. 6. Power consumption of the 128-EEA1 (with SNOW 3G) for different S_R -Box (SR) and S_Q -Box (SQ) approaches compared to a look-up table (LUT) implementation.

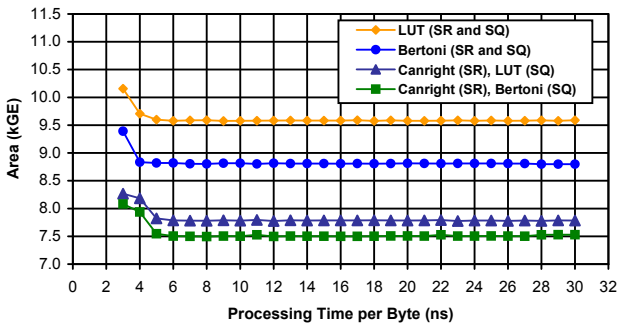


Fig. 7. Area effort of the 128-EEA1 (with SNOW 3G) for different S_R -Box (SR) and S_Q -Box (SQ) approaches compared to a look-up table (LUT) implementation.

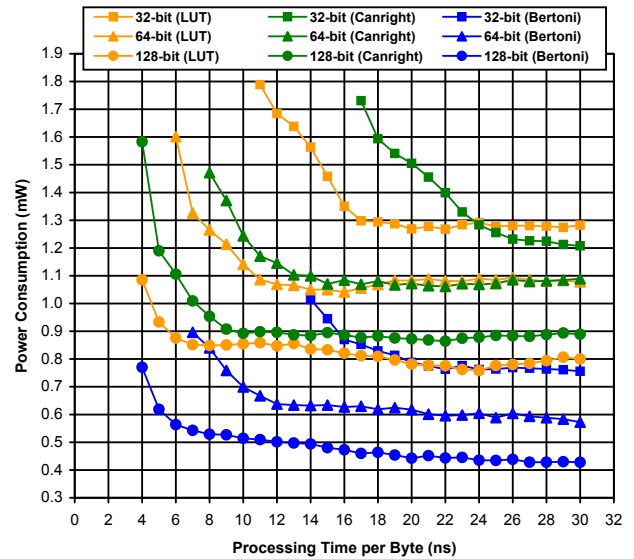


Fig. 8. Power consumption of the 128-EEA2 (with AES) for different data path widths and various S_R -Box approaches compared to a look-up table (LUT) implementation.

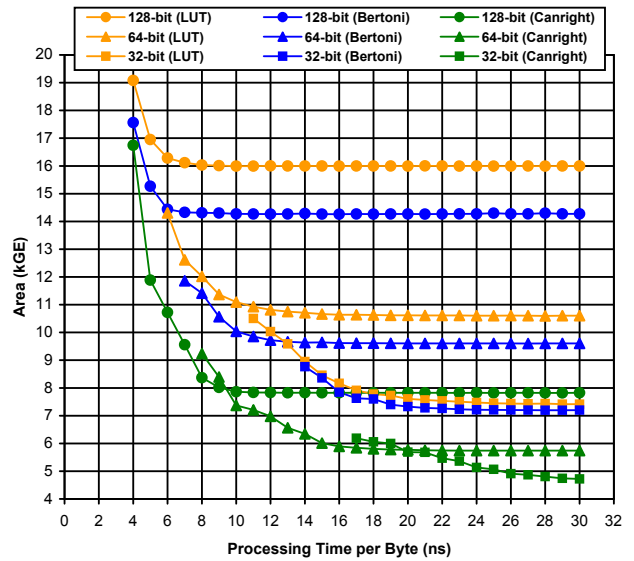


Fig. 9. Area effort of the 128-EEA2 (with AES) for different data path widths and various S_R -Box approaches compared to a look-up table (LUT) implementation.

ing time per byte and thus smaller clock frequencies. On the other hand, the area effort of each implementation is almost constant, except for processing times per byte below 5 ns. Due to higher clock frequencies, the circuit effort is increased in order to fulfill the critical path requirements.

