A Hardware Implementation of CURUPIRA Block Cipher for Wireless Sensors

Paris Kitsos[1]
Computer Science
Hellenic Open University and Dept. of Computer Science and Technology
University of Peloponnese, Greece
E-mail: pkitsos@ieee.org

George Selimis and Odysseas Koufopavlou
VLSI Lab, Department of Electrical & Computers Engineering,
University of Patras, Patras, Greece
E-mails:(gselimis, odysseas) @ece.upatras.gr

Athanassios N. Skodras
Computer Science
Hellenic Open University, Greece
E-mail: skodras@eap.gr

Abstract

An architecture and VLSI implementation of a new block cipher called Curupira is presented in this paper. This cipher is suitable for wireless sensors and RFID applications. Our 0.13 μm implementation requires resources of 9450 gate equivalences and is capable to encrypt a plaintext in 10 clock cycles. The cipher achieves a maximum throughput up to 2361 Mbps at 246 MHz for encrypting/decrypting. When clocked at 100 KHz a throughput of up to 960 Kbps is achieved and an average power of 0.04 mW is drawn.

1. Introduction

Wireless sensor networks (WSN) are making their way from research to real-world deployments. Personal-area networks, intelligent homes, environmental monitoring there is almost nothing left that is not going to be "smart" and "networked" [1]. On the other hand, Radio Frequency Identification (RFID) systems allow products to be read simultaneously and provide a reading range superior to existing systems. In such a system, RFIDs have to be highly cost effective. As its size directly affects the manufacturing cost, so the design of tag's integrated circuit has to be highly silicon efficient.

Block ciphers have been used as a basic security building block for applications such as smart cards. Similar algorithms can be applied to tiny devices such WSNs and low-cost RFID tags. In such devices it is impossible to implement multiple security primitives for cost reasons. For example, in the case of RFID tags it is estimated that security resources may be limited to hundreds of bits of storage, roughly 500-5,000 gates [2]. So a compact hardware implementation of a block cipher could be the most promising candidate for security in such applications.

Another critical issue in this type of devices is the limited amount of power available. Only a small, finite amount of energy may be available from the miniature batteries that use these devices. [1]

[1]This work was funded by the State Scholarships Foundation

Until now, there are some block cipher implementations for RFID or WSNs applications. The DESL, DESX and DESXL [3] are three of them. They achieve a throughput of up to 44.4 Kbps at a clock frequency of 100 KHz. Their hardware resources cover 1848, 2629 and 2168 GE respectively. Another block cipher is the mCrypton [4]. It achieves a throughput of up to 492.30 Kbps and covers 2681 GE. Finally, in [5] and [6] two lightweight block ciphers are presented, Hight and Present, respectively. The first one achieves a throughput of 188.20 Kbps and covers up to 3048 GE, while the second one for 80-bit key achieves a throughput of 200 Kbps and covers up to 1570 GE.

In this paper the first hardware implementation of the new lightweight block cipher called Curupira [7] is investigated. This paper demonstrates the area and time efficiency of this cipher. Curupira operates on a 96-bit cipher state using a key of 96-, 144- or 192-bit for variable number of rounds from 10 to 23.

This paper is structured as follows. In Section 2 the Curupira block cipher is briefly presented. In Section 3, the proposed architecture is analyzed. In Section 4, synthesis results and comparisons with previous published block ciphers are given. Finally, Section 5, concludes the paper.

2. The Curupira block cipher

The Curupira block cipher is an iterated block cipher that operates on a 96-bit cipher state organized as a matrix M_4 (3x4 matrix with arguments equal to one byte). It uses a $48t$-bit ($2 \leq t \leq 4$) key organized as a matrix M_{2t}.

2.1. The round function

The round function, $\rho[k^{(r)}]$, is based on combined operations from three algebraic functions. These functions are the nonlinear layer γ, the permutation layer π, and the linear diffusion layer θ. The round function is parameterized by the key k, and given by: $\rho[k^{(r)}] \equiv \sigma[k^{(r)}] \circ \theta \circ \pi \circ \gamma$. The block diagram of the block cipher basic round is shown in Fig. 1. The layer γ consists of the parallel application of a nonlinear S-box. The actual S-box can be computed from two

miniboxes, P and Q, as the Fig. 2 shows. The key addition $\sigma[k]$ consists of the bitwise addition of the key matrix as the following equation describes, $\sigma[k](a) = b \leftrightarrow b_{i,j} = a_{i,j} \oplus k_{i,j}, 0 \le i < 3,\ 0 \le j < n$.

