Several Masked Implementations of the Boyar Peralta AES S-Box

Ashrujit Ghoshal
Indian Institute of Technology, Kharagpur
Thomas De Cnudde
KU Leuven, imec-COSIC, Belgium

Introduction

Power Analysis

Exploit information from the correlation between the instantaneous power consumption of the device and the intermediate results of the cryptographic algorithm.

Variants:

1. Simple Power Analysis (SPA)
2. Differential Power Analysis (DPA)
    a. Difference Of Means (DoM)
    b. Correlation Power Analysis (CPA)
    c. Templates

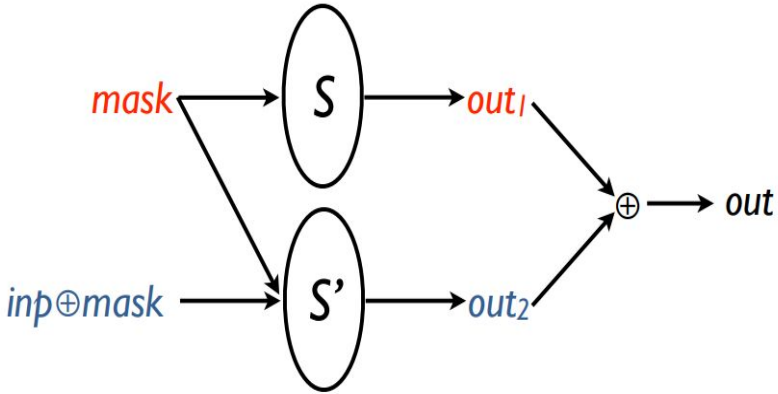
Introduction

DPA Countermeasures

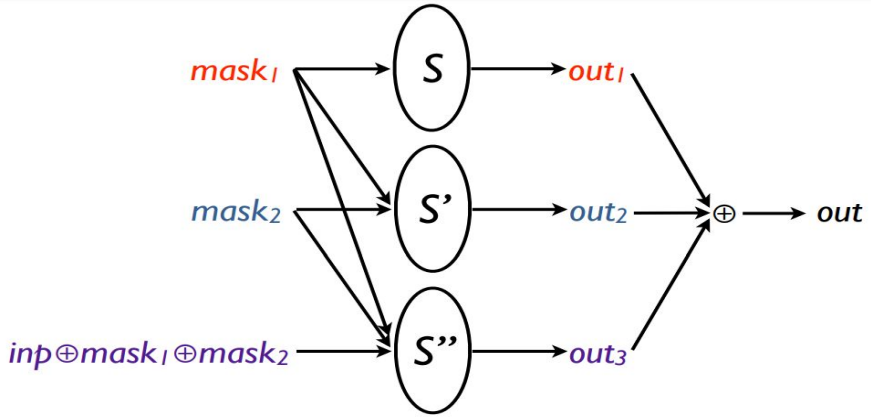
1. Introducing random delays or dummy operations(Not Provably Secure)
2. Masking(provably secure)
3. Leakage resilient crypto (limits encryptions per key)
4. Hiding (specific logic styles, like WDDL)

Introduction

Masking



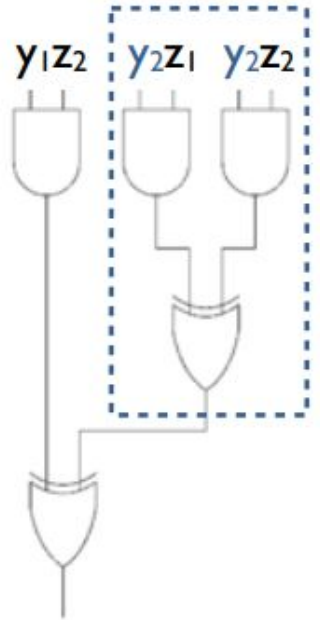
First-order Masking



Second-order Masking

Introduction
Masking

y_2	z_1	z_2	#AND	#XOR	#TOTAL
0 → 1	0	0	0	0	0
1 → 0	0	0	0	0	0
0 → 1	1	1	2	1	3
1 → 0	1	1	2	1	3
0 → 1	1	0	1	1	2
1 → 0	1	0	1	1	2
0 → 1	0	1	1	1	2
1 → 0	0	1	1	1	2



