On the Construction of Hardware-friendly 4×4 and 5×5 S-boxes

Stjepan Picek   Bohan Yang   Vladimir Rozic   Nele Mentens

KU Leuven ESAT/COSIC and iMinds, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium

28th October 2016

Outline

Lightweight cryptography

- Besides the security, in lightweight cryptography we are also interested to make the ciphers as small and as efficient as possible.
- Therefore, some of goals are area-efficient ciphers, energy-efficient ciphers, and latency-efficient ciphers.
- Furthermore, large part of ciphers is realized as SPN structures.
- IN SPNs, the S-box part is often the most expensive one!

Motivation

- We concentrate on 4×4 and 5×5 sizes of S-boxes.
- We explore how to obtain S-boxes that are hardware-friendly but have optimal cryptographic properties.
- Search space is too large to conduct an exhaustive search when accounting the cost of the evaluation part.
- Therefore, we explore how efficient is heuristics for such a task.

Cryptographic Properties

- Bijectivity.
- Nonlinearity.
- δ-uniformity.
- Algebraic degree.
- Number of fixed points.

Static and Dynamic Power

- Dynamic power consumption originates from the switching activity of the circuit.
- Static power consumption is caused by subthreshold currents and gate leakage.
- Static power consumption is constant over time and does not depend on the clock frequency or the switching activity.
- In older technology nodes the dynamic power consumption was dominant in the total power consumption and the static power consumption was negligible.
- With smaller technology nodes, the relative contribution of the static leakage power consumption has increased.

4×4 Size

- Around 2^{44} bijective S-boxes.
- 16 optimal, not affine-equivalent classes.
- We are interested only in optimal S-boxes

$$S_a(x) = B(S_b(A(x) \oplus \vec{a})) \oplus \vec{b}, \quad (1)$$

- In general, it is possible to conduct exhaustive search (when considering only cryptographic properties).
- There are also other classifications.

5×5 Size

- Impossible to conduct exhaustive search.
- We know only classification of APN functions of dimension 5.
- Here, S-boxes with suboptimal cryptographic properties are also used.

Simulation Setup

- Clock frequency of 10 MHz and NANGATE 45 open cell library.
- Generate S-boxes in a lookup table style.
- Matlab script to generate the HDL description of the S-box.
- Synopsys Design Compiler to produce the gate-level netlist and the delay file.
- Test-bench that goes through all possible $n \times (n-1)$ input transitions of the S-box.
- Modelsim SE PLUS 6.6d to simulate the wave file containing the switching activity of all nodes.
- Design Compiler is used to estimate the power consumption.

Power/area evaluation

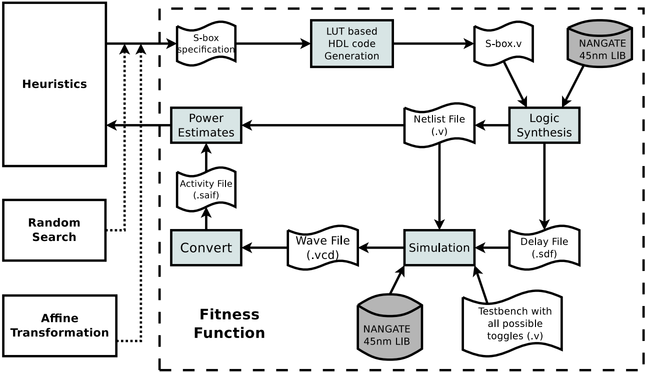
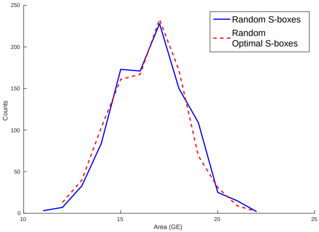


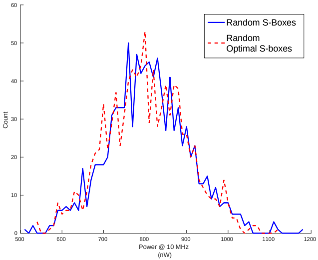
Figure: Simulation setup for the generation/evaluation of S-boxes.

Random Search

- Create random S-boxes as permutations of values between 0 and $2^n - 1$ and check the results in terms of area and power.
- When evaluating only the optimal S-boxes, our results show that the power consumption is higher than 550 nW.
- In terms of area, the optimal S-boxes obtained through random search have an area larger than 20GE.



(a) Area results for randomly chosen 4×4 S-boxes.



(b) Area results for randomly chosen 4×4 S-boxes.

Figure: 1 000 random and optimal 4×4 S-boxes

Genetic algorithms

- Metaheuristics intended to provide good results on a wide range of problems.
- Black-box technique.
- Natural representation for S-boxes is permutation-based.
- Natural representation for invertible matrices and constants is bitstring-based.

Pseudocode for population-based metaheuristics

Input : *Parameters of the algorithm*
Output : *Optimal solution set*
$t \leftarrow 0$
$P(0) \leftarrow$ *CreateInitialPopulation*
while *TerminationCriterion* **do**
    $t \leftarrow t + 1$
    $P'(t) \leftarrow$ *SelectMechanism* $(P(t-1))$
    $P(t) \leftarrow$ *VariationOperators*$(P'(t))$
end while
Return OptimalSolutionSet(P)

Objective Functions

- Maximize the value of:

$$fitness = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points). \quad (2)$$

- Two-stage evaluation.