2.2. The key scheduling round function

A $48t$-bit user key $\mathcal{K}$, externally stored as a byte array of length $6t$ is internally represented as a matrix such that $K_{i,j} = \mathcal{K}[i+3j],\ 0 \le i < 3,\ 0 \le j < 2t$. The cipher key is updated during the cipher operation by a reversible transform defined by two linear operations ξ and μ.

Also, let K, be the cipher key. Defining the initial key stage as $K^{(0)} = K$, the key evolution function, ψ, computes key stage $K^{(r)}$ from the key stage $K^{(r-1)}$ as $\psi_r \equiv \mu \circ \xi \circ \sigma[q^{(r)}] = \omega \circ \sigma[q^{(r)}]$. So, the key stages are computes according to the following equation $K^{(r)} = (O_{i=1}^{r} \omega \circ \sigma[q^{(i)}])(K)$. Fig. 3 depicts the block diagram of key scheduling round function for the encryption operation. For the decryption operation the key evolution function becomes $\psi_r^{-1} = \sigma[q^{(r)}] \circ \omega^{-1}$. The effective round subkeys $k^{(r)}$ used by the cipher are computed via the key selection function, ϕ_r defined so that: $k^{(r)} = \phi_\kappa(\mathrm{K}) \leftrightarrow k_{0,j}^{(r)} = S[K_{0,j}^{(r)}]$ and $k_{1,j}^{(r)} = K_{i,j}^{(r)}$ for $i > 0,\ 0 \le j < 4$.

2.3. The complete block cipher

Curupira is defined for the cipher key K and R rounds as the mapping $CURUPIRA[K]$ given by $CURUPIRA[K] = \sigma[k^{(R)}] \circ \pi \circ \gamma \circ (\mathrm{O}_{r=1}^{R-1} \sigma[k^{(r)}] \circ \theta \circ \pi \circ \gamma) \circ \sigma[\kappa^{(0)}]$.

3. Proposed hardware architecture

The proposed Curupira implementation uses 96-bit key and operates for 10 rounds. Consists of the two main components: the Key Scheduling Unit, which is responsible for the round keys generation, and the Curupira Core Unit, which executes the basic encryption procedure.

The VLSI implementation of the Curupira Core Unit is illustrated in Fig. 4. The nonlinear layer γ, is composed of 12 substitution tables, S-Boxes. It consists of two 4-bit miniboxes P and Q. These miniboxes have been implemented by Look-Up-Tables (LUTs). Next, the permutation layer π, is implemented by using wired shifters. The first row of data is kept unchanged. In the second row, the data of the first argument with the second one, and the data of the third argument with the forth one, are swapped. Finally in the third row, the data of the first argument with the third one and the data of the second argument with the forth one, are swapped. The diffusion layer θ, is a matrix multiplication between the round state and a generator matrix, D. The layer θ input stage is a 3x4 matrix. Each argument of this matrix is one byte. An efficient method for hardware implementation is provided in equation 1. $a_{0,i}$, $a_{1,i}$ and $a_{2,i}$ represent the three bytes of the i column of the output state of θ layer.

$$
\begin{aligned}
a_{0,i} &= b_{0,i} \oplus X[b_{0,i} \oplus b_{1,i} \oplus b_{2,i}] \\
a_{1,i} &= b_{1,i} \oplus X^2[b_{0,i} \oplus b_{1,i} \oplus b_{2,i}] \quad (1) \\
a_{2,i} &= b_{1,i} \oplus X[b_{0,i} \oplus b_{1,i} \oplus b_{2,i}] \oplus \\
&\quad X^2[b_{0,i} \oplus b_{1,i} \oplus b_{2,i}]
\end{aligned}
$$

Table X implements the multiplication by the polynomial $g(x) = x \bmod x^8 + x^6 + x^3 + x^2 + 1$ in $GF(2^8)$. The key addition ($\sigma[k^{(r)}]$) consists of eight 2-input XOR gates for any byte of the state.