Assume y_2 arrives late

Threshold Implementations

Boolean Masking Scheme based on Secret Sharing and Multiparty Computation

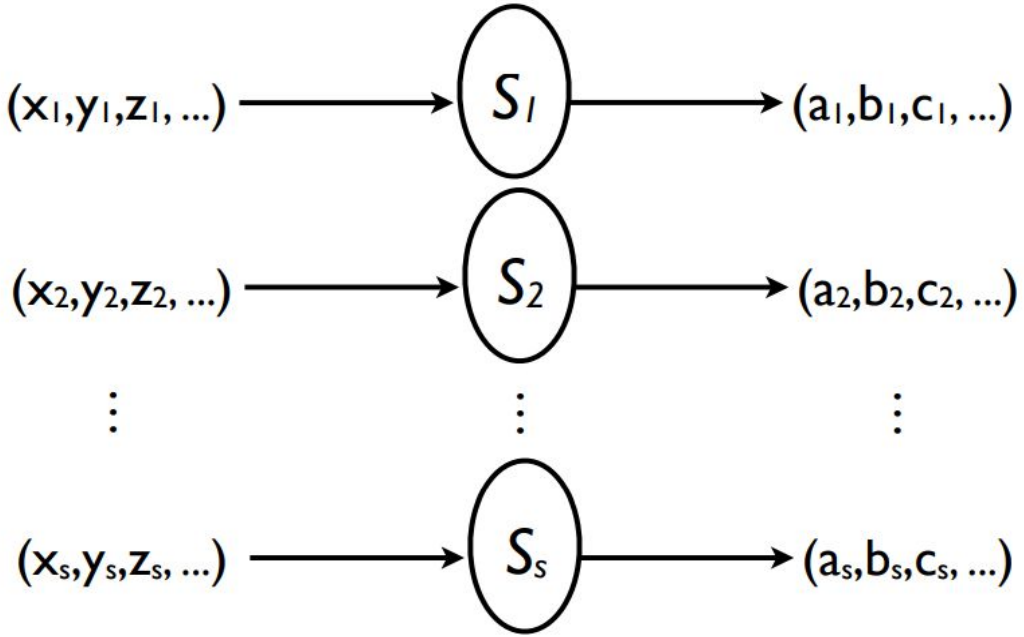
Pros:

1. Security in a circuit with glitches
2. Efficient in Hardware
3. Any HW technology

Con:

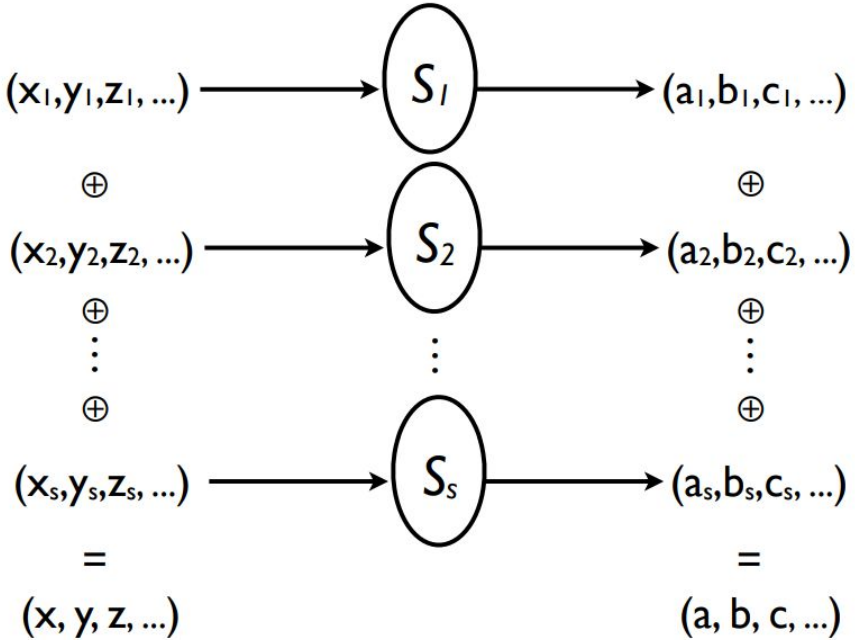
Non-linear functions are challenging

Threshold Implementations



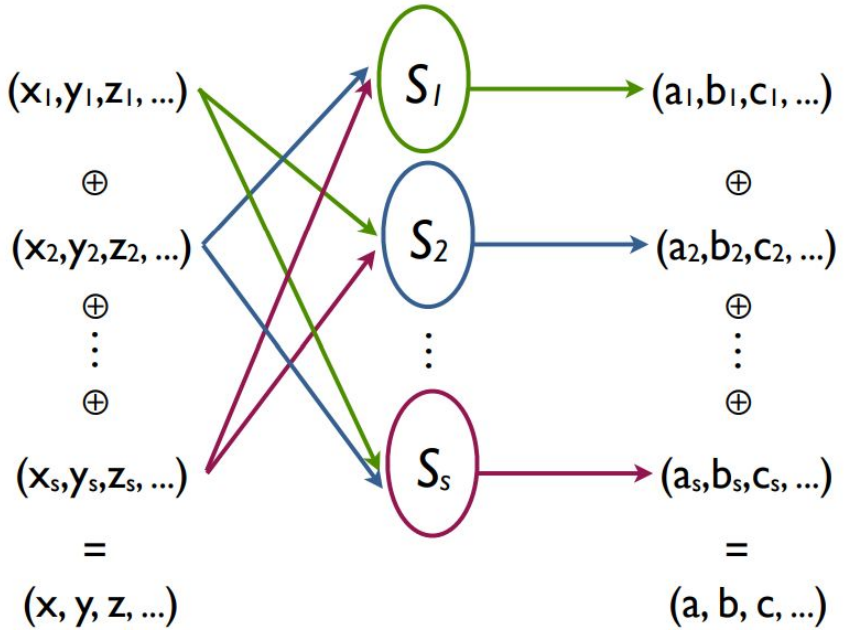
3 Properties

Threshold Implementations



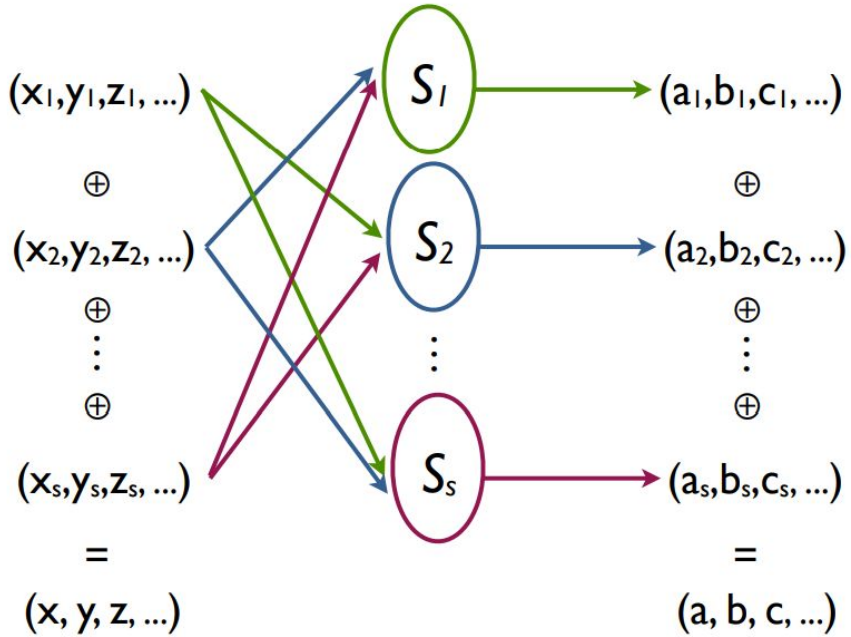
1. Correctness

Threshold Implementations



2. Non-completeness

Threshold Implementations



3. Uniformity

Threshold Implementations

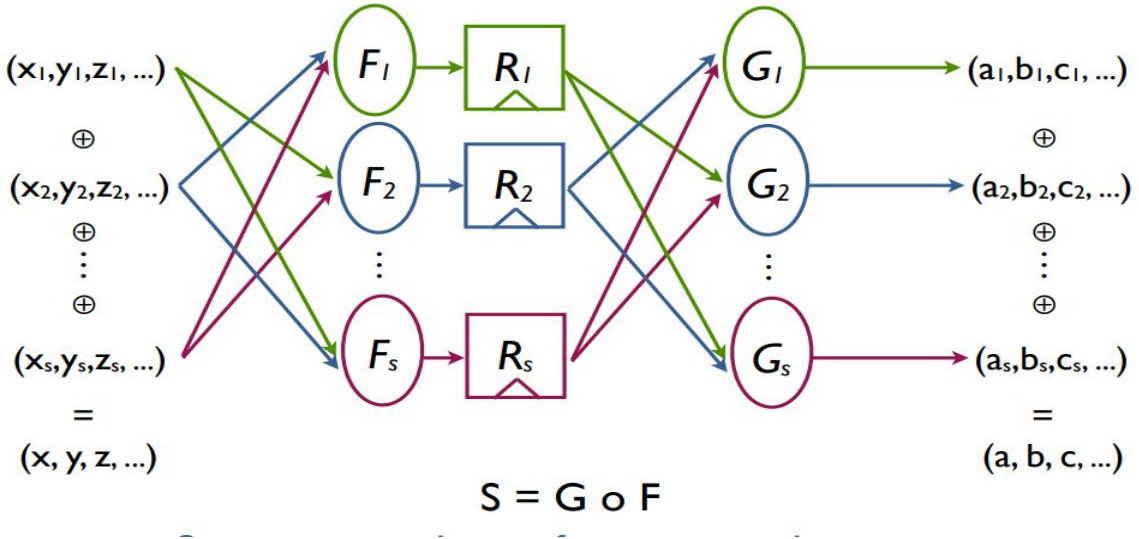
Uniformity

If uniformity can not be achieved during S_i calculation:

- Apply re-masking
- Increase the number of shares

Threshold Implementations

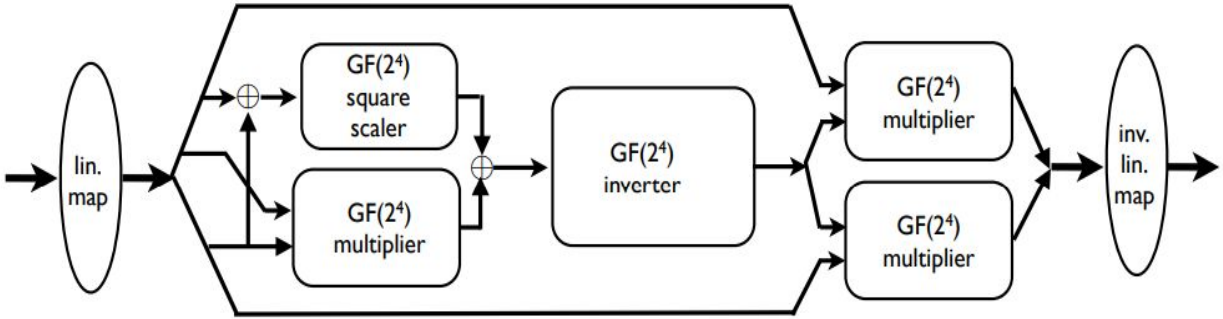
Decomposition



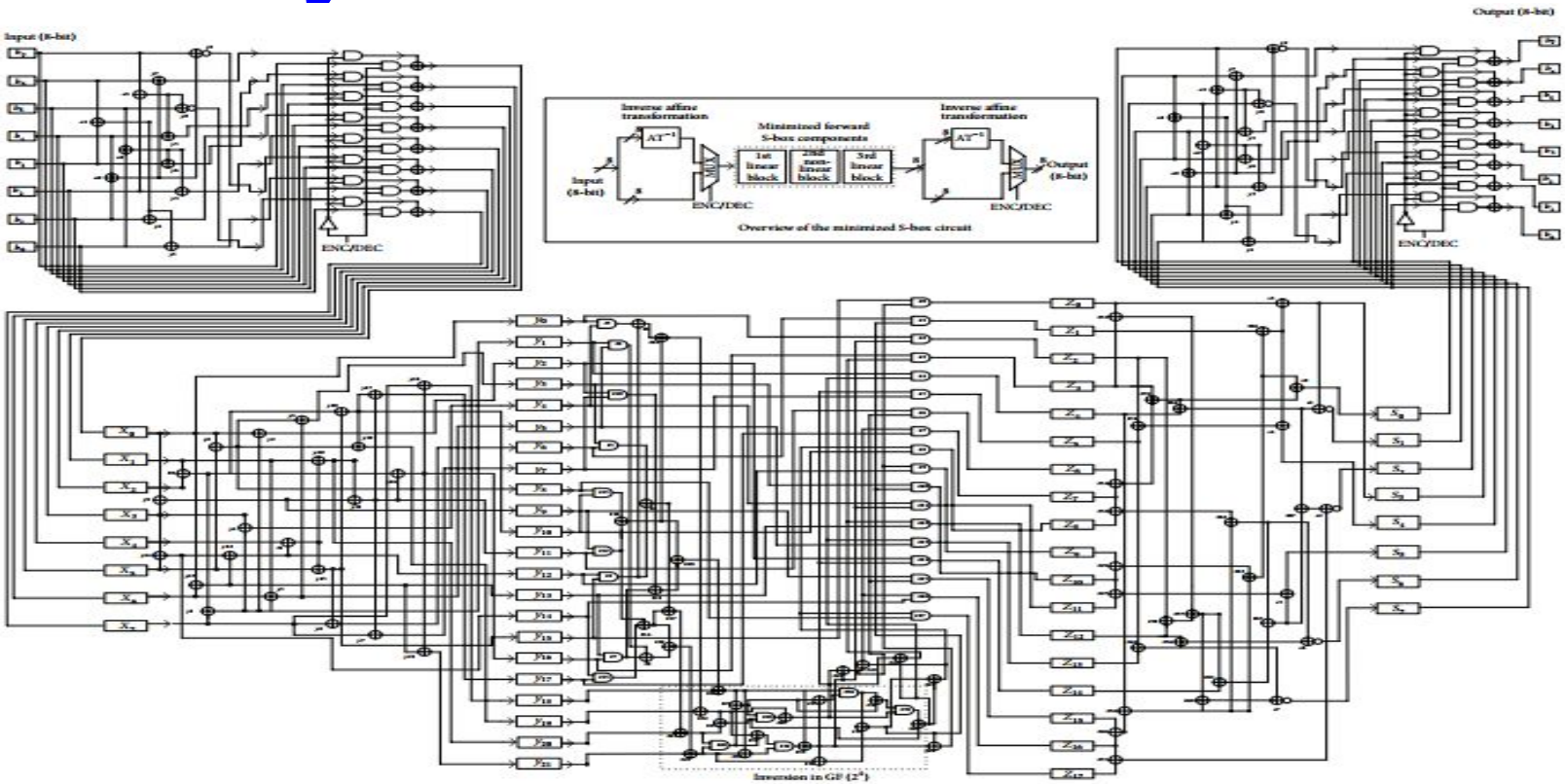
Separate non-linear functions with registers

TI on AES S-Box

The Canright S-box has been used predominantly for TI of AES, e.g. by Moradi [3] and by Bilgin [4].



