Analysis of Vivado implementation strategies regarding side-channel leakage for FPGA-based AES implementations

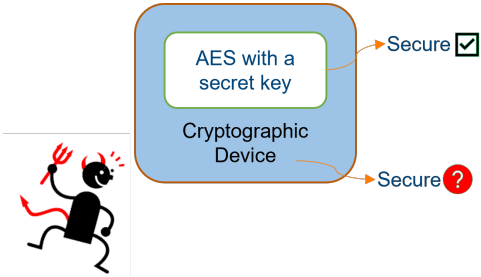
HASP'24

Ali Asghar, Andreas Becher, Daniel Ziener

November 2, 2024

Table of Contents

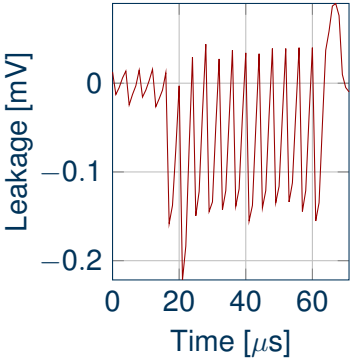
- Modern security systems rely on encryption algorithms
- Encryption algorithms like AES are mathematically secure
- Cryptographic implementations on computing devices can be vulnerable to attacks



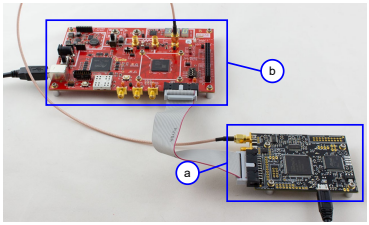
Cryptographic devices in possession or close vicinity of an attacker are vulnerable to physical attacks

Background

- Side channels like: timing, power, and EM radiations can leak useful information related to the encryption algorithm.
- Cryptographic devices implementing encryption algorithms can be attacked[4, 5].



An example of AES power trace

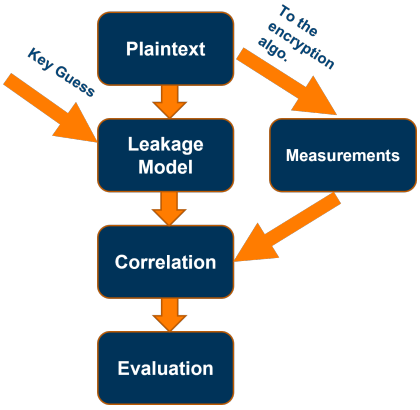


ChipWhisperer Measurement Setup. a). CW-Lite and b). CW305 target board

Background

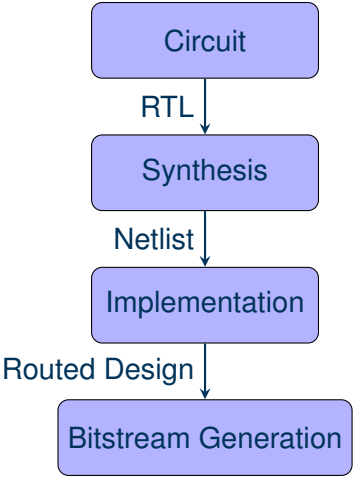
Side Channel Analysis

- Adversary takes advantage of the correlation between the processed data and some physical observations, e.g., Correlation Power Analysis (CPA).
- Create an effective leakage model[a] Generally Hamming Weight (HW) or Hamming Distance (HD) model is used.

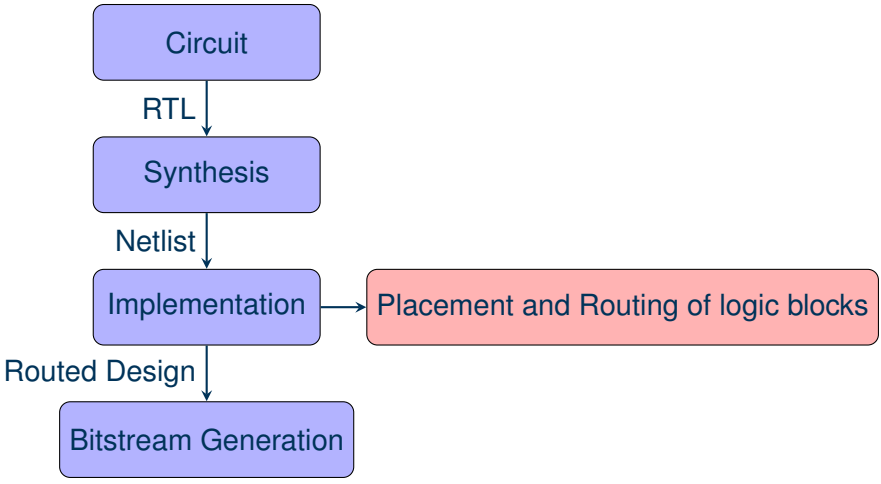


[a] A leakage model is a mathematical function which can establish an effective relation b/w the power consumed and the processed data

