Altera vs. Xilinx
which one keeps your design hidden?

22. Aug. 2013

Amir Moradi
Embedded Security Group
Ruhr University Bochum, Germany
Acknowledgment

- Alessandro Barenghi
- Markus Kasper
- Timo Kasper
- David Oswald
- Pawel Swierczynski
- Christof Paar
SCA on Bitstream Encryption Feature
Broken Families

Virtex-II Pro, SASEBO

Virtex-4, Xilinx DevBoard

Virtex-5, SASEBO-GII
Broken Families

Spartan-6, SASEBO-W

Stratix-II, SASEBO-B
New Targets

Stratix-III, Altera DevKit

Kintex-7, SASEBO-GIII
EM Analysis
EM Analysis
EM Analysis
EM Analysis
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Decapping
Altera’s Key Derivation

Key1, Key2 → FPGA

AES Key
Altera’s Key Derivation

AES Key = ENC_{Key1}(Key2)

Selecting an arbitrary Key1’
Key2’ = DEC_{Key1’}(AES Key)
(Key1’, Key2’) works the same as (Key1, Key2)
no added security!
Old vs. New Generations

- **Altera:**
  - AES-128 is replaced by AES-256
    - Key derivation stays the same
  - Counter is not increased arithmetically
    - much heuristics + proprietary schemes
      - revealed by reverse engineering the PC software
Old vs. New Generations

- **Altera:**
  - AES-128 is replaced by AES-256
    - Key derivation stays the same
  - Counter is not increased arithmetically
    - much heuristics + proprietary schemes
      - revealed by reverse engineering the PC software

- **Xilinx:**
  - AES-256 in CBC mode (as before)
  - HMAC is introduced (Virtex-6 and all 7 series)
    - no place in FPGA to save the HMAC key!
    - The first block of the encrypted bitstream is the HMAC key!