The Boyar Peralta AES S-Box



Overview of the minimized S-box circuit

The Boyar Peralta AES S-Box

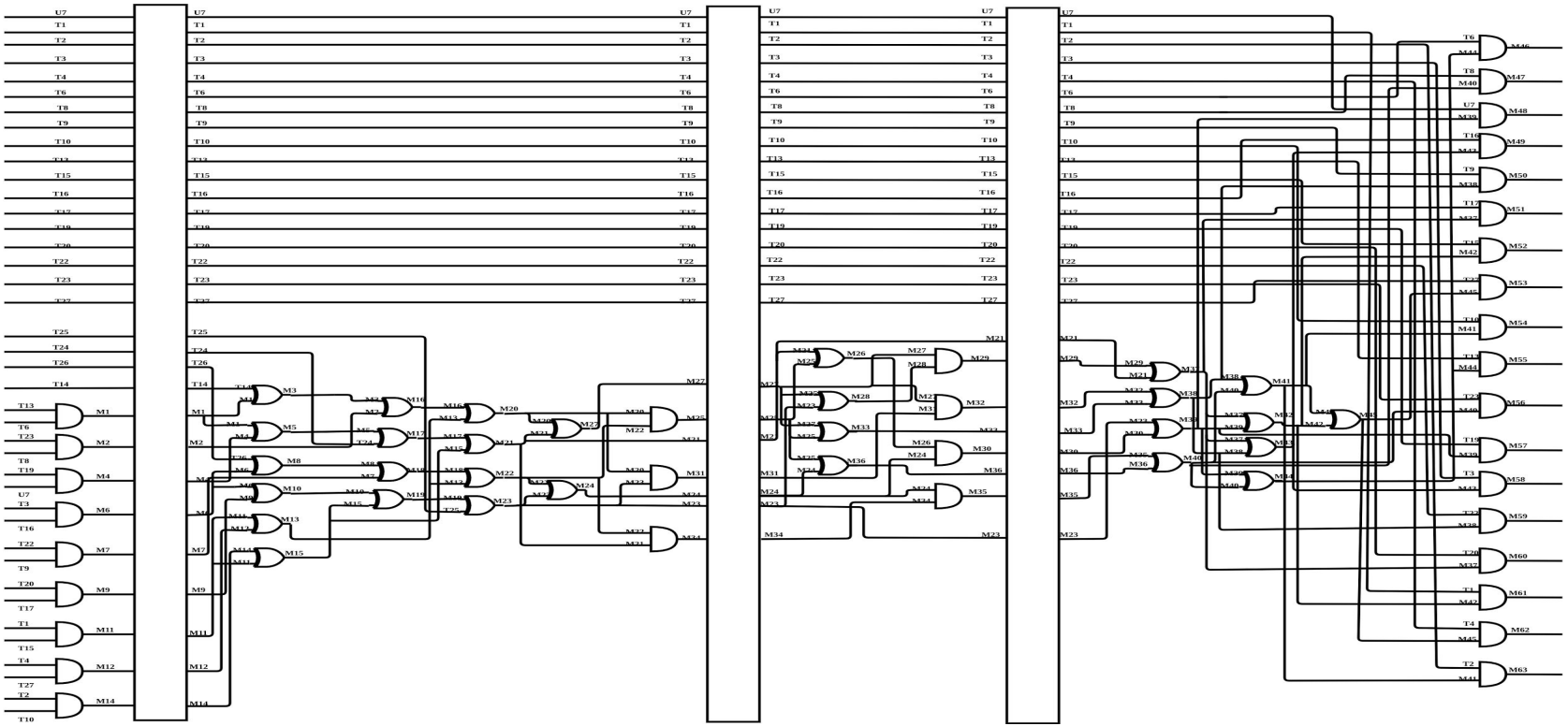
- The circuit can be divided into 3 stages: a linear layer, followed by a non-linear layer followed by a linear layer.
- The circuit has a total of 34 AND gates and 94 linear operations(XOR and XNOR)
- 2-input AND gates are the only non-linearity present in the circuit
- AES S-Box with smallest known circuit depth of 16.

Towards a first-order TI

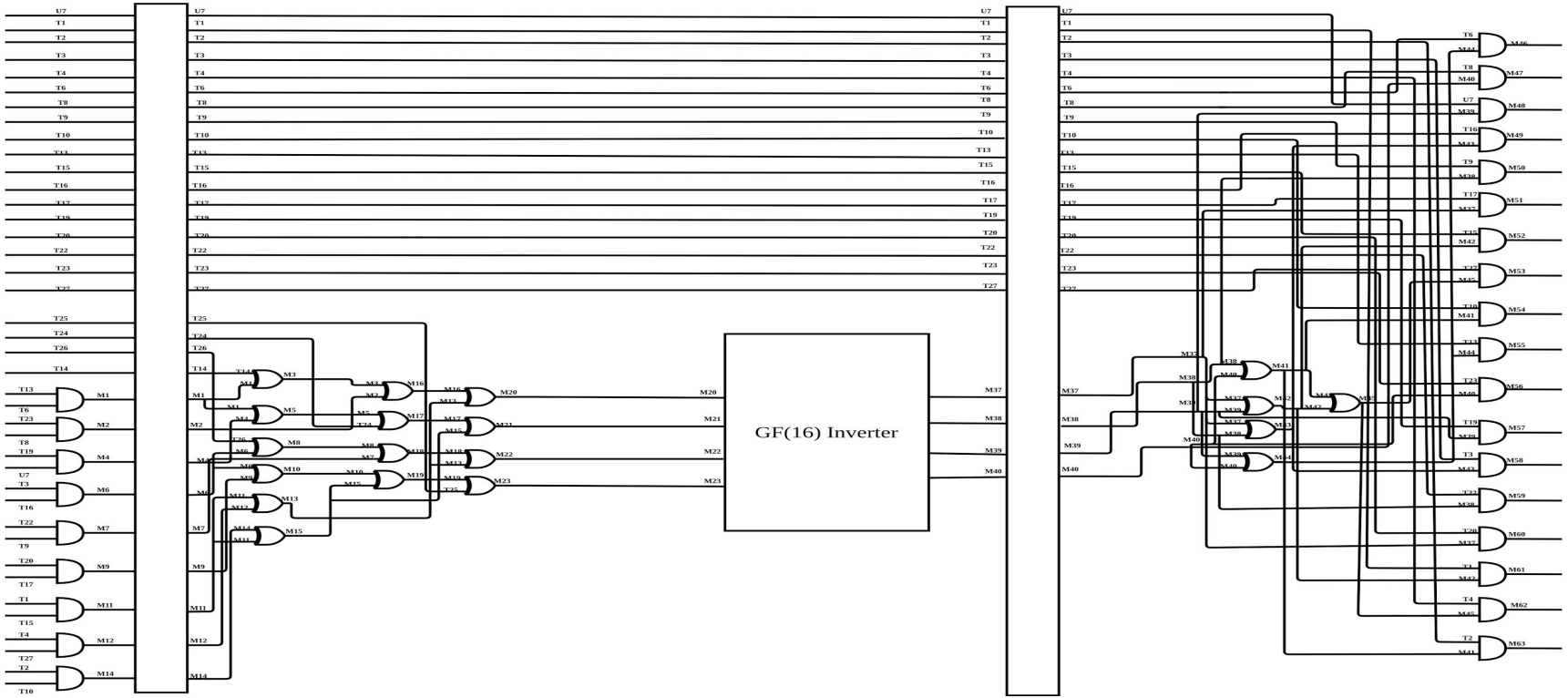
Important Points:

1. The sharing for the linear operations can be done trivially by duplicating the linear operations for each share.
2. To preserve non-completeness, divide the non-linear part of the circuit into stages or levels and insert registers after each level of non-linear operation.