FPGA CAD Flow



FPGA CAD Flow



Motivation

Observation: Different implementations of a cryptographic implementation offer different resistance to SCA[1, 3].

- What makes an implementation more/less leaky?
- Can we perform a fine-grained analysis (down to bit-level) for an implementation?
- Does more leakage translates to more vulnerability against SCA?

Motivation

Proposed Approach

- Create several implementations of an encryption core.
- Quantify leakage of individual bits of an implementation.
- Identify implementations with a high no. of most/least leaky bits.
- Correlate the leakage with SCA resistance.

Experimentation

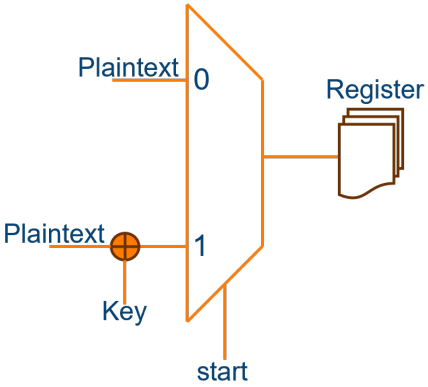
Measurement Setup

- Power measurements collected on Chipwhisperer's CW-305 board[2].
- A total of 33 variants[a] are generated for two AES encryption cores.
- Placement of both cores is constrained to a Pblock.
- For CPA, we use the Hamming Distance between the last round switch-box input and the corresponding ciphertext byte as the leakage model.

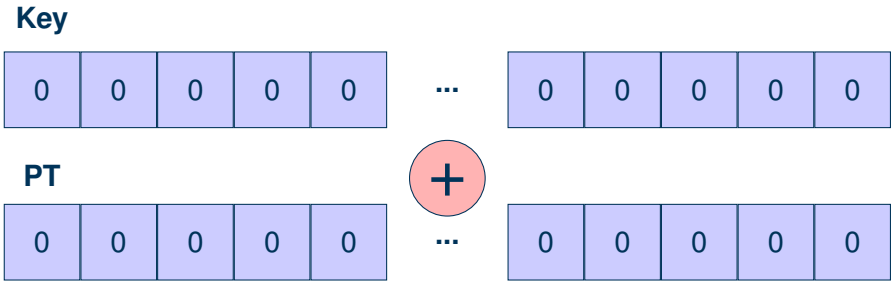
[a] Vivado design suite has a total of 33 implementation strategies using different placement and routing approaches