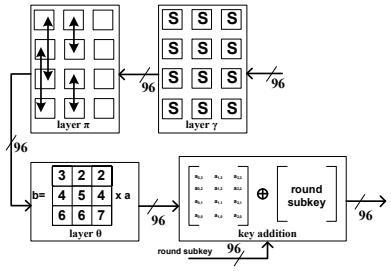

Fig. 1. Block diagram of the basic round

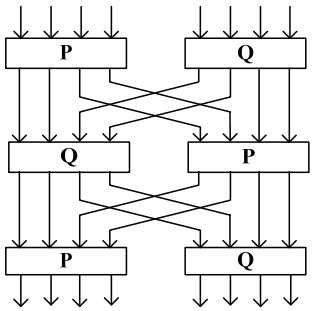

Fig. 2. The Curupira S-box

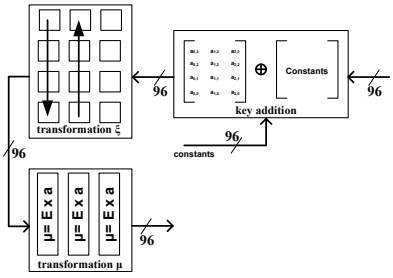

Fig. 3. Block diagram of the key scheduling

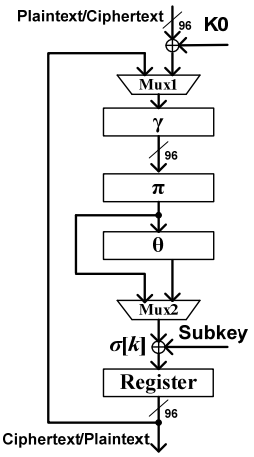

Fig. 4. VLSI implementation of the Curupira core unit

Every bit of the round key is XORed with the appropriate bit of the cipher state. The Key Scheduling Unit is depicted in Fig. 5. The two basic functions, Key Evolution and Key Selection, are also separated.

The Key Evolution Function mainly comprises of the key addition component, the linear transforms ξ and μ, five 2-to-1 96-bit multiplexers and a 96-bit register. The key addition ($\sigma[k^{(r)}]$) consists of eight 2-input XOR gates for any byte of the state. Every bit of the first row of the state is XORed with the appropriate bit of the, predefined and stored in the ROM, constants ($q^{(s)}$). So, a 32-bit XOR gate is used. Then, the linear transform ξ, rotates its input state according to the following rule: it keeps the first row of its argument unchanged, rotates the second row one position to the left, and rotates the third row one position to the right during the encryption operation. During the decryption operation the linear transform works reversely. The flag e defines the cipher operation. Finally, the transform μ is a matrix multiplication between the input state and a generator matrix E. An efficient method for hardware implementation (Fig. 6) is provided. $a_{0,i}$, $a_{1,i}$ and $a_{2,i}$ represent the three bytes of the i column of the output state of the transform μ.

During the encryption the e flag is true (e=1), while during the decryption the e flag is false ($\bar{e}=0$). The implementation shares two tables, X and $X^2 = X \circ X$, and eight 2-input XOR gates.

The Key Selection Function mainly consists of four S-boxes and a diffusion layer θ. The multiplexer selects the appropriate subkeys for encryption (e) or decryption ($\bar{e}$). The S-boxes are applied in the first row of the key state.

The plaintext/ciphertext block is set to the Curupira Core Unit (Fig. 4) input simultaneously with the 96-bit key to the Key Scheduling Unit (Fig. 5) at a t=0.

