How Lightweight Is the Hardware Implementation of Quasigroup S-Boxes

Article *in* Advances in Intelligent Systems and Computing · January 2013

DOI: 10.1007/978-3-642-37169-1_12

CITATIONS
8

READS
103

3 authors:

Hristina Mihajloska
Ss. Cyril and Methodius University in Skopje
15 PUBLICATIONS **49** CITATIONS
SEE PROFILE

Tolga Yalcin
Google Inc.
30 PUBLICATIONS **903** CITATIONS
SEE PROFILE

Danilo Gligoroski
Norwegian University of Science and Technology
172 PUBLICATIONS **1,573** CITATIONS
SEE PROFILE

How Lightweight is the Hardware Implementation of Quasigroup S-boxes

Hristina Mihajloska [1], Tolga Yalcin [2], and Danilo Gligoroski [3]

Faculty of Computer Science and Engineering, UKIM, Skopje, Macedonia
hristina.mihajloska@finki.ukim.mk
Embedded Security Group, HGI, Ruhr-University Bochum, Germany
tolga.yalcin@rub.de
Department of Telematics, NTNU, Trondheim, Norway
danilog@item.ntnu.no

Abstract. In this paper, we present a novel method for realizing S-boxes using non-associative algebraic structures - quasigroups, which - in certain cases - leads to more optimized hardware implementations. We aim to give cryptographers an iterative tool for designing cryptographically strong S-boxes (which we denote as Q-S-boxes) with additional flexibility for hardware implementation. Existence of the set of cryptographically strong 4-bit Q-S-boxes depends on the non-linear quasigroups of order 4 and quasigroup string transformations. The Q-S-boxes offer the option to not only iteratively reuse the same circuit to implement several different strong 4-bit S-boxes, but they can also be serialized down to bit level, leading to S-box implementations below 10 GEs. With Q-S-boxes we can achieve over 40% area reduction with respect to a lookup table based implementation, and also over 16% area reduction in a parallel implementation of PRESENT. We plan to generalize our approach to S-boxes of any size in the future.

Keywords: lightweight cryptography, S-boxes, ASIC implementation, quasigroup S-boxes

1 Introduction

Today, we live in a time where computing reaches its third phase, in which pervasive computing takes over important roles in daily life. Huge deployment of pervasive devices (lightweight solutions), on one hand, promises many benefits that makes our lives better, but on the other hand, opens huge number of questions related to the security and privacy of these devices. In this sense, implementing security into lightweight solutions requires special cryptographic techniques that should be applied in resource-constrained environment. These can be done via lightweight cryptography. The term "lightweight" does not necessary imply weak cryptography. On the contrary, its main goal is to achieve uncompromised security by means of new secure algorithms whose implementation requires as lightweight hardware and software area as possible [1].

Example for lightweight devices can be RFID tags or smart cards which are used in many industries. They are immensely used in electronic payment, product tracking, transportation and logistics, access control and identification systems. RFID tags use radio-frequency electromagnetic fields to transfer data from a tag attached to an object. In general, they can be divided into passive and active according to the power source that they use. Active RFID tags provide their own power supply in form of battery, whereas passive are powered by the electromagnetic fields used to read them [2]. They have to be extremely low-cost and low-power devices, meaning that they have to be implemented in minimal chip area. Therefore, in what follows we limit our focus to passive RFID tags. We note that in the rest of this paper talking about RFID tags, we refer to the passive tags.

Because of the fact that RFID tags are deployed into untreated environment where enemy has full control over the device and physical access to it, greater attention must be paid to their security. To provide security of these devices, new cryptographic protocols have been proposed in the last few years. These protocols basically used symmetric cryptography, which can be successfully implemented in constrained hardware environment. In most cases of RFID tags, low cost and low power require symmetric cryptographic primitives with low gate count to be used. In order to achieve better results for low gate count in RFID tags, researchers have focused on block ciphers [3] [4] and hash functions [5] [6].