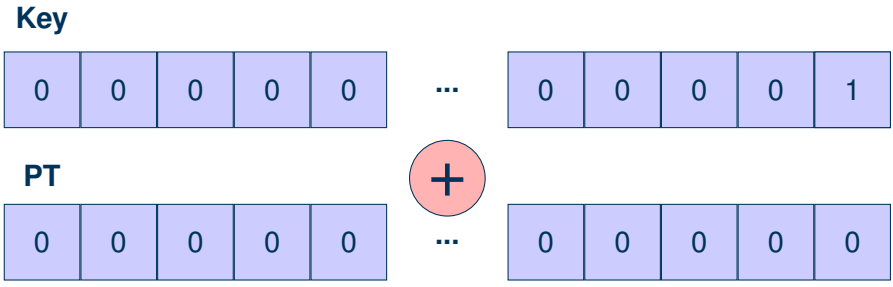
Bitwise Leakage

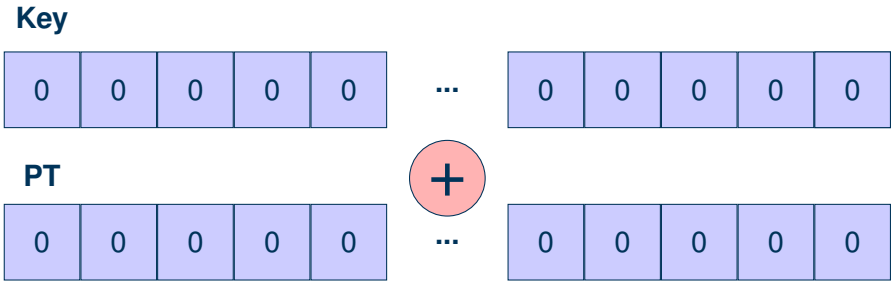
- Leakage values for instants at which only the concerned bit is toggling.
- Observed only during the Plaintext (PT) register overwrite during the first AES round.

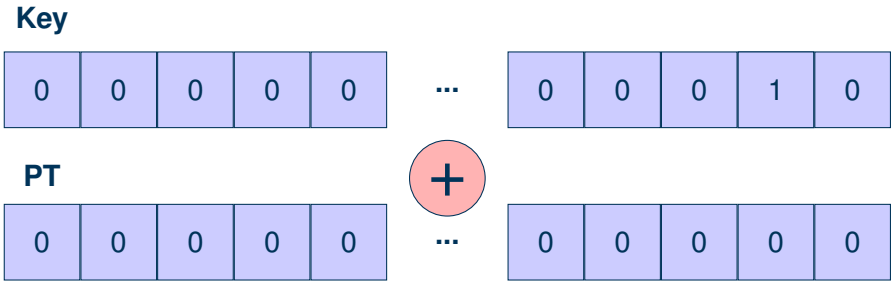


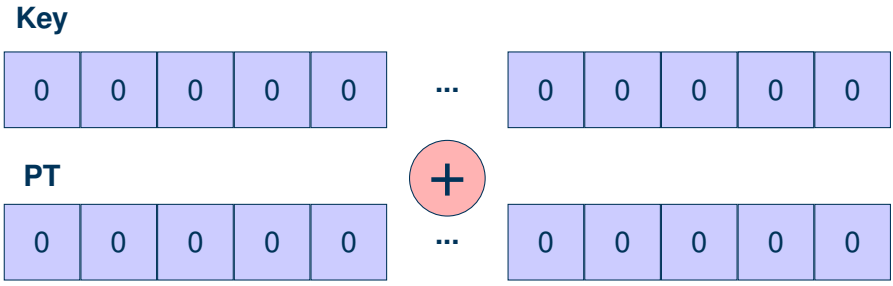
Register overwrite operation in the first AES Round

