Testing Framework for eSTREAM Profile II Candidates*

L. Batina[1], S. Kumar[2], J. Lano[1], K. Lemke[2], N. Mentens[1],
C. Paar[2], B. Preneel[1], K. Sakiyama[1] and I. Verbauwhede[1]

[1] Katholieke Universiteit Leuven, ESAT/COSIC,
B-3001 Leuven, Belgium
[2] Horst Görtz Institute for IT Security
Ruhr University Bochum, 44780 Bochum, Germany

Abstract. The aim of eSTREAM Profile II is to identify a small number of stream ciphers that are suitable for low resource circuitry based implementation. Besides algorithmic properties and security evaluation to theoretical attacks, performance evaluation is another important task of eSTREAM that is being considered. In this contribution we summarize and explain our testing framework for eSTREAM Profile II candidates regarding hardware implementations.

Keywords: stream ciphers, hardware implementations, implementation attacks

1 Introduction

The main motivation of the eSTREAM project is to identify stream ciphers that can be used as replacements for AES in both high throughput software based implementations (Profile I) and low resource hardware (circuitry) based implementations (Profile II).

Whereas the approach undertaken for performance testing of Profile I candidates is well known, detailed test plans for Profile II candidates have not been presented, yet. Our contribution encourages an open approach for this framework. This work is produced by the VAMPIRE lab as part of the ECRYPT project.

2 Performance Criteria for Profile II Candidates

* The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT, the European Network of Excellence in Cryptology. KUL researchers are also supported by FWO projects (G.0141.03, G.0450.04), GOA Mefisto 2000/06, GOA Ambiorix 2005/11.

The primary aim of eSTREAM Profile II is to find stream ciphers that require lower resources that an AES implementation in circuitry yielding at least the same throughput as an AES implementation. For evaluating the performance of Profile II candidates we consider the categories

1. Compactness (Area),
2. Performance (Throughput),
3. Power Consumption,
4. Flexibility/Scalability/Pipelining and
5. Simplicity/Completeness/Clarity

Each test category is explained in more detail below.

Our main approach is to consider the possible trade-offs between these categories. Among them, compactness and performance are the most important ones and a trade-off metric for compactness and performance is preferable. We mention also a firm requirement for a low power consumption, which is of crucial importance for wireless applications such as PDAs, mobile phones, RFIDs *etc.*

We especially compare with current AES implementation benchmarks (see Section 3). Candidates which are not able to outperform AES implementations in terms of compactness and performance can probably not be advanced further in the eSTREAM Profile II project. Secondary, we compare among eSTREAM candidates. An open question is whether the value of versatile algorithms that are proposed for both Profile I and Profile II is considered differently than pure Profile II submissions.

Note that the most important criterium for analysis of eSTREAM, *i.e.*, mathematical security of the algorithm, is not evaluated as part of this framework.

2.1 Compactness (Area)

For the hardware oriented stream ciphers, the silicon area determines the cost of the implementation. This feature is one of the first to be taken into consideration, because the main goal of stream ciphers is to be smaller than block ciphers. That is why the area of the proposed stream ciphers should be compared to the area of a compact AES implementation. The benchmarks that can be used for this comparison are described in Sect. 3.

2.2 Performance (Throughput)

The properties that are taken into account when evaluating the performance of the stream cipher implementation are frequency, bits per second (throughput) and bits per cycle. Performance is, together with area, one of the most important design criteria. In Sect. 3 performance benchmarks are given for area constrained AES implementations.

2.3 Power Consumption

As stream ciphers are used in small handheld devices, power consumption should be taken into account to estimate the battery's capabilities. However, estimating the power consumption of a design is not straightforward. Power estimation tools such as SPICE can help for this matter, but are not always reliable especially without back-annotating physical layout information.

2.4 Flexibility/Scalability/Pipelining

The flexibility of a stream cipher is determined by the variety of possible implementation options. A high flexibility usually results in a large design parameter space with area and performance as the two main dimensions: implementations can be optimized for speed or for area or the design criterium can be a trade-off of these two. By scalability we mean the ability to scale the design with respect to the width of the data path. This results again in a trade-off between area and speed. Inserting registers for pipelining allows to increase the frequency and the throughput of the implementation.

These criteria do not only consider the inherent flexibility/scalability/pipelining of the design stressed by the author, but also possibilities to realize these properties detected by the implementer.

