A random permutation based method for secure hardware implementations

Robin Schrijvers
Master IW Elektronica-ICT

Introduction

Problem: Side Channel Attacks

Solution: Randomizing measured data via random permutation

Test platform: S-BOX in ASIC

PRESENT cipher S-BOX:

Problem: implementing S-BOX in ASIC

Methods and implementations

4x4 S-BOX in ASIC

- Configurable LUT from FPGA to ASIC
- Registers are configurable ⇒ Permutation generation
- NanGateOpenCell library

Area on ASIC: 4063.7μm²

Random permutation

```
for (int i = 0; i < 16; i++) {
    int x = numbers[i];
    for (int j = 0; j < 16; j++) {
        y = perm_array[j];
        if (x >= y) {
            y = y + 1;
            perm_array[j] = y;
        }
    }
    perm_array.shift();
}
```

PRESENT S-BOX

- Stored data in FFs = permutation of cut set input
  - Permutation of input groups of cut set
  - Multiplexers decide which permutation will be selected
    - Data group of cut set = SELECT (3 or 4 bits)

Area of new S-BOX design: 7909.2μm²

Maximum clock frequency: 1.72 GHz

Conclusions

- Copied LUTs from FPGA to ASIC
  - Cascade of multiplexers
    - 4063.7μm²
- Dynamic use of S-BOX with permutation generation possible
- Randomized the data on intersections where attackers perform DPA's
  - Stored random permutation of input groups in FFs
  - Cut set: 7909.2μm²
  - Maximum clock frequency: 1.72 GHz
- Further research:
  - Improve speed and area
  - Copy more fields of FPGA to ASIC with NanGateOpenCell library

Promotoren / Copromotoren: Prof. dr. ir. Nele Mentens, Bohan Yang