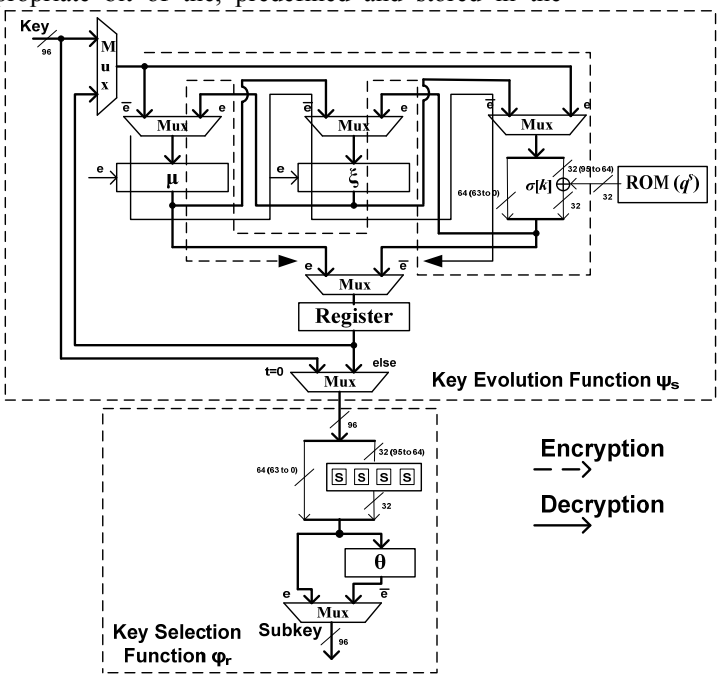

Fig. 5. VLSI Implementation of the Key Scheduling Unit

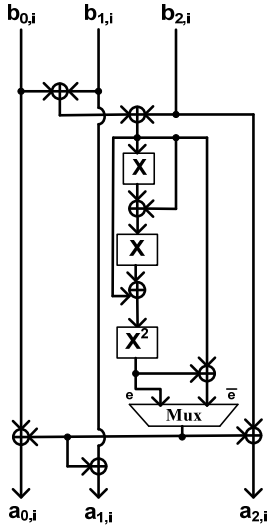

Fig. 6. Compact Implementation for μ

From this key the initial subkey (K0) is generated and added to the 96-bit plaintext/ciphertext (Fig. 4). The rest 10 subkeys are XORed (for $0<t<9$) with the cipher state of the θ layer. The last subkey is XORed with the cipher state of the π layer (through the multiplexer). A round subkey is produced in one clock cycle. After 10 execution rounds the ciphertext/plaintext is produced. Each produced subkey is also used in the next clock cycle for the production of the next subkey by the Key Scheduling Unit. The dashed path indicates the operation during the encryption, while the solid path indicates the decryption operation.

4. Synthesis results and analysis

The proposed implementation was captured by using VHDL, with structural description logic. The VHDL code has been synthesized with Synopsys synthesis tools (Design Compiler) and the target technology library is 0.13 μm. The total power dissipation is measured using Synopsys PrimePower.

The 4-bit miniboxes (P and Q) were designed by using Look Up Tables (LUTs).

The proposed implementation achieves a throughput equal to 2361 Mbps for a maximum frequency of 246 MHz. When clocked at 100 KHz a throughput of up to 960 Kbps is achieved and an average power of 0.04 mW is drawn.

The experimental results are shown in Table 1. Since no other implementations do exist, comparisons with other lightweight block ciphers, [3]–[6] have been added. The current sensors and RFID applications allow an operating clock frequency up to 100 KHz.

Table 1. Experimental Results and Comparison s

Cipher	Cycles per block	Throughput at 100 KHz (Kbps)	Logic Process (μm)	Area (GEs)
DESL [3]	144	5.55	0.18	1848
mCrypton [4]	13	492.30	0.13	2681
Hight [5]	34	188.20	0.25	3048
Present [6]	32	200	0.18	1570
Curupira	10	960	0.13	9450

The algorithm constants ($q^{(s)}$) are stored in 10x32 bit ROM blocks. The proposed implementation achieves the highest throughput at the expense of a higher number of GEs. This is an expected result of the algorithm philosophy and not an implementation tradeoff. Also, the proposed implementation handles data up to 128-bit (96-bit plaintext/ciphertext and 96-bit key) in parallel. The critical path is dictated by the cipher Key Scheduling Unit.

The mCrypton implementation in [4] achieves the best throughput and covers 1848 GEs. The DESL in [3] achieves the lowest throughput value equal to 5.55 Kbps. However, it fits better in 8-bit processors. The Hight [5] and Present [6] implementations achieve comparable time performance from 188.20 Kbps to 200 Kbps, respectively. Finally, the implementation in [6] consumes the least area resources compared to all others.

5. Conclusions

In this paper a hardware architecture for the new lightweight Curupira block cipher and its VLSI implementation are presented. The synthesis results show that the Curupira cipher implementation is a flexible solution in applications with area restricted specification demands. On the other hand, it is proved that the Curupira is not suitable for applications with ultra area restricted specification demands like RFID applications.

6. References

[1] C. Castelluccia, H. Hartenstein, C. Paar, and D. Westhoff, *"Security in ad-hoc and sensor networks"*, Proc. of 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), LNCS 3313, 2004.

[2] S. A. Weis, S. E. Sarma, R. L. Rivest and D. W. Engels, *"Security and privacy aspects of low-cost radio frequency identification systems"*, Conf. on Security in Pervasive Computing - SPC 2003, LNCS 2802, Springer-Verlag, 2003, pp.454-469.

[3] G. Leander, C. Paar, A. Poschmann, and K. Schramm, *"New Lighweight DES Variants"*, Fast Software Encryption 2007 – FSE 2007, vol. 4593 of LNCS, pp. 196–210. Springer-Verlag, 2007.

[4] C. Hoon Lim and T. Korkishko, *"mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors"*, WISA 2005, LNCS, vol. 3786, pp. 243–258, Springer, Heidelberg (2006).

[5] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, *"HIGHT: A New Block Cipher Suitable for Low-Resource Device"*, CHES 2006, LNCS, vol. 4249, Springer, Heidelberg (2006).

[6] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin and C. Vikkelsoe, *"PRESENT: An Ultra-Lightweight Block Cipher"*, CHES 2007, LNCS, vol. 4727, Springer, Heidelberg (2007).

[7] Paulo S. L. M. Barreto, Marcos A. Simplício, *"CURUPIRA, a block cipher for constrained platforms"*, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2007, 2007, Belém (PA).