For the implementations of the 128-EEA2 algorithm with AES, the same properties with regard to Bertoni's and Canright's solutions can be observed (see Fig. 8 and Fig. 9). Additionally, different data path widths are investigated. While a higher data path width at a constant processing time per byte results in a reduced clock frequency and therefore in a decreased power consumption, it leads to a clearly increased

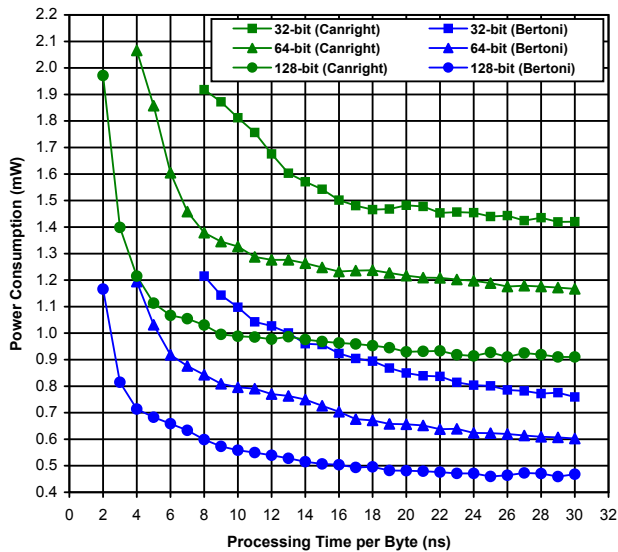


Fig. 10. Power consumption of 128-EEA2 with two AES engines.

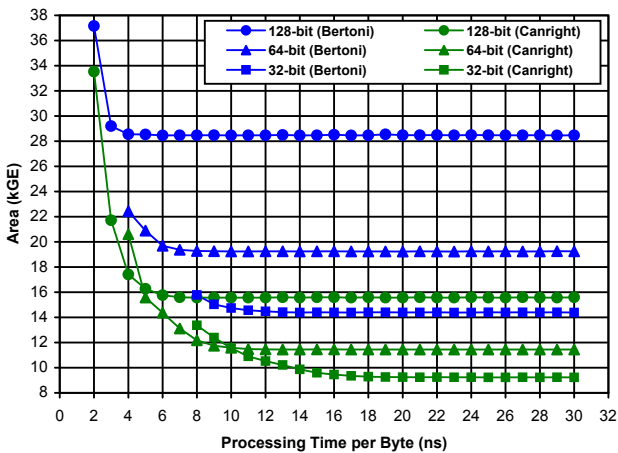


Fig. 11. Area effort of 128-EEA2 with two AES engines.

area demand on the other hand. Furthermore, the diagrams show that the minimum processing time per byte is increased with smaller data path widths, since their clock frequencies are doubled or even quadrupled. Therefore, the maximum clock frequency is reached at higher processing times per byte.

Fig. 10 and Fig. 11 depict the hardware characteristics for the dual-AES approach in the 128-EEA2. Once again, the same relation between the architectural variants can be monitored. The absolute area values however are clearly increased (almost doubled) compared to the single AES solutions in Fig. 9, while the power numbers are only slightly higher due to halved clock frequencies. Additionally, two AES cores in parallel enable higher data throughputs and thus smaller processing times per byte.

In order to compare the hardware efforts for the 128-EEA1 and 128-EEA2 implementations, the power-area product is illustrated in Fig. 12. A processing time per byte of 4, 14 and 24 ns is thereby considered to account exemplarily

for the different requirements for the investigated data rates in Fig. 5. Obviously, the 128-EEA1 algorithm shows better properties than its 128-EEA2 counterpart. Although the area effort is comparable to the 32-bit AES architectures, the power consumption is lower due to a reduced clock frequency. While the SNOW 3G generates a 32-bit keystream block each clock cycle, the AES needs up to 45 clock cycles for a 128-bit keystream block (see table II). Within the 128-EEA1, architectures with Bertoni's S-Box approach have the lowest power-area product. For the 128-EEA2 however, the LUT approach is clearly outperformed by Bertoni and Canright at 14 and 24 ns. Here, both solutions offer similar power-area products, except for the 32-bit architecture with Canright that is not applicable at 4 and 14 ns, because the critical paths in these architectures exceed the corresponding clock periods. At a processing time per byte of 4 ns, the 128-bit architectures are the only choice, since the 32-bit and 64-bit solutions cannot fulfill the computational demands. Regarding the S-Box variants, Bertoni once again comes up with the best solution: The smallest power-area product of comparable data path widths is given at 24 ns (32-bit), 14 ns (64-bit) and 4 ns (128-bit). Due to the lowest power consumption of all investigated solutions (see Fig. 8), Bertoni's S-Box combined with a 128-bit data path in the AES architecture is the first choice for ciphering with the 128-EEA2 in LTE terminals.