2.5 Simplicity/Completeness/Clarity

Because the new stream cipher standard will be adopted in many applications, the description should be clear. More specific, all details needed for the implementation should be given in the describing document. To decrease the non-recurring engineering time, a simple description is preferred. Some stream ciphers are more simple by nature and therefore allow a more simple description. However, even the more complicated stream ciphers should be introduced in an illustrative manner. That is why the new stream ciphers should be evaluated on simplicity, completeness and clarity of the describing document.

3 AES Hardware Implementation

The Advanced Encryption Standard (AES) [11] was standardized by the National Institute of Standards and Technologies (NIST) in 2001. AES is a block cipher that operates on 128-bit blocks of data using a 128-bit, 192-bit or 256-bit key. The most common key size is 128-bit and is solely considered in this testing framework. For a complete specification of AES we refer to [11].

A recent report with a strong focus on AES hardware architectures can be found in [5]. For the purpose of this testing framework, the lightweight implementations of [5] are the most important ones.

Most of the previous work on compact AES implementations outlines benchmarks for either ASIC or FPGA implementations. Here, we aim to give both benchmarks for ASIC and FPGA implementations as FPGAs have attracted more attention in the last years. Therefore, we selected two reference implementations for both ASIC and FPGA implementations.

For ASIC implementations, the reference implementations are from Feldhofer *et al.* [6] and Satoh *et al.* [14]. The former uses an 8-bit architecture and is currently the most compact AES ASIC implementation. On the other hand the work of Satoh *et al.* [14] gives results for different architectures ranging from 32-bit to 128-bit and therefore yields an increased throughput of data. In Table 1 we give the circuit benchmarks based on compactness. Both implementations use combinatorial logic for the S-Box implementation which is more suited for low-cost implementations than the use of a ROM table. There is also the work of Canright [2] that evaluates all options for basis, irreducible polynomial *etc.* to make the S-Box implementation even more compact in order to obtain further optimizations.

For low-cost FPGA benchmarks we select Good/Benaissa [7] and Chodowiec/Gaj [4] as references. The former is based on an 8-bit architecture, whereas [4] uses a 32-bit architecture. Benchmarks are summarized in Table 2.

	Feldhofer [6]	Satoh [14]	Satoh [14]	Satoh [14]
Architecture	8-bit	32-bit	64-bit	128-bit
No. S-boxes	1	4	8	20
Area [GEs]	3,400	5,398	7,998	12,454
Cycles per encryption [1]	1,032	54	32	11
Throughput [bits/cycle]	0.12	2.37	4.00	11.64
Technology [μm]	0.35	0.11	0.11	0.11
Clock frequency [MHz]	80	131	137	145
Throughput [Mbps]	9.9	311	548	1,691

Table 1. Benchmarks for AES-128 low-cost ASIC Implementations

	Good/Benaissa [7]	Chodowiec/Gaj [4]
Architecture	8-bit	32-bit
No. S-Boxes	1	4
FPGA	Xilinx Spartan-II XC2S15-6	Xilinx Spartan II XC2S30-6
Slices	124	222
No. of Block RAMs	2	3
Bits of Block RAM used	4,480	9,600 [7]
Total Equiv. Slices	264	522 [7]
Clock frequency [MHz]	67	60
Throughput [Mbps] [2]	2.2	69

Table 2. Benchmarks for AES-128 low-cost FPGA Implementations

[1] [6] includes the key schedule. For [14], add ten cycles for the key schedule.

[2] For comparison we use the definition of average throughput given by [7].

4 Performance Evaluation

The hardware performance measurements will be similar to Round 2 of AES where different AES candidates were implemented by NSA in an unbiased way. The design analysis consists of hardware designing (mostly based on the stream cipher designers' suggestions), coding in a hardware modeling language, simulation and synthesis for various hardware platforms. We would be concentrating on the low cost FPGAs and semi-custom ASIC with standard CMOS libraries. For a fair analysis, we provide an equivalent treatment for all the ciphers with basic optimizations that would be done during the normal hardware design phase. This would provide a meaningful comparison between the results of various designs and may be suitable only for this specific context of hardware performance measurement.