The structure of the paper is the following. In Section 2, we give a brief overview of the lightweight implementation of S-boxes of proposed lightweight block ciphers. In Section 3, we present an iterative tool for designing cryptographically strong S-boxes with the help of quasigroups of order 4. In section 4, hardware implementation results for Q-S-boxes are provided and the results are compared with previous work. Finally, the paper is concluded with future directions in Section 5.

2 Previous Work

From cryptographic point of view, unarguably, the most critical components in lightweight symmetric cipher design are substitution boxes, S-boxes. They are responsible for confusing the input data via their highly non-linear properties. On the other hand, from a lightweight implementation point of view, they are the highest area occupying modules within the cipher and basically determine the overall cipher area. Therefore, implementing lightweight solutions for block ciphers depends on how lightweight can be presented their S-boxes.

In what follows, we are focused on lightweight S-box implementations. It is very common practice to state the area of a digital block implemented on silicon in term of gate equivalents (GE). One GE is equivalent to the area of the two-input NAND gate with the lowest driving strength of the corresponding technology [4].

mCRYPTON is the first special design of block cipher targeted for tiny pervasive devices. It is designed by following the overall architecture of Crypton,

but with a little bit simplification of the building blocks to fit the block/key sizes. The nonlinear substitution building block uses four 4×4-bit S-boxes, each of which occupies 27 GE [7].

In [8], a lightweight hardware implementation of the known cipher DES is presented by Leander *et al.* at FSE 2007. DESXL is a modified variant of DES, where the eight S-boxes are substituted by a single cryptographically stronger S-box which is repeated eight times. This 6×4-bit S-box is implemented in a serialized ASIC design, which requires 32 GE for a single 4-bit S-box.

In the same year, Bogdanov *et al.* at CHES conference presented a new ultra-lightweight block cipher, PRESENT [9]. The design of PRESENT is extremely hardware efficient, and the serial implementation of it with 80-bit key length requires as low as 1000 GE [4]. The PRESENT S-box is a 4-bit S-Box which requires 28 GE.

In 2009, Hummingbird is proposed as a new ultra-lightweight crypto algorithm for RFID tags by Engels and Smith *et al.* [10]. It has a hybrid structure of block cipher and stream cipher and was developed with a minimal hardware footprint in mind. Hummingbird uses four Serpent type 4-bit S-boxes which require hardware area in the rage of 20 to 40 GE [11].

In [12], a new lightweight cipher proposal is presented at CHES 2011. This is a 64-bit block cipher, LED, which is as small as PRESENT and faster in software, but slower in hardware. It uses the same PRESENT type S-box in its non-linear layer.

Also in 2011, TWINE, another of the family of lightweight block ciphers [13], was presented. It is the first cipher that combines features like no bit permutation, no Galois-Field matrix and generalized Feistel network. The components that are used are only one 4-bit S-box and 4-bit XOR. The ASIC hardware implementation of TWINE S-box is about 30 GE.

In all these cases, all research made on the 4-bit S-boxes has focused on lookup table based S-boxes. While, they are very well established and easy to analyze, their design space is very limited with a minimum achievable area of about 20 GE, without compromising security. They cannot be optimized via folding, serializing, or any other means, without additional area overhead. Our approach defers from the existing via its specific methodology, which offers a more optimized hardware implementation of S-boxes suitable for lightweight ciphers in RFID tags.

3 Our Approach

In this paper, we present a novel method for realizing S-boxes using non-associative algebraic structures - quasigroups, which - in certain cases - leads to more optimized hardware implementations. We aim to give cryptographers an iterative tool for designing cryptographically strong S-boxes (which we denote as Q-S-boxes) with additional flexibility for hardware implementation. Existence of the set of cryptographically strong 4×4-bit Q-S-boxes depends on the non-linear quasigroups of order 4 and quasigroup string transformations.