For the approach with two AES cores, the power-area products are significantly increased, mainly due to area efforts that are almost doubled compared to their single-core counterparts. But multi-AES architectures are a reasonable option for mobile devices far beyond LTE (like LTE-Advanced with peak data rates up to 1 Gbit/s [20]), since parallel engines in general allow for higher data throughputs.

VI. CONCLUSION

In this paper we analyze hardware implementations for user confidentiality in (beyond) LTE terminals in order to identify suitable architectures that fulfill the required performance while keeping the power consumption and silicon effort at a reasonable level. The 3rd Generation Partnership Project (3GPP) specifies two algorithms for data plane ciphering in the protocol layer 2, namely 128-EEA1 and 128-EEA2. While the 128-EEA1 is based on the SNOW 3G, the 128-EEA2 adopts the Advanced Encryption Standard (AES). Both algorithms contain cryptographic substitution boxes (S-Boxes) that are known for consuming most of the power and area budget. Therefore, different S-Box architectures are explored for the 128-EEA1 and 128-EEA2 as well as various numbers of S-Box instances in the 128-EEA2 by changing the AES data path width. The hardware characteristics are obtained from VHDL model synthesis based on a 90 nm CMOS technology library from Faraday. The timing requirements are thereby given by execution time measurements of the ciphering procedure in a virtual mobile phone platform comprising an LTE protocol stack model that runs on top of emulated, ARM based hardware. By the evaluation of power-area products, the S-Box approach that uses a one-hot decoding and encoding with a

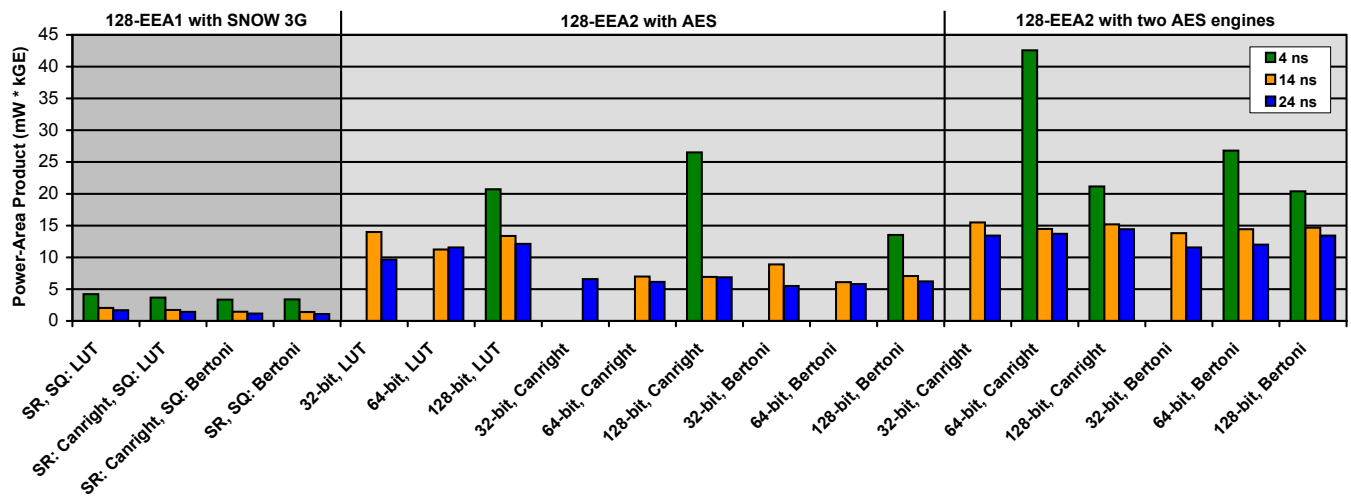


Fig. 12. Power-area product of the investigated ciphering modules for different timing requirements (and thus data rates) in terms of processing time per byte. Two S_R -Box (SR) and S_Q -Box (SQ) approaches are compared to look-up table (LUT) implementations. Additionally, different data path widths are analyzed in the AES algorithm. Some implementations are not applicable at 4 and 14 ns and therefore, the corresponding bars (green and orange) are missing.

suitable switching block in between turns out to be the best solution for ciphering with 128-EEA1 and 128-EEA2. For the latter algorithm, the AES architecture with a 128-bit data path is the most promising solution for LTE terminals, whereas the employment of two or more AES cores is an option for data rates far beyond the LTE requirements.