In Section 2, we mentioned the various performance parameters that will be considered. Since all performance parameters cannot be met in a single design, we would have to find possible trade-offs and possibly implement multiple designs. The flexibility of the algorithm would be the deciding factor for multiple designs. But compiler design constraint settings like delay and area are also another way to find various trade-off points. Our main approach will be to find designs that have low area and medium speed. An iterative kind of algorithm would be the standard choice for the designs.

We would be measuring the key-setup time, iv-setup time and the throughput performance of each of the designs. Our designs will be compared with efficient low-area implementations of AES mentioned in Section 3. Our aim would be to find designs that would be more compact than a low-area AES design but still faster in performance.

The different designs will be modeled using VHDL (VHSIC Hardware Description Language). The designs will be implemented following the standard methodology used by ASIC designers. This would include identifying various sub-blocks from the algorithm that would help to implement a small area iterative design. During this phase, a major deciding factor would be the algorithmic designer's suggestions mentioned in the specifications submitted to eSTREAM. A different approach would be taken only if the hardware designer feels a huge gain in performance than the one suggested. This will be followed by simulation and synthesis of the design model under different area/delay constraints to obtain the various performance measurements. The final physical layout and fabrication for ASIC designs would be beyond the scope of this testing.

For the unbiased approach we neglect the overhead for interfacing to the outside world by providing a standardized interfacing within each of the implementations. Though any input parameter needs that are constraining to a good hardware design would be noted in the final results. This user interface provides the algorithm with the key, initialization vector and the plaintext. It receives the key stream from the algorithm and XORs it to the plaintext, providing the ciphertext to the outside world. All other control signaling to the algorithm are also done from a common control block.

5 Evaluation of Other Implementation Properties

Besides performance criteria, we aim to evaluate also other implementation properties of stream ciphers in Phase II.

This task consists of the test categories

1. Design Analysis,
2. Side Channel Susceptibility,
3. Fault Analysis Susceptibility and
4. Probing Susceptibility.

The task "Design Analysis" deals with possible improvements and guidance for the final specification of the algorithms. The remaining three tasks evaluate the susceptibility of the implementations of eStream candidates towards implementation attacks. Counteracting implementation attacks typically requires additional implementation costs which are not considered in Section 2, yet.

Each task is explained in more detail below.

5.1 Design Analysis

The other main objective of the design analysis would be to find hardware efficient sub-blocks in the various algorithm. This will provide an easily identifiable list of functions that are good for hardware design and hence enable cryptographers to design a more hardware efficient stream cipher in the future.

5.2 Side Channel Susceptibility

Here we discuss vulnerabilities of hardware implementations of stream ciphers to side-channel attacks. It is very important to consider these already in the design phase as from the previous work some general recommendations for the design and countermeasures are known.

Implementation attacks in general exploit weaknesses in specific implementations of a cryptographic algorithm. Sensitive information, such as secret keys or a plaintext can be obtained by observing some side-channel information such as the power consumption, the electromagnetic radiation, *etc.*

In the 90's Kocher *et al.* performed successful attacks by measuring the power consumption while the cryptographic circuit is executing the implemented algorithm [9]. The most straightforward power analysis, called Simple Power Analysis (SPA), uses a single measurement to reveal the secret key by searching for patterns in the power trace. However, implementations that are resistant against SPA attacks, can still be broken by using a more advanced technique, namely Differential Power Analysis (DPA). In this case many power measurements are evaluated using statistical analysis. A similar terminology is used when the observed side-channel is electromagnetic radiation. In that case typical attacks are SEMA and DEMA.

Template attacks were invented by Chari *et al.* [3] and it was shown by Rechberger [12] that they can be also a serious threat to stream ciphers as well as all other ciphers.

From the power and electromagnetic analysis point of view there is not much previous work done on stream ciphers. However, the work of Lano *et al.* considers a DPA attack on synchronous stream ciphers with resynchronization mechanism [10]. Hence, their conclusion should be verified for the candidates in this class of stream ciphers. Also the work of Rechberger and Oswald [13] gives some recommendation for stream ciphers in order to avoid simple side-channel attacks.

5.3 Fault Analysis Susceptibility

Fault analysis is an active implementation attack that aims to disturb the computation of a cryptographic algorithm in such a way that an erroneous result is obtained. By applying mathematical cryptanalysis these erroneous results can be used to extract cryptographic key material. Reference [8] provides several general attacks that are applicable at LFSR based stream ciphers. For RC4, two different approaches have been presented in [1].

