International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34647-34651

FPGA Implementation and Performance Analysis of Stream Ciphers for Cryptographic Applications

[1]K. Kalaiselvi, [2]Dr. H. Mangalam,
[1]Assistant Professor, Department of ECE, Hindusthan College of Engineering and Technology, Coimbatore.
[2]Professor and Head, Department of ECE, Sri Krishna College of Engineering and Technology, Coimbatore.

Abstract- The rapid increase in computer networks and communication system usage needs information encryption to ensure security. Many ciphers such as ZUC, Snow3g, Grain V1, Mickey V2, Trivium, E0 and encryption algorithm have been developed over the years. The encryption of information in any system normally increases the system complexity and it may lead to degradation in performance. To overcome these issues, VLSI implementation of stream ciphers and encryption techniques have been proposed in many works. In this paper, fieldprogrammable gate array (FPGA) implementation of three stream ciphersGrain V1, Mickey V2, Trivium are presented which are popular in wireless communication protocols. Implementation results are compared in termsof throughput, consumed area, critical path delay and power consumption. The designs were simulated using VHDL language and implemented on a Xilinx virtex FPGA device. All the ciphers consumed approximately 1% of the resources of the FPGA device. The proposed implementations are efficient while comparing with existing hardware implementation and highest throughput can be achieved in Grain V1 cipher.

Keywords- Stream cipher, Encryption, very large scale integration (VLSI) implementation, power consumption, field programmable gate array (FPGA)

1. Introduction

In the present eraof information processing through computers and the threat ofsecret information access, encryption of the messages in various forms has become inevitable. Over the recent years, internet has been extensively used for bank account maintenance, transaction of money, business through online shopping and transactions. Cryptography has an important role in providing security for data transmission. It is the best method for data protection against passive and active fraud. There are mainly two types of encryption algorithms, a private key and public key. Private key is also called as symmetric key which utilizes a single key for performing both encryption and decryption. Private key is also known as asymmetric key where separate key is used for encryption and decryption [1].

In the software performance analysis of crypto systems, speed is the important metric to be considered [2]. However, hardware performance focuses on three important aspects, they are area, throughput and power. A hardware implementation and testingprocedure needs to concentrateon five dimensions based on the application. They are compactness, throughput, power consumption,scalability and simplicity [3].Generally private key algorithms are less complex comparing with public key algorithms. Hence the private algorithms are preferred in VLSI implementation, where gates and flip-flops are used. The throughput of the private key algorithms can be improved due to its simple circuit nature.

Cryptographic systems are designed based on either stream ciphers or block ciphers. Stream cipher comprises of key stream, plain text and cipher text. The cipher text in stream ciphers is obtained by performing xor operation between plain text and key stream. The transformation of bits is performed on individual data bits in stream ciphers whereas group of bits are transformed in block ciphers. Though the buffering capacity in stream ciphers is limited, the stream ciphers are widely preferred in crypto systems because of its inherent implementation advantage [4]. Each secret key can be obtainedfrom the key stream sequence in the receiver by performing the same xor operation performed in the transmitter. Based on the above operation, the message sequence can be easily retrieved in the receiver end [5]. As discussed above, the VLSI implementation of stream ciphers are important in cryptosystem design. In addition, hardware complexity, throughput and power issues need to be analyzed and addressed. In this paper, we propose the fieldprogrammable gate array (FPGA) implementation of threeimportant stream ciphersGrain V1, Mickey V2 and Trivium. The proposed implementation is analyzed in term of area, throughput and power. The hardware implementations of all the stream ciphers are targeted toward the Xilinx Spartan 3 family of FPGAs.

2. Encryption techniques and Stream ciphers

Encryption techniques play an important role in cryptosystem design and implementation. Encryption process coverts the plain text (original data) into cipher text. On the other hand, decryption retrieves the original data from the cipher text through a key stream which is known only by the authenticated user. Figure 1 shows the encryption/decryption procedure.