Division of the Non-linear Layer into Stages



Division of Non-linear Layer into Stages after isolating an inverter



First Order TI Designs

We designed the following first order TIs for the Boyar-Peralta AES S-Box:

1. with 4 shares and no randomness
2. with 3 shares and 68 bits randomness
3. with 3 shares and 34 bits of randomness
4. with 3 shares and using sharing with $s_{in} = 5$ and $s_{out} = 5$ for a GF(2^4) inverter
5. with 3 shares and using sharing with $s_{in} = 4$ and $s_{out} = 4$ for a GF(2^4) inverter

Number 1: Threshold Implementation with 4 shares

1. A Uniform 4-to-4 sharing of the 2-input AND gate

- For the non-linear 1-bit multiplication we first used this following uniform 4-to-4 sharing which is a novel modification of the 4-to-3 uniform and has been used in [3].
- Takes 4 clock cycles
- First TI of Full AES S-Box that uses no randomness.

$$A = X.Y$$
$$X = x_1 \oplus x_2 \oplus x_3 \oplus x_4$$
$$Y = y_1 \oplus y_2 \oplus y_3 \oplus y_4$$
$$A = a_1 \oplus a_2 \oplus a_3 \oplus a_4$$
$$a_1 = (x_2 \oplus x_3 \oplus x_4).(y_2 \oplus y_3) \oplus y_3$$
$$a_2 = ((x_1 \oplus x_3).(y_1 \oplus y_4)) \oplus (x_1.y_3) \oplus x_4$$
$$a_3 = (x_2 \oplus x_4).(y_1 \oplus y_4) \oplus x_4 \oplus y_4$$
$$a_4 = (x_1.y_2) \oplus y_3$$

Threshold Implementation with 3 shares

- Our next aim was to reduce the complexity of the circuit and reducing the number of shares.
- For this we have a trade-off and need to introduce some randomness to achieve uniformity in sharing.
- We needed to focus on reducing the number of shares for the non-linear part ie. the 2-input AND gate.

Number 2: Threshold Implementation with 3 shares

2. Sharing with 2 units of randomness for each AND gate

- For a 1-bit multiplier we use the following remasking for uniform sharing as shown in [4].
- It uses 2-bits of randomness per AND gate.
- Our total S-Box circuit required 2x34=68 bits of randomness
- It requires 4 clock cycles.

$$A = X.Y$$
$$X = x_1 \oplus x_2 \oplus x_3$$
$$Y = y_1 \oplus y_2 \oplus y_3$$
$$A = a_1 \oplus a_2 \oplus a_3$$
$$a_1 = (x_2.y_2) \oplus (x_2.y_3) \oplus (x_3.y_2) \oplus \mathbf{r_1} \oplus \mathbf{r_2}$$
$$a_2 = (x_3.y_3) \oplus (x_1.y_3) \oplus (x_3.y_1) \oplus \mathbf{r_2}$$
$$a_3 = (x_1.y_1) \oplus (x_1.y_2) \oplus (x_2.y_1) \oplus \mathbf{r_1}$$

$\mathbf{r_1}$ and $\mathbf{r_2}$ are 2 units of randomness

Number 3: Threshold Implementation with 3 shares

3. Sharing with 1 unit of randomness for each AND gate

- We use the following sharing that is uniform and requires one bit of randomness per AND gate.
- It is referred to as virtual sharing and was used in [7].
- Our total S-Box circuit required 34 bits of randomness
- It requires 4 clock cycles.

$$A = X.Y$$
$$X = x_1 \oplus x_2 \oplus x_3$$
$$Y = y_1 \oplus y_2 \oplus y_3$$
$$A = a_1 \oplus a_2 \oplus a_3$$
$$a_1 = (x_2.y_2) \oplus (x_2.y_3) \oplus (x_3.y_2) \oplus \mathbf{r}$$
$$a_2 = (x_3.y_3) \oplus (x_1.y_3) \oplus (x_3.y_1) \oplus (x_1.\mathbf{r}) \oplus (y_1.\mathbf{r})$$
$$a_3 = (x_1.y_1) \oplus (x_1.y_2) \oplus (x_2.y_1) \oplus (x_1.\mathbf{r}) \oplus (y_1.\mathbf{r}) \oplus \mathbf{r}$$

r is a unit of randomness

Number 4: Threshold Implementation with 3 shares

4. Reducing Number of clock cycles: Sharing the inverter

- We isolate an inverter in GF(2^4).
- There is a 5-to-5 uniform sharing of the inverter mentioned in [3] that we used for sharing the inverter.
- Linear parts do not need uniform inputs.