Objective Functions

- Maximize the value of:

$$fitness = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points). \qquad (2)$$

- Two-stage evaluation.
- Only optimal S-boxes are evaluated for power/area.

Objective Functions

- Maximize the value of:

$$fitness = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points). \qquad (2)$$

- Two-stage evaluation.
- Only optimal S-boxes are evaluated for power/area.
- The whole population is sent at the same time for evaluation.

4 × 4 Size

Table: Best evolved 4 × 4 S-boxes

Area results					
With fixed points	13, 1, 3, 11, 12, 2, 7, 10, 0, 5, 8, 9, 4, 6, 14, 15				
Area:	14.33*GE*				
Without fixed points	7, 10, 11, 8, 5, 3, 1, 9, 6, 2 15, 0, 4, 14, 12, 13				
Area:	13.33*GE*				
Power results					
With fixed points	3, 1, 2, 10, 14, 5, 7, 15, 4, 6, 0, 11, 13, 12, 8, 9				
Dynamic Power:	237.16*nW*	Leakage Power:	297.52*nW*	Area:	14.67*GE*
Without fixed points	13, 5, 10, 4, 7, 1, 2, 0, 14, 6, 8, 12, 15, 3, 9, 11				
Dynamic Power:	206.1*nW*	Leakage Power:	240.73*nW*	Area:	12.67*GE*

Involutive S-boxes

- 2 027 025 involutive S-boxes with 0 fixed points, 16 216 200 with 2 fixed points, 18 918 900 with 4 fixed points, 7 567 570 with 6 fixed points.
- Out of that, there are ≈ 3 000 000 optimal S-boxes with various number of fixed points.

Involutive S-boxes

- 2 027 025 involutive S-boxes with 0 fixed points, 16 216 200 with 2 fixed points, 18 918 900 with 4 fixed points, 7 567 570 with 6 fixed points.
- Out of that, there are ≈ 3 000 000 optimal S-boxes with various number of fixed points.
- For S-boxes with 4 fixed points the best result for area is $13GE$ while the best result for power is an S-box with a dynamic power of $201.8418nW$ and a leakage power of $255.1868nW$.

Involutive S-boxes

- 2 027 025 involutive S-boxes with 0 fixed points, 16 216 200 with 2 fixed points, 18 918 900 with 4 fixed points, 7 567 570 with 6 fixed points.
- Out of that, there are ≈ 3 000 000 optimal S-boxes with various number of fixed points.
- For S-boxes with 4 fixed points the best result for area is $13GE$ while the best result for power is an S-box with a dynamic power of $201.8418nW$ and a leakage power of $255.1868nW$.
- For optimal involutive S-boxes with 6 fixed points the best result for area is $15GE$ and the best result for power is an S-box with a dynamic power of $223.3748nW$ and a leakage power of $293.5608nW$.

5×5 Size

Table: Best evolved 5×5 S-boxes

		Area results			
With fixed points					
Area:	39.33GE				
Without fixed p.					
Area:	38GE				
		Power results			
With fixed points					
Dynamic Power:	801.8934nW	Leakage Power:	777.7131nW	Area:	39.67GE
Without fixed p.					
Dynamic Power:	734.7164nW	Leakage Power:	754.2006nW	Area:	39.33GE

Affine Transformation Results

Table: Best evolved 5 × 5 S-boxes, affine transformations

Keccak					
Dynamic Power:	488.6914nW	Leakage Power:	496.4189nW	Area:	26GE
PRIMATEs					
Dynamic Power:	751.4109nW	Leakage Power:	723.7496nW	Area:	37GE
APN S-box					
Dynamic Power:	913.5057nW	Leakage Power:	942.5685nW	Area:	48GE

Discussion

- For 4×4 size, the results are very good.
- For 5×5 size, the results are somewhat worse when looking for S-boxes, but again good when investigating affine transformations.

Discussion

- For 4×4 size, the results are very good.
- For 5×5 size, the results are somewhat worse when looking for S-boxes, but again good when investigating affine transformations.
- Difficult to compare with other approaches.
- LUT based S-box are maybe not the best way how to represent/implement, but as far as we are aware, they are the most fair approach.
- Furthermore, synthesis tools do translate LUTs into combinatorial circuits, only not optimal ones.

Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.

Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.
- Interesting approach would be to evolve only involutive S-boxes.

Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.
- Interesting approach would be to evolve only involutive S-boxes.
- Currently, we are running an exhaustive search on optimal involutive S-boxes, but for now we obtained no better results than with GA (we found S-box with the same area, but worse power consumption).

Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.

Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.

Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
- Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.

Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
- Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.
- Naturally, our results are as good as the synthesis tools allow.

Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
- Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.
- Naturally, our results are as good as the synthesis tools allow.
- Good results in naive combinatorial circuit implementation can only improve with smarter implementations, while vice versa does not necessarily hold.

Thank You for Your attention.
Questions?