Quasigroups of order 4, themselves are 4×2-bit S-boxes. If they are presented as vector valued Boolean functions [14] and more precisely in their ANF, it can be seen that their maximal algebraic degree is 2. Quasigroup string transformations have an advantage to transform a given string with length n bijectively to output string with the same length n, and also, to raise the algebraic degree of the final bijection output. Therefore, quasigroups of order 4 with algebraic degree 2 (called non-linear quasigroups) and quasigroup string transformation (called e-transformation) with adequately chosen leaders generate 4×4-bit Q-S-boxes. An algorithm for generating Q-S-boxes is given in our previous paper [15]. The minimum number of rounds (iterations) for this methodology is 4. Hence, with this methodology we can generate Q-S-boxes in different ways depending on the number of rounds and the number of leaders that we have chosen. In the previous paper [15] we defined an optimal S-box as a Boolean map $S : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ which is a bijection, has algebraic degree of 3 on all the output bits, and has linearity of 1/4 and differential potential of 1/4. Also, we presented results with 2, 4 and 8 different leaders and 4 and 8 rounds, respectively, and we listed all the Q-S-boxes that fulfill the predetermined criteria to be an optimal.

3.1 Sample Q-S-box

We apply the methodology for generating cryptographically strong Q-S-boxes to generate a sample Q-S-box with a given quasigroup of order 4 (chosen from the class of 432 non-linear quasigroups) and two different leaders.

Example 1. Let $(Q, *)$ be one non-linear quasigroup and l_1 and l_2 (leaders) be elements from the set $Q = \{0, 1, 2, 3\}$. We present leaders and quasigroup with two-digit binary representation, like:

$l_1 = 1 \to 01$

$l_2 = 3 \to 11$

$*$	0	1	2	3
0	0	2	1	3
1	2	1	3	0
2	1	3	0	2
3	3	0	2	1

$\Longrightarrow$

$*$	00	01	10	11
00	00	10	01	11
01	10	01	11	00
10	01	11	00	10
11	11	00	10	01

We get the first input block of 4 bits in lexicographic ordering (0000), and then the method for producing the output is shown graphically in Figure 1.

Afterwards, we repeat this procedure on all possible 4-bit input values in lexicographic order, and we obtain permutation of order 16, which is our Q-S-box. The corresponding Q-S-box from this example given in its hexadecimal notation is shown in Table 1.

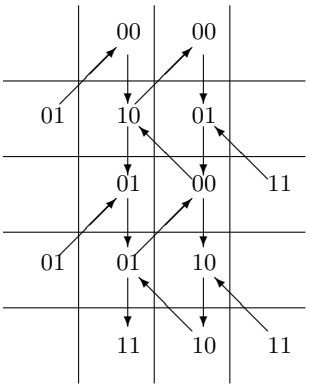

Fig. 1. Four e-transformations that bijectively transforms 4 bits into 4 bits by a quasigroup of order 4.

Table 1. Sample of Q-S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	E	6	C	B	0	1	8	2	D	3	A	F	9	5	4	7

4 Hardware Implementation of Q-S-boxes

We start our hardware implementation by mapping the algorithm for generating optimal Q-S-boxes directly to hardware. Like in the Example 1, we implement a Q-S-Box of $4-th$ degree, i.e. with 4 layers of non-linear mappings. This requires 4×2-bit lookup tables and multiplexers as shown in Figure 2. The resulting Q-S-box has been synthesized with non-linear quasigroup and several different leaders. In all the cases, it has been found to occupy between 38-46 GE. This number is far above the gate counts of existing ciphers. However, the S-box has 4 layers of non-linear mappings inside, which means that it should be possible to implement the S-box in a round-based approach. This is equivalent to implementing only 2 or 1 of the non-linear mapping layers inside the Q-S-box (which we refer to as Q-S-box rounds) and executing it twice or 4 times, respectively, for each substitution layer operation. One might argue that such a solution would require additional registers for the storage of temporary state of Q-S-box outputs. However this is not the case, since the registers are already integral part of the cipher because of the round based operation. Therefore, it can be realized with minimum overhead with respect to a single-round-operation S-box. Figures 3 and 4 show the multi-round Q-S-box (with a single non-linear layer and 4-round operation) alone and with a generic SP-network cipher.

