Efficient Implementation of the Rijndael S-box

Vincent Rijmen*

Katholieke Universiteit Leuven, Dept. ESAT,
Kard. Mercierlaan 94,
B–3001 Heverlee, Belgium
vincent.rijmen@esat.kuleuven.ac.be

Abstract

We discuss an efficient hardware implementation of the Rijndael S-box.

1 Introduction

The Rijndael [2] S-box is based on the mapping $x \to x^{-1}$, where x^{-1} denotes the multiplicative inverse in the field. There exist several efficient methods to calculate multiplicative inverses in a finite field GF(2^m). In [1], an algorithm is presented, that is based on Euclid's algorithm. It has an area complexity of $O(m)$ and requires $2m$ time steps. Another possibility is to do calculations in GF(16). This method is discussed in the next section.

2 The Efficient Construction

Every element of GF(256) can be written as a polynomial of the first degree with coefficients from GF(16). Multiplication is performed modulo an irreducible polynomial with degree two. Denoting the irreducible polynomial as $x^2 + Ax + B$, the multiplicative inverse for an arbitrary polynomial $bx + c$ is given by

$$(bx+c)^{-1} = b(b^2B + bcA + c^2)^{-1}x + (c + bA)(b^2B + bcA + c^2)^{-1}.$$

*F.W.O. Postdoctoral Researcher, sponsored by the Fund for Scientific Research - Flanders (Belgium)

The problem of calculating the inverse in GF(256) is now translated to calculating the inverse in GF(16) and performing some multiplications, squarings and additions in GF(16). The inverse in GF(16) can be stored in a small table. We use an optimal normal basis in order to simplify the other operations. The squaring operation is then a simple rotate (which is for free in hardware) and the multiplication is rotation-symmetrical and simple. Moreover, we can use the freedom we have for the choice of A and B to select A equal to the unit element (denoted 1111) and B a value with low Hamming weight, say 0001. Figure 1 gives a schematic representation of the required calculations.

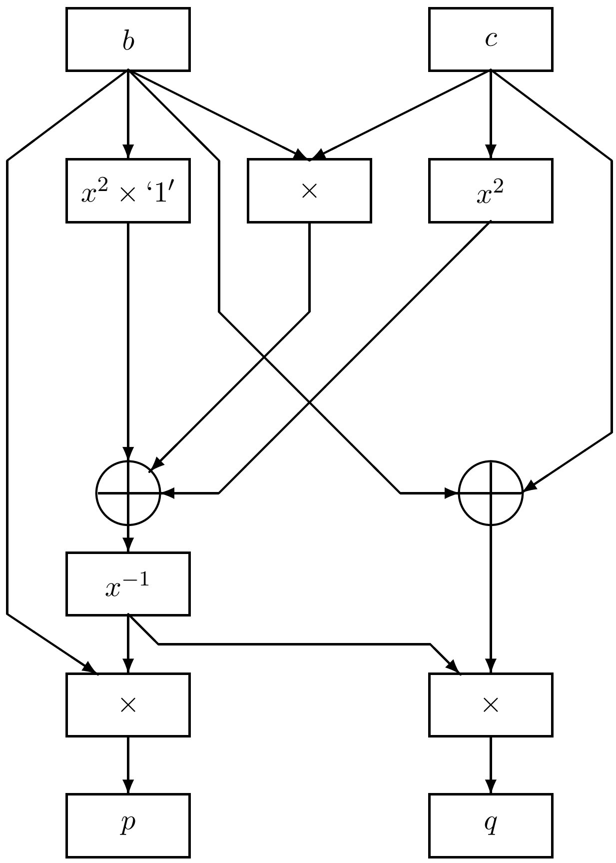

Figure 1: Schematic representation of a hardware-efficient calculation of the inverse in GF(2^8).

In order to implement the Rijndael S-box, the mapping $x \to x^{-1}$ must be followed by an affine transform. This transform can be implemented directly on the bit level. Probably, it is also possible to save a few more gates by using an affine transform that requires less gates, but has the same security level.

3 Conclusion

We did not implement the Rijndael S-box in hardware, nor did we try to simulate an implementation. If a good VHDL compiler is used, it might produce already an optimal circuit if the S-box is given as a table. In that case, the description given in this note has no practical consequences.

References

[1] H. Brunner, A. Curiger, M. Hofstetter, "On computing multiplicative inverses in GF(2^m)," *IEEE Transactions on Computers,* Vol. 42, No. 8, August 1993, pp. 1010–1015.

[2] J. Daemen and V. Rijmen, "The Block Cipher Rijndael," NIST's AES home page, `http://www.nist.gov/aes`.