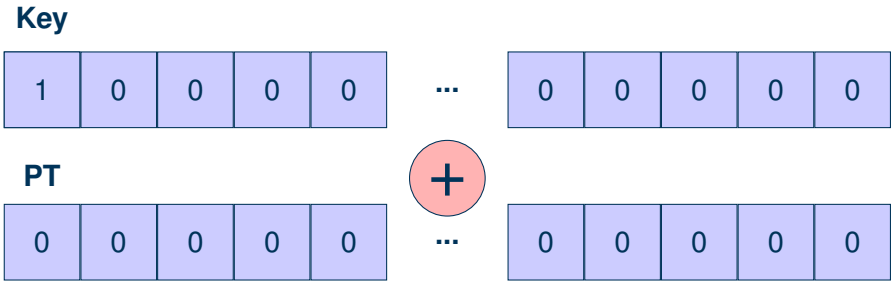

Key
0 0 0 0 0 ... 0 0 0 0 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+

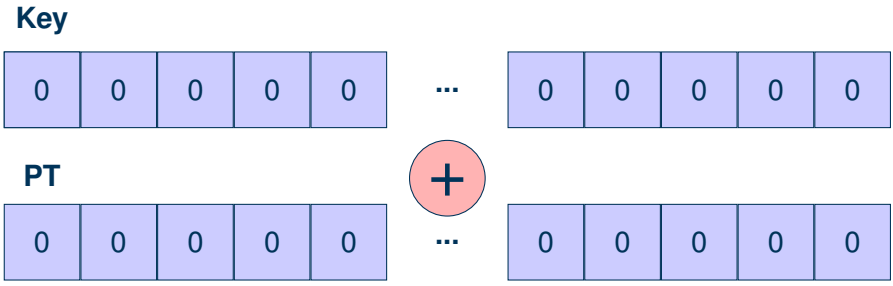

Key
0 0 0 0 0
...
0 0 0 0 1
PT
0 0 0 0 0
+
...
0 0 0 0 0

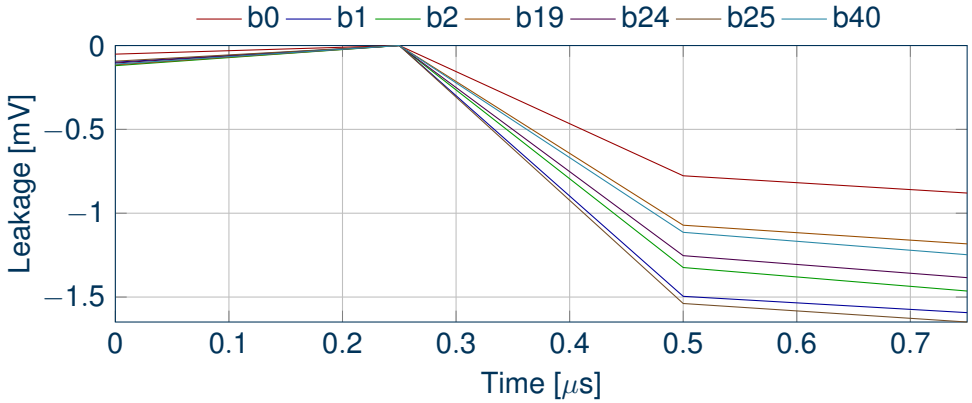

Key
0 0 0 0 0 ... 0 0 0 0 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+


Key
0 0 0 0 0 ... 0 0 0 1 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+

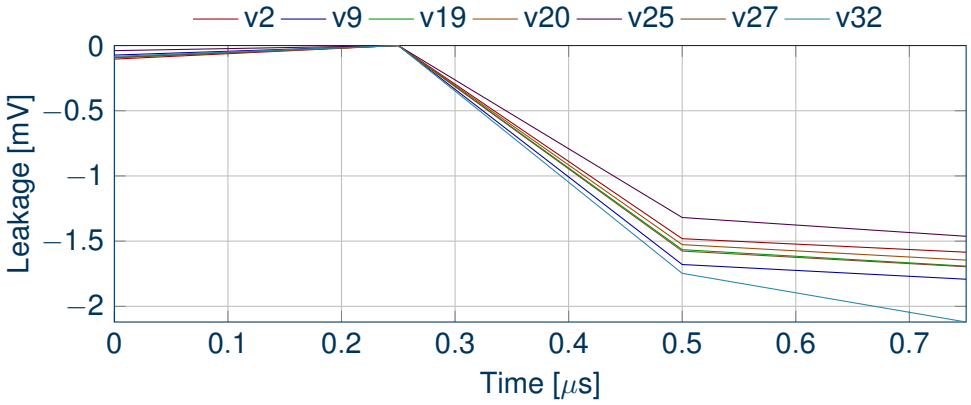

Key
0 0 0 0 0 ... 0 0 0 0 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+


Key
1 0 0 0 0 ... 0 0 0 0 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+


Key
0 0 0 0 0 ... 0 0 0 0 0
PT
0 0 0 0 0 ... 0 0 0 0 0
+



Leakage behaviour of different bits of a variant during the Add Round Key operation in the first AES round

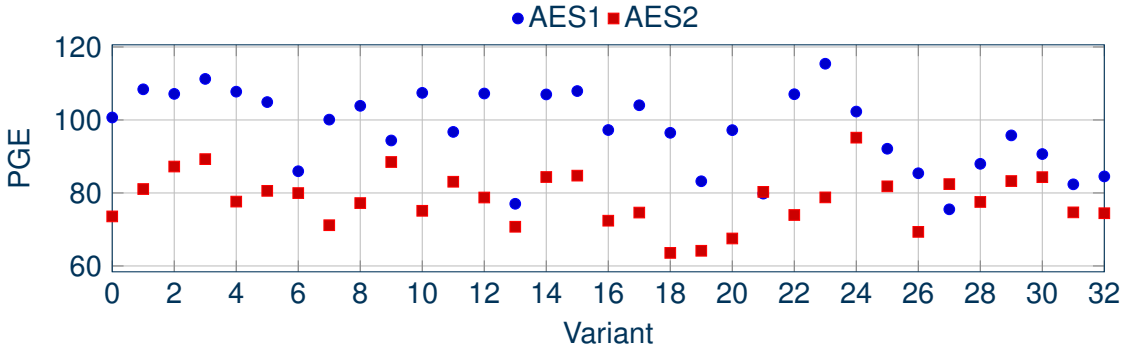


Leakage behaviour of bit2 of different variants during the Add Round Key operation in the first AES round