Gate count of such a Q-S-box heavily depends on the overall cipher structure. Its standalone gate count would be huge due to the extra registers and multiplexers required for the multi-round operation. However, the actual (or effective) gate count depends on the reduction it causes on the overall area of the cipher it is part of. Therefore, we chose the PRESENT block cipher as our target platform

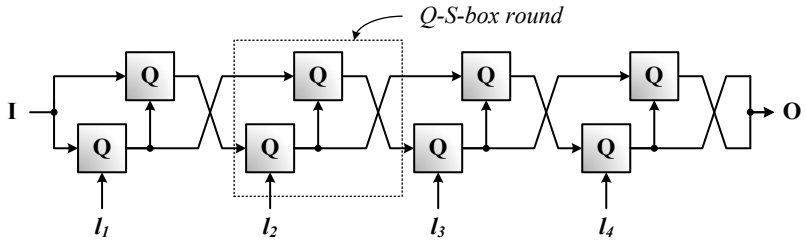

Fig. 2. Direct Q-S-box implementation

and implemented it first with its standard S-boxes in order to determine the overall cipher area and the area of the S-boxes. In the next step, we replaced the PRESENT lookup table based S-box with several different 4-round Q-S-boxes that have the same cryptographic properties as the original PRESENT S-Box. As a result, we have observed that the effective area of the 4-round Q-S-box to be between 15.5 to 19.75 GE. This corresponds to a $30-45\%$ reduction on the S-box area and $12-18\%$ on the overall cipher area, depending on the choice of quasigroup and leaders.

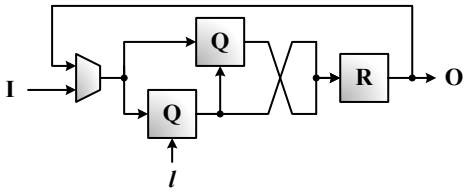

Fig. 3. Multi-round Q-S-box implementation

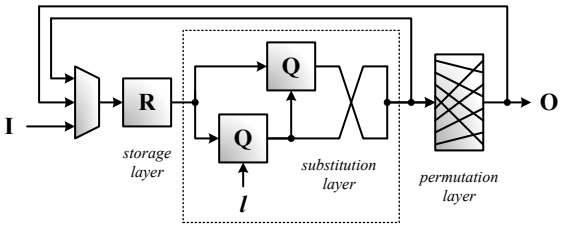

Fig. 4. Multi-round Q-S-box based PRESENT-like cipher implementation

The resultant cipher has 25% of the data throughput of the original cipher. However, compared to a serialized implementation, where the throughput drops to $1/16-th$, it is still an acceptable compromise. In terms of time-area product,

the Q-S-box parallel cipher solution is 3.2 times worse than the original parallel cipher, whereas a serialized implementation is 10 times worse.

We repeated the same procedure to a 2-round Q-S-box based cipher, in the hopes of enhancing the time-area product. However, it turned out that the resultant Q-S-box can achieve only the gate count of the original PRESENT S-box, in the very best case, with zero overall improvement in the cipher area. In similar way, we implemented Q-S-boxes with only a single 4×2 lookup table, with which we could reduce the area by another $2-3\%$ w.r.t. a full 4×4 lookup table based approach, in certain cases. But we deem this much reduction not practical considering the further throughput reduction by a factor 2.

Another argument is that the resultant cipher does not any more hold the cryptographic properties of the original PRESENT cipher. With a completely different substitution layer, all of its cryptanalysis have to be done from scratch. This, in fact, is true. However, it should be noted that we have chosen PRESENT only as the development and proof-of-concept platform for our Q-S-box proposal. In a completely new cipher, designed from scratch with properties and hardware structures of Q-S-boxes taken into consideration, it will be possible to achieve the same security levels as the PRESENT cipher, and possibly even more reduction in gate count. As a matter of fact, applying the Q-S-boxes in another existing cipher might result in completely different area reduction figures. Another active research we still work on is the ability to represent any given S-box as a Q-S-boxes, which would solve all the above mentioned problems.

