Date: Sat, 13 May 2000 15:07:45 +0200
From: BORA <p.bora@gorc.imbo.wat.waw.pl>
X-Mailer: Mozilla 4.5 [pl] (Win98; I)
X-Accept-Language: pl
To: AESround2@nist.gov
Subject: Comment to implementation of SERPENT algorithm

I'm sending you on AES discussion forum our paper about implementation of SERPENT algorithm in ALTERA chips FPGA.This implementation is a review of possibilities of realisation this algorithm in curently accessible FPGA ALTERA chips series FLEX10K.

IMPLEMENTATION OF THE SERPENT ALGORITHM USING ALTERA FPGA DEVICES

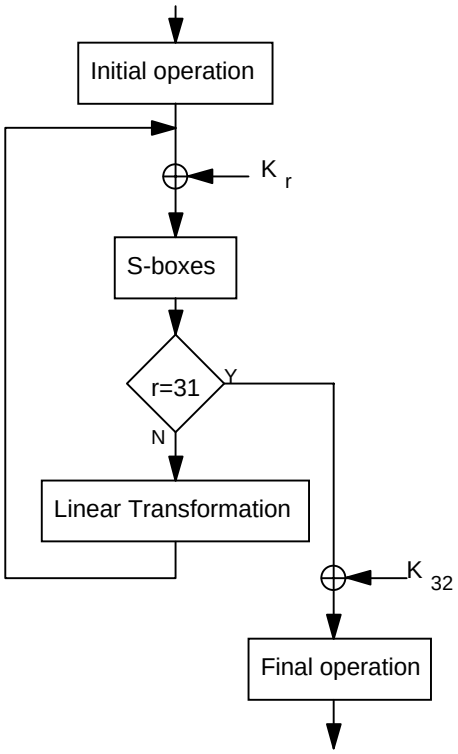
Piotr BORA*, Tomasz CZAJKA**
* - Military University of Technology – Institute of Mathematics and Operations Research -
- Faculty of Cybernetics
** - Military Communication Institute

INTRODUCTION

Algorithm SERPENT and algorithms RJINDAEL, TWOFISH, MARS, RC6 are accepted in second round Advanced Encryption Standard. This algorithm is very simple for hardware implementation.

THE CIPHER

The SERPENT block cipher algorithm was designed by R. Anderson, E. Biham, L. Knudsen and its specification is given in [6]. The schemes of encryption and decryption units of the SERPENT algorithm are depicted in Figures 1 and 2.



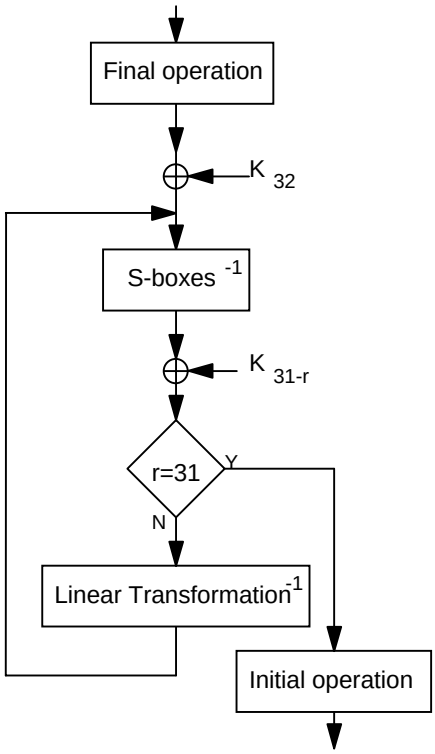


Fig. 1. The encryption unit. Fig. 2. The decryption unit.

We have implemented the version of the algorithm with 128 bit blocks of encrypted text and 128 bit encryption key. The SERPENT has 32 rounds and each round needs one 128 bit subkey and one 128 bit subkey is needed to EXOR with the text block. The 33 subkeys are generated by the key generation algorithm depicted in Figure 3.

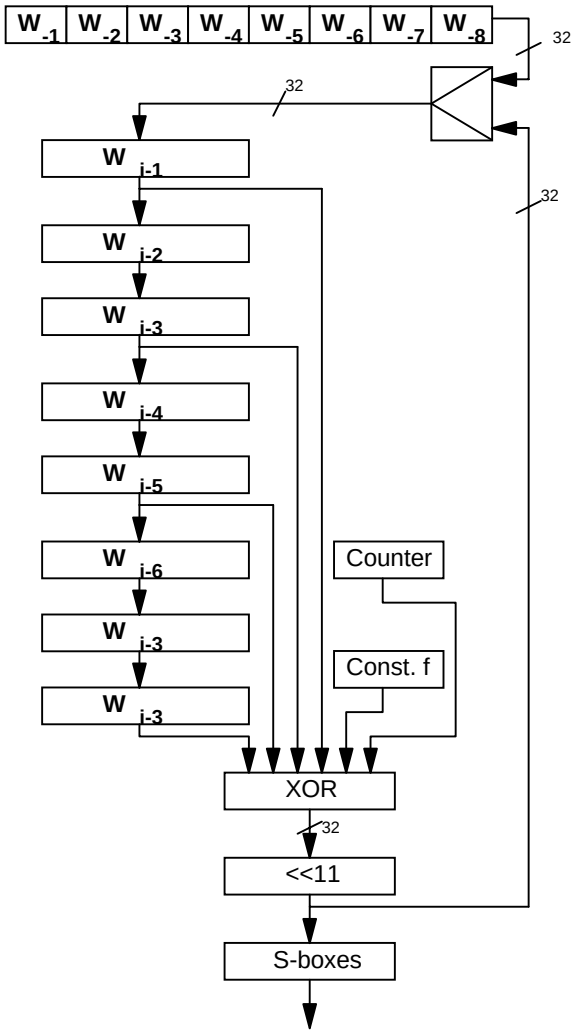


Fig. 3. The key generation algorithm.