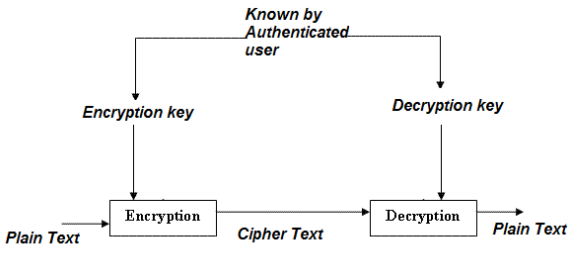


Fig. 1 Encryption/Decryption procedure

Two major categories in encryption algorithm are symmetric and asymmetric. Symmetricencryption algorithms arefurtherclassified into block ciphersand stream ciphers algorithms [6]. Block ciphers deal withthe input data in the form of *N*-bit blocks and produce*N*-bit blocks of encrypted or decrypteddata by combining secret key with the input data. Figure2 (a) shows the block cipher encryption procedure for converting N-bit plain text into cipher text. Cipher text can be obtained by mathematical equations, manipulations and some of algorithmic procedure.However, stream ciphers perform the encryption serially by combining a stream of key bit, referred to as keystream. Figure2 (b) shows the stream cipher encryption procedure for converting plain text into cipher text. Keystream is generated based on the key provided at the sender side and initialization vector (IV). Keystream allows the data to encrypt or decrypt based on the requirement.One of the major advantages of stream ciphers is that they do not suffer fromthe error propagation which generally happens in block ciphers [7]. Error propagation can be avoided in the stream ciphers using individual bit encryption and decryption. Anotherbenefit is that it is easily possible to implement in both softwareand hardware while comparing with block ciphers. Hence, stream ciphersare widely preferred for several telecommunication standards suchas Global System for Mobile (GSM) [8], Long Term Evolution (LTE)[9] and Bluetooth [10].

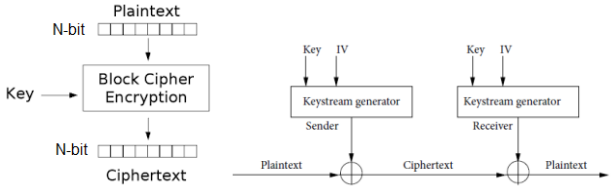


Fig. 2 (a) Block cipher encryption (b) Stream cipher encryption

Six stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium, E0 are the popular choice in the telecommunication applications. Three of them (ZUC, Snow3g and E0) havebeen used in the security related protocols and secure communication. Specifically,ZUC algorithm is part of the 128-EEA3 and the 128-EIA3 protocolsused for confidentiality and integrity, respectively, in thewireless transmissions in LTE. This set of protocols has been developedby 3GPP [11] and GSM association. Snow3g is another stream cipher that is mainly used for security protocols for maintaining confidentialityand integrity. Because of its secure nature, Universal MobileTelecommunication System (UMTS) networks use this stream cipher [9]. While designing a cryptographic system, it is necessaryto address various aspects. These include speed, security and simplicity.While analyzing several ciphers, it is noted that the software implementation can be very simple, but the hardware implementation might be quite complex.

The rapid development in portable devices with miniaturized size, it is important to develop cryptographic algorithms with low hardwarecomplexity. An RFID tag is a typical example of a product where the amount ofmemory and power is very limited. Microchips in this tags are capable of transmittingan identifying sequence upon a request from a reader. Forging an RFID may result in disastrous consequences,hence there is a need for cryptographic primitives implemented in these tags [12], [13].In this work, three stream ciphers Mickey, Grain and Trivium are considered for FPGA implementation.

i) Mickey V2 stream cipher

Mickey V2 is a bit-oriented stream cipher for performing encryption. It comprises of 80-bit inputs, the Key and the Initial Vector (IV) and the Input bit[14]. Ithas shift register R and the register S for performing logical operations. XOR gate is placed in between the register cells.These XOR gates are driven by two control bits, Control bit R andControl bit S for register R and register S, respectively. Then bothcontrol bits are fed back to registers.In order for the system to be initialized, first the two registersare set to zero and then they are clocked for 100 cycles. Then thecipher outputs the Keystream bits. The pseudo code for Mickey V2 operations is given below

Step1. Control bit R = $s_{34} \oplus r_{67}$;

Step2. Control bit S = $s_{67} \oplus r_{63}$;

Step3. If in Initialization state, then Input Register R = Input bit $\oplus$ s_{60},

else Input Register R = Input bit;

Step4. Input Register S = Input bit;

Step5. Clock the two registers

Step6.Keystream = $r_0 \oplus s_0$.