Bit leakage for *AES1* variants

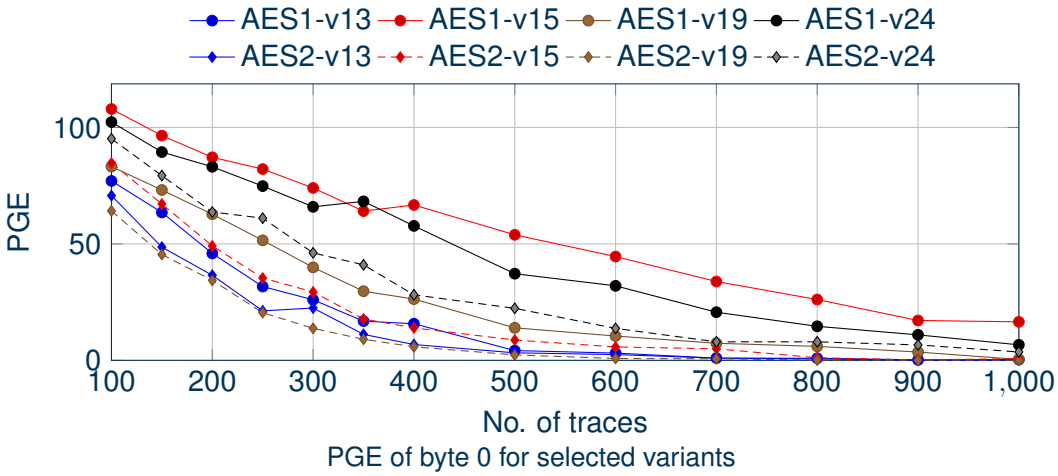
Variant	Most	Least
v0	0	5
v1	0	6
v2	0	1
v3	1	1
v4	1	3
v6	5	3
v7	1	0
v9	0	14
v10	11	3
v11	2	2
v13	4	6
v14	0	2
v15	0	7
v17	1	0

Variant	Most	Least
v18	7	4
v19	7	1
v20	16	4
v21	6	3
v22	0	4
v24	1	5
v25	4	22
v26	1	3
v27	21	2
v28	2	11
v29	2	0
v30	2	0
v31	1	13
v32	33	4



PGE [1] of first byte of *AES1 and AES2* (100 traces)

[1] Partial Guess Entropy PGE provides an estimate of how far the guessed key value is from the actual key. An implementation with a high PGE value offers more resistance to SCA.



PGE of byte 0 for selected variants

Conclusion

- Implementation strategies can have a significant impact on the leakage of FPGA-based cryptographic implementations.
- Bitwise leakage analysis reveal varying leakage for different bits of the same variant and same bits of different variants.
- Higher leakage doesn't always correspond to greater vulnerability to SCA.
- In the future, we intend to analyze the root-cause of this varying leakage behaviour by analyzing implementations at the netlist level.

References

[1] Ivan Bow et al. “Side-channel power resistance for encryption algorithms using implementation diversity”. In: *Cryptography* 4.2 (2020), p. 13.

[2] Alex Dewar, Jean-Pierre Thibault, and Colin O’Flynn. “NAEAN0010: Power Analysis on FPGA Implementation of AES Using CW305 & ChipWhisperer”. In: Last Update: Oct 29, 2020.

[3] Benjamin Hettwer et al. “Securing cryptographic circuits by exploiting implementation diversity and partial reconfiguration on FPGAs”. In: *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2019, pp. 260–263.

[4] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential power analysis, advances in cryptology-CRYPTO’99”. In: *Proc. 19th Annual International Cryptology Conf*. 1999, pp. 388–397.

[5] Paul C Kocher. “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”. In: *Annual International Cryptology Conference*. Springer. 1996, pp. 104–113.

Thank you for your attention!

ali.asghar@tu-ilmenau.de | www.tu-ilmenau.de