Threshold Implementation with 3 shares

Reducing Number of clock cycles: Sharing the inverter

- For the AND gates in Stage 3, the inputs are remasked to increase the number of output shares of the inverter.
- Hence randomness is only required for increasing and decreasing the number of shares
- For increasing number of shares from 3 to 5 , $4 \times 4 = 16$ bits of randomness is required and for decreasing the number of shares from 5 to 3, $4 \times 2 = 8$ bits of randomness is required. Therefore 24 bits of randomness is required in total.
- It requires 3 clock cycles.

Number 5: Threshold Implementation with 3 shares

5. Further Reducing Randomness in sharing the inverter

- There is a 4-to-4 non-uniform sharing of the inverter mentioned in [3] that we used in this case.

- Since the output of the inverter is reshared anyway to decrease the number of shares, we do not need uniformity in output of the inverter.

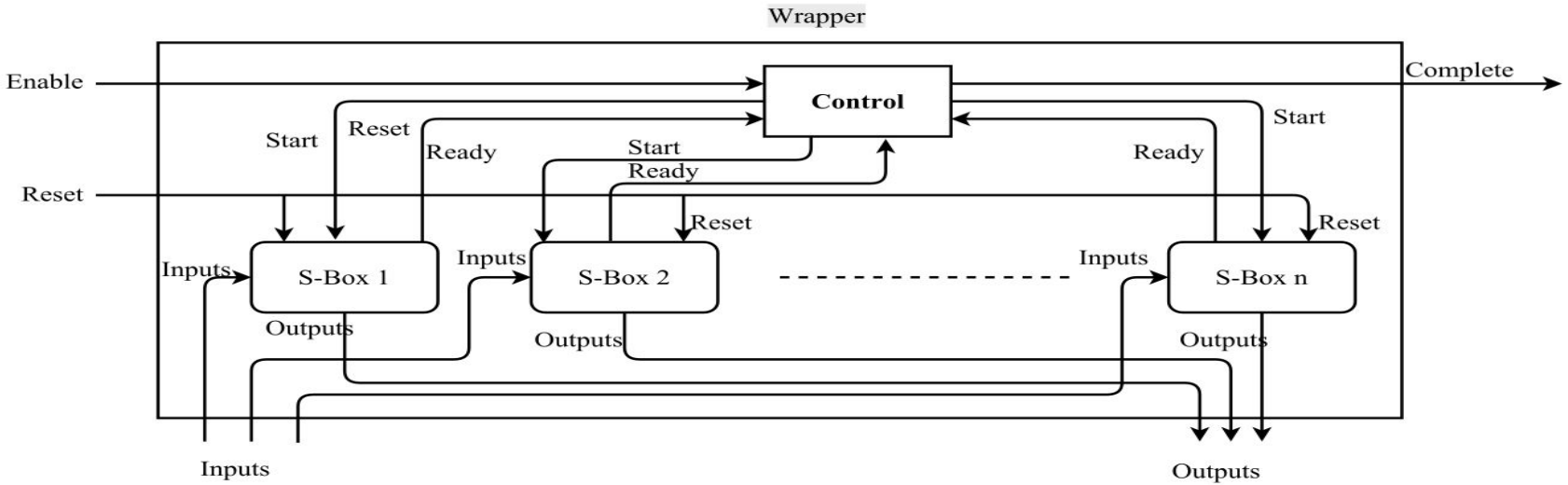
Threshold Implementation with 3 shares

Further Reducing Randomness in sharing the inverter

- In this case too, randomness is only required for increasing and decreasing the number of shares
- For increasing number of shares from 3 to 4 , $3 \times 4 = 12$ bits of randomness is required and for decreasing the number of shares from 4 to 3, $4 \times 2 = 8$ bits of randomness is required. Therefore 20 bits of randomness is required in total.
- It requires 3 clock cycles.

Side Channel Evaluation

Structure of circuit for sequential evaluation of the S-Boxes



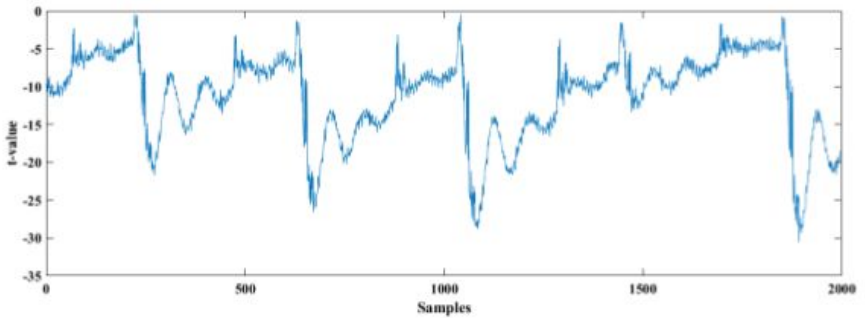
Side Channel Evaluation

Testing Methodology

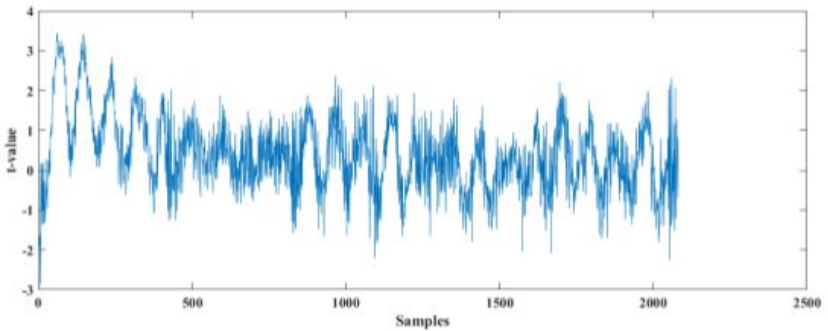
- We used Test Vector Leakage Assessment(TVLA) method
- The fix class of the leakage detection is chosen as the zero input in all our evaluations.
- first turn off the PRNG to switch off the masking countermeasure to confirm that the experimental setup is sound (we can detect leakage).
- We then proceed by turning on the PRNG