ii) Grain V1 stream cipher

Grain V1 has two inputs, an 80-bit Key and an 64-bit InitialVector (IV)[15]. Grain V1 is a bitorientedstream cipher similar to Mickey cipher. It has three main logical parts, an 80-bitLinear Feedback Shift Register (LFSR), an 80-bit Nonlinear FeedbackShift Register (NFSR) and an output function H. The content of theNFSR is denoted by b_i and the content of the LFSR is denoted by s_i.The equations that describe the outputs of the NFSR, the LFSR andthe function H are given below:

NFSR = s0 $\oplus$ b62 $\oplus$ b60 $\oplus$ b52 $\oplus$ b45 $\oplus$ b37 $\oplus$ b33 $\oplus$ b28 $\oplus$ b2 $\oplus$ b14

LFSR = s62 $\oplus$ s51 $\oplus$ s38 $\oplus$ s23 $\oplus$ s13 $\oplus$ s0

The Grain V1 cipher operates in two modes, Key Initializationmode and Stream Generation mode. During the Key Initializationmode the cipher is clocked for 160 times without producing anykey. The output function H is fed back and XO Red with the input, both to the LFSR and to the NFSR. During Stream Generation theoutput of H function is masked with some bits from the NFSR producing the Key stream.

iii) Trivium stream cipher

Trivium is also a bit-oriented stream cipher that has two80-bit inputs, the key, the initial vector (IV) and input bit[16].The two important phases of Trivium stream cipheralgorithm are Initialization and keystream generation. Thebits are managed as a group of 288-bit which is called Internalstate and is denoted by S_i. The initializationphase comprises of the following steps.

For i = 1 to 4 x 288 do

Step1.x1<= S66 $\oplus$ S93

Step2.x2<= S162 $\oplus$ S177

Step3.x3<= S243 $\oplus$ S288

Step4. t1 <= x1 $\oplus$ (S91 AND S92) $\oplus$ S171

Step5. t2 <= x2 $\oplus$ (S175 AND S176) $\oplus$ S264

Step6. t3 <= x3 $\oplus$ (S286 AND S287) $\oplus$ S69

Step7. S1..S93 <= t3 || S1.S92

Step8. S94..S177 <= t1 || S94.S176

Step9. S178..S288 <= t2 || S178.S287

3. FPGA Implementation of Stream ciphers

The algorithm implementations could be software or hardwareoriented. However, because of the continuously growing requirementsfor high speed solutions, the hardware implementation needs to be efficient. To achieve higher levels of secure communicationthe algorithms tend to be more sophisticated. This means thatthose algorithms have also higher demands for processing power.

FPGAs are well suited to the datapath intensive designs encountered in encryption applications.For example, Quicklogic's QuickRAM family with its register and routing rich architecture, and internal RAM, lends itself well to the encryption architecture. The technique of partial product accumulation, which predetermines the multiplied coefficient outputs via the use of adders and shifters, eliminates the need for multipliers. The RAM can be configured as a ROM, to preload the coefficient values, thus saving core logic for implementing perimeter control/support logic,which in turn increases the overall sampling rateAlso, in these implementationsthe full bandwidth utilization could not be achieved.

In this paper, the hardware implementations of the threestream ciphers Mickey, Grain and Trivium are presented. Figure 3 shows the implementation of Grain cipher using xor gates and multiplexer units. The similar procedure could be followed for Mickey and Trivium cipher implementations. The performance metrics are thethroughput, the area consumption and the efficiency for the hardwareimplementation in terms of throughput-to-area ratio. Thesemetrics are basic for comparisons and analysis in hardware.

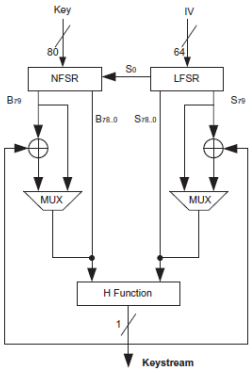


Fig. 3 FPGA Implementation of Grain cipher

For the implementation an FPGA device was used. This solution is ahighly promising alternative because superior performances couldbe achieved. For design's implementations the hardware descriptionlanguage VHDL was used. The software tools used for synthesize,simulation, measuring throughput and area consumption are ISE tool and Modelsim.