ACKNOWLEDGMENT

The authors acknowledge the excellent cooperation with all project partners within the EASY-C project and the support by the German Federal Ministry of Science and Education (BMBF). Further information is available on the project website: <http://www.easy-c.de>.

REFERENCES

- [1] D. Szczesny, A. Showk, S. Hessel, U. Hildebrand, V. Frascolla and A. Bilgic, "Performance Analysis of LTE Protocol Processing on an ARM based Mobile Platform", in preparation for *7th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC-09)*, Vancouver, Canada, Aug. 2009.
- [2] *Evolved Universal Terrestrial Radio Access (E-UTRA): Packet Data Convergence Protocol (PDCP) specification*, 3GPP Std. TS 36.323, Rev. 8.4.0, Dec. 2008.
- [3] *3GPP System Architecture Evolution (SAE): Security Architecture*, 3GPP Std. TS 33.401, Rev. 8.2.1, Dec. 2008.
- [4] *Technical Specification Group Services and System Aspects; 3G Security; Security Architecture*, 3GPP Std. TS 33.102, Rev. 8.1.0, Dec. 2008.
- [5] *Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW 3G Specification*, ETSI/SAGE Specification, Version 1.1, Sep. 2006.
- [6] *Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 1: UEA2 and UIA2 Specification*, ETSI/SAGE Specification, Version 1.1, Sep. 2006.
- [7] National Institute of Standards and Technology (NIST), "FIPS Publication 197: Advanced Encryption Standard (AES)", Nov. 2001, available: <http://csrc.nist.gov/publications/PubsFIPS.html>
- [8] National Institute of Standards and Technology (NIST), "NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation - Methods and Techniques", Dec. 2001, available: <http://csrc.nist.gov/publications/PubsSPs.html>.
- [9] G. Bertoni, L. Breveglieri, P. Fragneto, M. Macchetti and S. Marchesin, "Efficient Software Implementation of AES on 32-Bit Platforms", in *4th International Workshop on Cryptographic Hardware and Embedded System (CHES 2002)*, Revised Papers, LNCS Vol. 2523, pp. 159-171, 2002.
- [10] M. Feldhofer, J. Wolkerstorfer and V. Rijmen, "AES Implementation on a Grain of Sand", in *IEEE Proceedings on Information Security*, Vol. 152, Issue 1, pp. 13-20, Oct. 2005.
- [11] M. Alam, S. Ghosh, D.R. Chowdhury and I. Sengupta, "Single Chip Encryptor/Decryptor Core Implementation of AES Algorithm", in *21st International Conference on VLSI Design (VLSID 2008)*, Hyderabad, India, pp. 693-698, Jan. 2008.
- [12] S. Morioka and A. Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design", in *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002)*, Revised Papers, LNCS Vol. 2523, pp. 172-186, 2002.
- [13] S. Tillich, M. Feldhofer, T. Popp and J. Großschädl, "Area, Delay, and Power Characteristics of Standard-Cell Implementations of the AES S-Box", in *Journal of Signal Processing Systems*, Volume 50, Issue 2, pp. 251-261, Feb. 2008.
- [14] D. Canright, "A very compact S-Box for AES", in *7th International Workshop on Cryptographic Hardware and Embedded System (CHES 2005)*, LNCS Vol. 3659, pp. 441-455, 2005.
- [15] G. Bertoni, M. Macchetti, L. Negri and P. Fragneto, "Power-efficient ASIC Synthesis of Cryptographic Sboxes", in *14th ACM Great Lakes Symposium on VLSI (GLSVLSI 2004)*, Boston, USA, pp. 277-281, Apr. 2004.
- [16] A. Satoh, S. Morioka, K. Takano and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization", in *7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 2001)*, LNCS Vol. 2248, pp. 239-254, 2001.
- [17] P. Kitsos, G. Selimis and O. Koufopavlou, "A High Performance ASIC Implementation of the SNOW 3G Stream Cipher", in *16th International Conference on Very Large Scale Integration (VLSI-Soc 2008)*, Rhodes Island, Greece, Oct. 2008.
- [18] D. Szczesny, S. Hessel, F. Bruns and A. Bilgic, "On-the-fly Hardware Acceleration for Protocol Stack Processing in Next Generation Mobile Devices", in *5th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS 2009)*, Grenoble, France, Oct. 2009.
- [19] The *VaST Systems Technology Corporation* website, available: <http://www.vastsystems.com>
- [20] *Requirements for further advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced)*, 3GPP Std. TS 36.913, Rev. 8.0.1, Mar. 2009.