The key generation algorithm processes 32 bit blocks and it is difficult to implement it in such a way that the subkey is calculated parallely with the realisation of the corresponding round. We have implemented the key generation algorithm as a separate unit of the cipher and the subkeys are stored in a memory block of 4096 cells.

IMPLEMENTATION OF ALGORITHM IN FPGA

The encryption and decryption algorithms have the same complexity. The decryption algorithm realises the inverse transformation of those used in the encryption process and the order of them is opposit. In practice the decryption unit needs a little more place on the chip.

The results of our implementation are given in Table 1.

Tab. 1. The SERPENT implementation.

Kind of implementation	Count of logic cells needed for the algorithm realisation (without key generation)	Minimum time of operation in ns	Speed of encryption in Mb/s
One round in one operation	1 976	34	117
Two round in one operation	2 259	45	177
Four round in one operation	2 444	70	228
Eight round in one operation	2 743	106	301

Serpent uses the set of different S-boxes with four bits inputs and four bits outputs. In each round there is used one of S-box working parallely 32 times. After first eight rounds all S-boxes are used and this unit of eight rounds is repeated four times. The S-box occupies four cells of the FPGA circuit and we need to implement 256 S-boxes which gives 1024 cells, there is needed additionally the multiplexer of choosing the S-box. Hence the independent implementation of each algorithm round is not economical. Table 1 gives the results of implementation one round, two rounds, four rounds and eight rounds, respectively. We can achieve this way some increase of the algorithm speed and decreasing the rewriting number of the intermediate results. On the other hand it increases the complexity of the whole logic circuit since the linear transformations (LT) must be copied; each LT occupies 128 cells and in the case of realising eight rounds as a one logic unit we need 896 cells. Table 2 gives the times of realisations of the transformations in the SERPENT round implemented using the ALTERA chips from the series FLEX10K.

Tab. 2. SERPENT round transformations.

Kind of type module in algorithm	Count of logic cells	Orientation time of module realisation in [ns]
LT	128	6
S-boxes (with multipexer 1 from 8)	1796	22
XOR with key	128	6

These are the results of algorithm simulation working when one round is implemented as one logic unit. In the case when several rounds are implemented as a logic unit the time of algorithm realisation is decreasing since the multiplexes choosing the S-boxes are reduced.

The subkeys are generated at the initial stage of algorithm realisation and they are stored in the EAB (Embedded Array Block) memory block or in additional logic cells. When the subkeys are stored only in logical cells then we need 4096 cells. This implementation does not depend on the used platform: e.g. ALTERA or XILINX. In the chips ALTERA series FLEX10KA there is needed about 7300 logic cells. When it increases the number of rounds put in on logic unit, then the number of cells needed to implement the algorithm is almost constant since then it decreases the number of multiplexers inputs needed to choose the subkeys and the S-boxes. The practical experience indicates that the most economical is the SERPENT implementation when the eight different rounds are put in one logical unit.

It would be a simplification of the whole circuit to put the subkey block in the EAB, then smaller structures of FPGA chips series FLEX10K could be used. It is impossible to implement this way the eight round of SERPENT in one logic unit.

CONCLUSIONS

The implementation of SERPENT algorithm is very simple. The optimal implementation of this algorithm is achieved when in one operation is realised eight rounds. This implementation needs four signal of clock to generate the encryption/decryption. This implementation has a very high operation speed >300Mb/s. Pipelining with realisation of more rounds count of needed logic cells dramatically increased.

To realise the pipeline realisation we need drastically more logic cells of the circuit. The full pipelining can lead to achieve the speed about from 1,2 Gb/s to 4 Gb/s.

ACKNOWLEDGEMENTS

We would like to sincerely thank Eli Biham for valuable comments and discussions on the implementations of the cipher SERPENT. We are grateful Janusz Szmidt for encouragement in the work and the help in the redaction of this article.

REFERENCES

1. Tłuba, K. Jasiński, B. Zbierzchowski „Specjalizowane układy cyfrowe w strukturach PLD i FPGA", Warszawa 1997r.

2. G. De Micheli „Synteza i optymalizacja układów cyfrowych" , Warszawa 1998r.

3 V. Fischer „realization of the round 2 AES candidates using ALTERA FPGA" http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html, March 2000r.

4. K. Gaj, P. Chodowiec „Implementations of the AES Candidate Algorithms using FPGA devices" Technical Report, George Mason University, April 2000r.

5. J. Nechvatal, E. Barker, D. Dodson, M. Dvorkin, J. Foti, E. Roback „Status Report on the First Round of the Development of the Advanced Encryption Standard" NIST report, August 1999r.

6. R. Anderson, E. Biham, L. Knudsen „SEPRENT: A Proposal for the Advanced Encryption Standard", http://csrc.nist.gov/encryption/aes/round2/AESAlgs/Serpent/Serpent.pdf.

7. Eli Biham, the private communications during his visit in Poland.