4. Results and Discussion

FPGA implementation procedure can be initiated using functional verification. The functional verification is done using simulation. Modelsim has been used for performing simulation. Synthesis, place-and-route, and timing analysis are performed using Xilinx ISE tool. Once the function of individual modules is verified for correctness, they are combined together using synthesis process.Fig. 4 shows the simulation output of Mickey cipher encryption. The design is synthesized in Xilinx Environment. The target device is xc5vlx30 in the family of Virtex5.

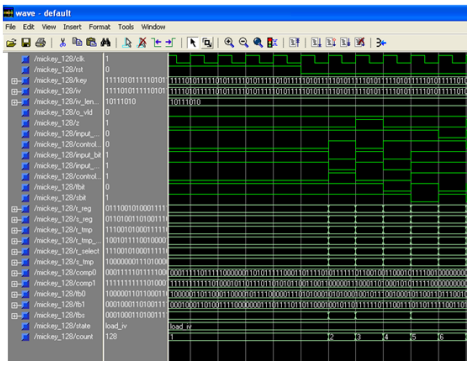
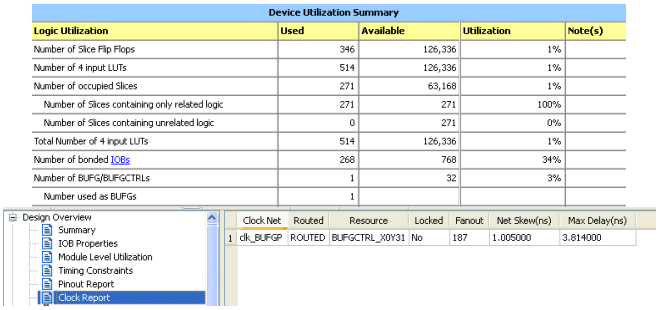


Fig.4 Simulation of Mickey cipher

The synthesis of the chip is performed within the XILINX tool targeting Xilinx Spartan 3 technology (XC3S400-4PQG208C target device). The combination of Modelsim and Xilinx design flow has been used for the entire process.Figure 5 lists the flip flops, LUTs and slices used for the Mickey cipher encryption process. The clock report is included with the critical path delay. The device utilization is less than 1% and the maximum delay is 3.81ns with clock skew of 1.005 ns.



Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	346	126,336	1%	
Number of 4 input LUTs	514	126,336	1%	
Number of occupied Slices	271	63,168	1%	
Number of Slices containing only related logic	271	271	100%	
Number of Slices containing unrelated logic	0	271	0%	
Total Number of 4 input LUTs	514	126,336	1%	
Number of bonded IOBs	268	768	34%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			

Design Overview: Summary, IOB Properties, Module Level Utilization, Timing Constraints, Pinout Report, Clock Report

	Clock Net	Routed	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
1	clk_BUFGP	ROUTED	BUFGCTRL_X0Y31	No	187	1.005000	3.814000

Fig.5 Device utilization and delay report of Mickey cipher

The register transfer logic (RTL) is obtained by synthesizing the VHDL coded design. Figure 6 shows the RTL schematic of Mickey cipher implementation. The timing simulation is additionally performed to verify the functional correctness of the planning

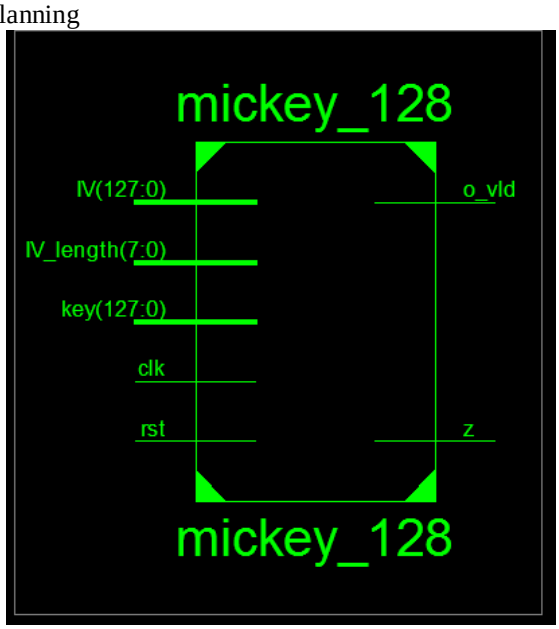


Fig. 6 RTL schematic of Mickey cipher

Fig. 7 shows the simulation output of Grain cipher encryption. Figure 8 lists the flip flops, LUTs and slices used for the Grain cipher encryption process. The clock report is included with the critical path delay. The device utilization is less than 1% and the maximum delay is 3.81ns with clock skew of 1.005 ns.