5 Conclusion and Future Work

Q-S-boxes offer the option to not only iteratively reuse the same circuit to implement several different strong 4-bit S-boxes, but they can also be serialized down to bit level, leading to S-box implementations below 10 GE. With Q-S-boxes we can get over the 40% of area reduction w.r.t. a lookup table based implementation, and also over the 16% of area reduction w.r.t. a parallel implementation of PRESENT.

As the future work, we will focus on two different directions: The first is to extend our current approach to S-boxes of any size, while the second, also the harder, is to express any given S-box as a Q-S-box. This might require application of different techniques such as application of Gröbner bases and brute-force search (possibly with hardware acceleration), all of which will help to extend the application of quasigroups into different fields of research.

References

1. Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., Uhsadel L.: A Survey of Lightweight-Cryptography Implementations. IEEE Des. Test vol. 24, no. 6, 522–533 (2007)
2. Finkenzeller, K.: RFID Handbook. John Wiley, Chichester (2003)

3. Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: AES Implementation on a Grain of Sand. Information Security IEEE Proc. vol. 152, no. 1, 13–20 (2005)
4. Rolfes, C., Poschmann, A., Leander, G., Paar, C.: Ultra-Lightweight Implementations for Smart Devices - Security for 1000 Gate Equivalents. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 89103. Springer, Heidelberg (2008)
5. ONeill, M.: Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In: Proceedings of RFIDSec (2008)
6. Yoshida, H., Watanabe, D., Okeya, K., Kitahara, J., Wu, H., Kucuk, O., Preneel, B.: MAME: A compression function with reduced hardware requirements. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 148165. Springer, Heidelberg (2007)
7. Lim, C.H., Korkishko, T. :mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In Song, J., Kwon, T. and Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243258. Springer, Heidelberg (2005)
8. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New Lighweight DES Variants. In Biryukov, A. (ed.) FSE 2007, LNCS, vol. 4593, pp. 196-210. Springer, Heidelberg (2007)
9. Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin. Y., Vikkelsoe, C.:PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
10. Engels, D., Fan, X., Gong, G., Hu, H., Smith, E.:Ultra-Lightweight Cryptography for Low-Cost RFID Tags: Hummingbird Algorithm and Protocol. Technical report, `http://cacr.uwaterloo.ca/techreports/2009/cacr2009-29.pdf`
11. Leander, G., Poschmann, A.: On the Classification of 4 Bit S-Boxes. In: Carlet, C., Sunar, B. (eds.) WAIFI 2007, LNCS vol. 4547, pp. 159–176. Springer, Heidelberg (2007)
12. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED Block Cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol 6917, pp. 326–341. Springer, Heidelberg (2011)
13. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A Lightweight, Versatile Block Cipher. ECRYPT Workshop on Lightweight Cryptography 2011
14. Gligoroski, D., Dimitrova, V., Markovski, S.: Quasigroups as Boolean Functions, Their Equation Systems and Gröbner Bases. In: Sala, M., Mora, T., Perret, L., Sakata, S., Traverso, C. (eds.). Gröbner Bases, Coding, and Cryptography, pp.415-420. Springer, Heidelberg (2009)
15. Mihajloska, H., Gligoroski, D.,: Construction of Optimal 4-bit S-boxes by Quasigroups of Order 4. SECURWARE 2012, The Sixth International Conference on Emerging Security Information, Systems and Technologies, Rome, Italy (2012)(Best paper award)
16. Chabaud, F., Vaudenay, S.: Links Between Differential and Linear Cryptanalysis. In Advances in Cryptology - EUROCRYPT94. LNCS, vol. 950, pp. 356-365. Springer, Heidelberg(1995)