In this task, it is evaluated whether an eSTREAM candidate is vulnerable against one of the general techniques of [8]. If so, the complexity of a successful attack is estimated. Additionally, alternative approaches of fault analysis are checked.

5.4 Probing Susceptibility

Probing is an active implementation attack that directly connects to the circuit and allows monitoring of internal data flow.

In this task, the susceptibility of the implementation of eSTREAM candidates towards probing attacks is evaluated. Our approach first identifies critical connections within the implementation. The metric used for evaluation is the entropy loss (of the key, respectively, of the current state) at each critical connection as well as the maximum entropy loss by probing a few critical connections simultaneously.

6 Ongoing Test Activities

Due to the number of submissions, current test activities have started first by using the remaining candidates that are not ‘broken’ yet by mathematical analysis. After moving to Phase II it is assumed that also selected algorithms with a tweaked version are included in Profile II performance testing.

Actually, the submissions tested at the transition to Phase II are summarized in Table 3.

Profile I and II	Profile II
Hermes8	EDON-80
NLS (2A)	MICKEY / MICKEY-128
Phelix (2A)	MOSQUITO
Rabbit	Trivium
Salsa20	VEST (2A)

Table 3. Candidates under test for both Profile I and II candidates and Profile II candidates (in alphabetical order).

7 Conclusion

Currently, test specifications are still in a draft state. We encourage any third-party contributions and assessments!

References

1. Eli Biham, Louis Granboulan, and Phong Nguyen. Impossible Fault Analysis of RC4 and Differential Fault Analysis of RC4. In *The State of the Art of Stream Ciphers, Workshop Record*, pages 147–155. ECRYPT Network of Excellence in Cryptology, 2004.
2. Dan Canright. A Very Compact S-Box for AES. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2005*, volume 3659 of *LNCS*. Springer, 2005.
3. S. Chari, J.R. Rao, and P. Rohatgi. Template attacks. In B.S. Kaliski Jr., Ç.K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, number 2523 in Lecture Notes in Computer Science, pages 172–186, Redwood Shores, CA, USA, August 13-15 2002. Springer-Verlag.
4. Pawel Chodowiec and Kris Gaj. Very Compact FPGA Implementation of the AES Algorithm. In Colin D. Walter, Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 2779 of *LNCS*, pages 319–333. Springer, 2003.
5. Martin Feldhofer, Kerstin Lemke, Elisabeth Oswald, François-Xavier Standaert, Thomas Wollinger, and Johannes Wolkerstorfer. State of the Art in Hardware Architectures. Technical report, ECRYPT Network of Excellence in Cryptology, 2005.
6. Martin Feldhofer, Johannes Wolkerstorfer, and Vincent Rijmen. AES Implementation on a Grain of Sand. *IEE Proceedings on Information Security*, 152:13–20, October 2005.
7. Tim Good and Mohammed Benaissa. AES FPGA from the Fastest to the Smallest. In Josyula R. Rao, editor, *Cryptographic Hardware and Embedded Systems — CHES 2005*, volume 3659 of *LNCS*, pages 427–440. Springer, 2005.
8. Jonathan J. Hoch and Adi Shamir. Fault Analysis of Stream Ciphers. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 3156 of *LNCS*, pages 240–253. Springer, 2004.

9. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology: Proceedings of CRYPTO'99*, number 1666 in Lecture Notes in Computer Science, pages 388–397. Springer-Verlag, 1999.
10. J. Lano, N. Mentens, B. Preneel, and I. Verbauwhede. Power analysis of synchronous stream ciphers with resynchronization mechanism. In *In ECRYPT Workshop, SASC - The State of the Art of Stream Ciphers*, pages 327–333, 2004.
11. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, 2001.
12. C. Rechberger. Side-channel analysis of stream ciphers. Master's thesis, TU Graz, Austria, 2004.
13. C. Rechberger and E. Oswald. Stream ciphers and side-channel analysis. In *In ECRYPT Workshop, SASC - The State of the Art of Stream Ciphers*, pages 320–326, 2004.
14. Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A Compact Rijndael Hardware Architecture with S-Box Optimization. In Colin Boyd, editor, *Advances in Cryptology — Asiacrypt 2001*, volume 2248 of *LNCS*, pages 239–254. Springer, 2001.