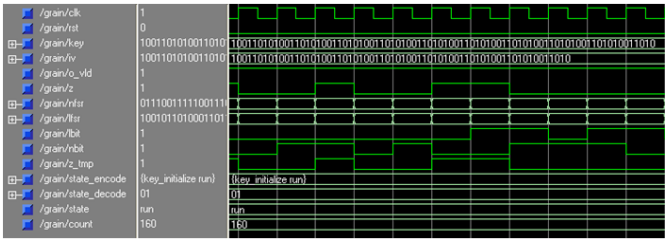
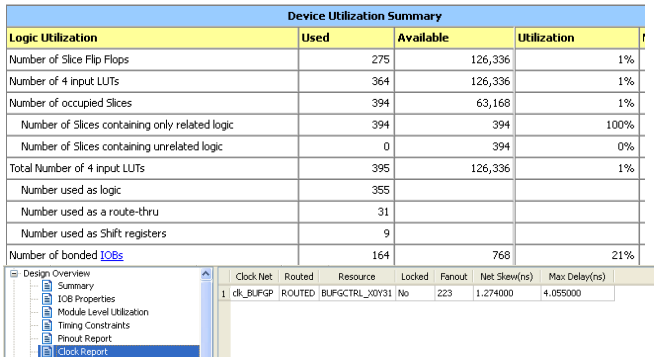


Fig.7 Simulation of Grain cipher



Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	275	126,336	1%
Number of 4 input LUTs	364	126,336	1%
Number of occupied Slices	394	63,168	1%
Number of Slices containing only related logic	394	394	100%
Number of Slices containing unrelated logic	0	394	0%
Total Number of 4 input LUTs	395	126,336	1%
Number used as logic	355		
Number used as a route-thru	31		
Number used as Shift registers	9		
Number of bonded IOBs	164	768	21%

Design Overview: Summary, IOB Properties, Module Level Utilization, Timing Constraints, Pinout Report, Clock Report

	Clock Net	Routed	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
1	clk_BUFGP	ROUTED	BUFGCTRL_X0Y31	No	223	1.274000	4.055000

Fig.8 Device utilization and delay report of Grain cipher

Fig. 9 shows the simulation output of Trivium cipher encryption. Figure 10 lists the flip flops, LUTs and slices used for the Trivium cipher encryption process. The clock report is included with the critical path delay. The device utilization is less than 1% and the maximum delay is 3.81ns with clock skew of 1.005 ns.

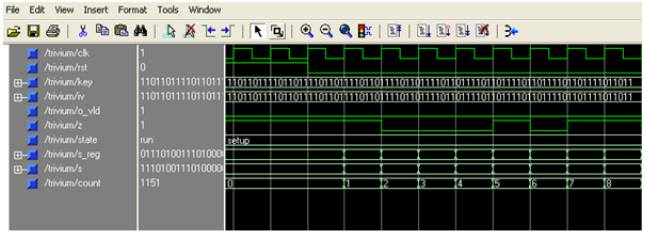
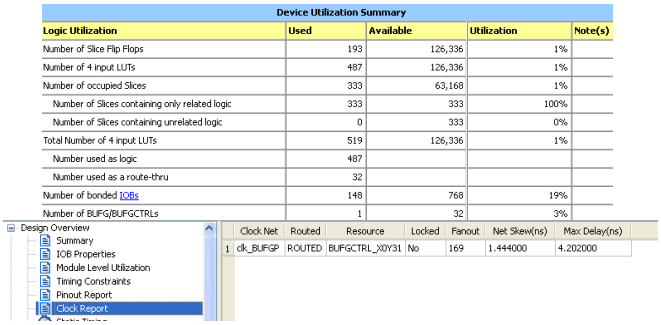