Side Channel Evaluation

Obtained Traces



(a) 5K Traces, Masks Off



(b) 10M Traces, Masks On

First Order leakage detection test for the S-Box with 3 shares and using a 4-to-4 sharing for a GF(2^4) inverter

Tradeoff between area, randomness and clock cycles

Implementation	Total Cell Area(In Gate Equivalents)	Bits of randomness required per round	No of Clock Cycles required
Unprotected	269	0	1
4 share	4609	0	4
3 share, 2 bits of randomness per AND gate	3630	68	4
3 share, 1 bit of randomness per AND gate	3798	34	4
3 share, inverter 5 to 5 sharing	3344	24	3
3 share, inverter 4 to 4 sharing	2913	20	3

The results of area have been obtained using Synopsys 2013.12 and NanGate 45nm Open Cell Library.

Comparision with existing implementations

Implementation	Total Cell Area(In Gate Equivalents)	Bits of randomness required per round	No of Clock Cycles required
Our Implementation	2913	20	3
Canright's S-Box in [3]	3708	44	3
Canright's S-Box in [4]	4244	48	4
Canright's S-Box in [9]	2835	32	3
Canright's S-Box in [8](First order TI)	1977	54	6

Conclusion

We can summarize the comparison of our implementations with related implementations as follows:

- We achieve an implementation that consumes no randomness.
- Two of our implementations, which use the sharing for inversion in GF(2^4), take 3 clock cycles, which is faster than implementations in [4,8]
- Our implementation that uses the 4-sharing of an inverter needs the same number of clock cycles as the smallest one in [9], while consuming less randomness for an increase in area of only 2.75%.
- The S-Box in [15] is the smallest known TI of the AES S-Box. Our implementation is 47% larger in comparison but we obtain a 63% reduction in randomness and 50% reduction in number of clock cycles required.

Future Work

These are some possible directions of future research on this topic:

- Starting from a masked Canright AES S-Box, using the optimizations mentioned in [2], arriving at a small and secure implementation of the Boyar-Peralta S-Box.
- Masking the Boyar Peralta S-Box with d+1 shares as shown in [5,6].
- Designing circuits for this S-Box with higher-order security levels, as a determined adversary can still break the first-order masking scheme with a second order attack.

References

1. Canright D. (2005) A Very Compact S-Box for AES. In: Rao J.R., Sunar B. (eds) Cryptographic Hardware and Embedded Systems CHES 2005. CHES 2005. Lecture Notes in Computer Science, vol 3659. Springer, Berlin, Heidelberg
2. Boyar J., Peralta R. (2012) A Small Depth-16 Circuit for the AES S-Box. In: Gritzalis D., Furnell S., Theoharidou M. (eds) Information Security and Privacy Research. SEC 2012. IFIP Advances in Information and Communication Technology, vol 376. Springer, Berlin, Heidelberg
3. Bilgin B., Gierlichs B., Nikova S., Nikov V., Rijmen V. (2014) A More Efficient AES Threshold Implementation. In: Pointcheval D., Vergnaud D. (eds) Progress in Cryptology AFRICACRYPT 2014. AFRICACRYPT 2014. Lecture Notes in Computer Science, vol 8469. Springer, Cham.
4. Moradi A., Poschmann A., Ling S., Paar C., Wang H. (2011) Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In: Paterson K.G. (eds) Advances in Cryptology EUROCRYPT 2011. EUROCRYPT 2011. Lecture Notes in Computer Science, vol 6632. Springer, Berlin, Heidelberg
5. Reparaz O., Bilgin B., Nikova S., Gierlichs B., Verbauwhede I. (2015) Consolidating Masking Schemes. In: Gennaro R., Robshaw M. (eds) Advances in Cryptology – CRYPTO 2015. CRYPTO 2015. Lecture Notes in Computer Science, vol 9215. Springer, Berlin, Heidelberg
6. Hannes Gross, Stefan Mangard, and Thomas Korak. 2016. Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. In Proceedings of the 2016 ACM Workshop on Theory of Implementation Security (TIS '16). ACM, New York, NY, USA, 3-3.
7. Bilgin B., Nikova S., Nikov V., Rijmen V., Sttz G. (2012) Threshold Implementations of All 3 3 and 4 4 S-Boxes. In: Prouff E., Schaumont P. (eds) Cryptographic Hardware and Embedded Systems CHES 2012. CHES 2012. Lecture Notes in Computer Science, vol 7428. Springer, Berlin, Heidelberg
8. De Cnudde, T., Reparaz, O., Bilgin, B., Nikova, S., Nikov, V., Rijmen, V.: Masking AES with d+1 Shares in Hardware. In: Cryptographic Hardware and Embedded Systems (CHES 2016), Springer, LNCS, 9813, pp. 192212 (2016)
9. Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., Rijmen, V.: Trade-Offs for Threshold Implementations Illustrated on AES. IEEE Trans. on CAD of Integrated Circuits and Systems 34(7), 11881200 (2015)

Thank You!