Fig.9Simulation of Trivium cipher



Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	193	126,336	1%	
Number of 4 input LUTs	487	126,336	1%	
Number of occupied Slices	333	63,168	1%	
Number of Slices containing only related logic	333	333	100%	
Number of Slices containing unrelated logic	0	333	0%	
Total Number of 4 input LUTs	519	126,336	1%	
Number used as logic	487			
Number used as a route-thru	32			
Number of bonded IOBs	148	768	19%	
Number of BUFG/BUFGCTRLs	1	32	3%	

Design Overview: Summary, IOB Properties, Module Level Utilization, Timing Constraints, Pinout Report, Clock Report

	Clock Net	Routed	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
1	clk_BUFGP	ROUTED	BUFGCTRL_X0Y31	No	169	1.444000	4.202000

Fig.10Device utilization and delay report of Trivium cipher

Table 1 compares the critical path delay, number of LUTs used and total power consumption of Mickey, Grain and Trivium ciphers. The device utilization and power consumption are less in Grain cipher comparing with other ciphers. However, the minimum critical path delay could be achieved using Mickey cipher implementation.

Table 1 Comparison of throughput and power

Stream cipher	Critical path delay (ns)	No. of 4 input LUTs	Total power(mW)
Mickey	3.81	514	77.92
Grain	4.05	364	63.13
Trivium	4.20	487	71.24

5. Conclusion

In this paper, threepopular stream ciphers are implementedin FPGA hardware. Implementation results are compared in terms of consumed area, delay and power consumption. The designs were simulated using VHDL language and implemented on a Xilinx SpartanXC3S400-4PQG208CFPGA device. All the ciphers consumed approximately 1% of the resources of the FPGA device. The proposed implementations using Grain cipher. The device utilization and power consumption are less in Grain cipher implementation, but the minimum critical path delay could be achieved using Mickey cipher implementation.

References

[1] Menezes A, van Oorschot P, Vanstone S. Handbook of applied cryptography. CRC Press; 1996
[2] L. Batina, S. Kumar, J. Lano, K. Lemke, N. Mentens, C. Paar, B. Preneel, K. Sakiyama and I. Verbauwhede, "Testing Framework for eSTREAM Profile II Candidates", SASC06, available at: www.ecrypt.eu.org/ stream
[3] F. K. Gürkaynak, P. Luethi, "Recommendations for Hardware Evaluation of Cryptographic Algorithms", available at: www.ecrypt.eu.org/stream
[4] Zhang M. Maximum correlation analysis of nonlinear combining functions in stream ciphers. J Cryptol 2000;13(3):301–13.
[5] SarbaniPalit, Bimal K. Some statistical attacks on stream cipher cryptosystems. J Ind Statist Ass 2004;42: 1–34. May.2007
[6] Bruce Schneier, Applied Cryptography – Protocols, Algorithms and Source Codein C, second ed., John Wiley & Sons, New York, 1996.
[7] Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography:Principles and Protocols, first ed., Chapman and Hall, CRC, 2007.
[8] Recommendation GSM 02.09, European Telecommunications StandardsInstitute (ETSI), Security Aspects, 1993.
[9] 3G TS 33 401 V 9.3.1 (2010-04) 3rd Generation Partnership Project; TechnicalGroup Services and System Aspects; 3 GPP System Architecture Evolution(SAE); Security architecture (Release 9).
[10] Specification of the Bluetooth system, vol. 1.1, February 2001. <http://www.bluetooth.org/spec/>.
[11] 3rd Generation Partnership Project (3GPP). <http://www.3gpp.org/>.
[12] P. Ekdahl and T. Johansson.A New Version of the Stream Cipher SNOW.SelectedAreas in Cryptography, SAC 2002, Springer Verlag, LNCS 2595, pp. 47–61, 2002.
[13] P. Hawkes and G. Rose.Primitive Specification for SOBER-128.IACR ePrintArchive, http://eprint.iacr.org/ 2003/81/, 2003.
[14] Martin Hell, Thomas Johansson, Willi Meier, in: Grain: a stream cipher forconstrained environments, International Journal of Wireless and MobileComputing 2 (1) (2007).
[15] Steve Babbage, Matthew Dodd, The Stream Cipher MICKEY-128 2.0, TheeSTREAM Project. <http://www.ecrypt.eu.org/stream/mickey128p2.html>.
[16] Christophe De Canniere, Bart Preneel, Trivium specifications, Thee STREAM Project. <http://www. ecrypt.eu.org/stream/p3ciphers/trivium/triviump3.pdf>.