Side-channel Power Resistance for Encryption Algorithms using Dynamic Partial Reconfiguration

I. Bow, N. Bete+, F. Saqib*, W. Che^, C. Patel#, R. Robucci#, C. Chan and Jim Plusquellic

+Google, *University of North Carolina, Charlotte, ^New Mexico State University, #University of Maryland, Baltimore Co., University of New Mexico

Side-channel Power Resistance for Encryption Algorithms using Dynamic Partial Reconfiguration

A Side Channel Attack countermeasure is proposed for FPGAs:

- Uses **implementation diversity** to construct multiple, functionally equivalent, implementations of replicated crypto engine components, e.g., SBOX
- Leverages **dynamic partial reconfiguration** (DPR) to reconfigure regions of the FPGA, creating a moving target architecture

The technique is called **SPREAD**, for Side-channel Power Resistance for Encryption Algorithms using Dynamic Partial Reconfiguration (**DPR**)

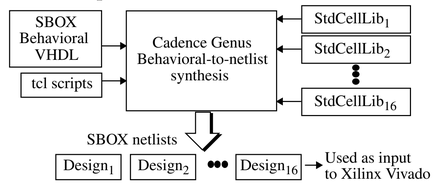
An embedded state machine periodically randomly selects an SBOX location and **reprograms it with a different implementation**

The diverse implementations of SBOX each have different path delays

The cause-effect relationship between path delays and power transients reduces correlations leveraged by **correlation power analysis** (**CPA**)

Logic Implementation Diversity

ASIC synthesis is used to construct the SBOX implementations using **different sets of standard cell components**



Adding and removing gates from the library forces synthesized netlists to vary widely **in structure and path lengths**

Design$_1$ **Design$_2$**

Logic Implementation Diversity

Diversity in standard cell usage after Cadence behavioral-to-netlist synthesis is run on a VHDL description of SBOX and four different standard cell libraries

Table 1:

Gate Type	INV	AND 2	AND 3	AND 4	AND 5	AND 6	OR 2	AO 1	AO 2	AO 3	AO 4	AO 5	AO 6	AO 7	...	AO n-2	AO n-1	AO n	Total
Design$_1$	8	20	2	5	3	19	76	1	6	1	13	1	29	8		14	18	-	300
Design$_2$	8	x	14	8	1	17	84	1	9	1	4	-	43	6		21	11	1	312
Design$_3$	8	25	4	6	24	x	77	-	8	1	10	-	24	8		16	13	-	312
Design$_4$	8	22	5	4	1	23	88	-	9	-	x	-	35	7		13	20	-	317

The table cell values give the number of instances of each standard cell gate type included in the design

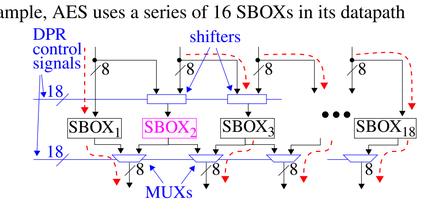
Cells marked '-' identify standard cells that were not used at all by the RTL compiler

Cells marked 'x' identify standard cells that were removed from the library before synthesis was run

Logic Diversity Countermeasure

Partial bitstreams are generated for each SBOX version using Xilinx Vivado, and stored on the FPGA for fast and secure access by the DPR state machine

For example, AES uses a series of 16 SBOXs in its datapath



The countermeasure includes **two redundant SBOXs** to enable DPR to be carried out while encryption continues at full speed

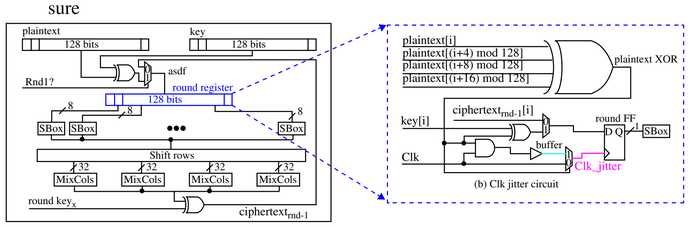
A state machine randomly selects an SBOX location and configures the shifters and MUXs **to create a 'hole' for DPR**

The 18th SBOX is simultaneously moved (much more frequently) to create additional diversity

Clock Jitter Countermeasure

Another major source of correlation leveraged by CPA are power transients associated with the latching event within registers

We introduce random delays to individual FFs (**clock jitter**) as a second countermeasure

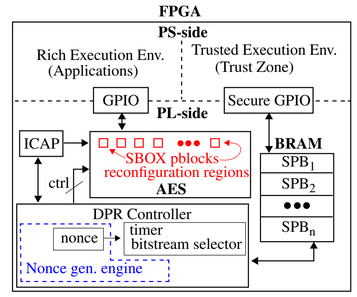


In our preliminary experiments, we use a deterministic source (plaintext components) to determine which clock signals to the FFs are delayed

A **TRNG** will eventually be used to control which FFs are delayed and by how much

SPREAD Architecture

The SPREAD engine is implemented as a state machine labeled **DPR Controller**

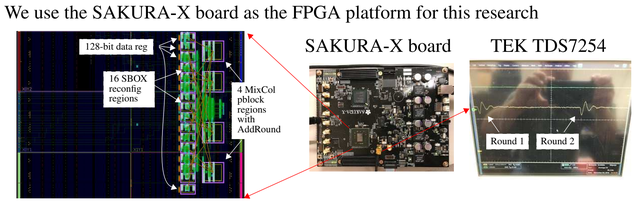


The sequence of operations carried out are as follows:

- TEE loads SBOX partial bitstreams (SPB) into PL-side BRAM resources
- DPR Controller starts TRNG to generate nonces to select SBOX location to DPR
- DPR Controller synchronizes with AES to reconfigure shifters/MUXs
- DRP Controller access ICAP to reconfigure the SBOX region

Setup for Proof-of-Concept Experiments

We use the SAKURA-X board as the FPGA platform for this research



Our proof-of-concept experiments are designed to find the optimal implementation diversity and clock jitter strategies

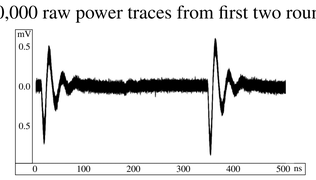
To accomplish this, we created **three diverse implementations**, V1, V2 and V3 and **three clock jitter models**, J1, J2 and J3 for a total of 12 static implementations

The clock jitter models differ in the amount of delay introduced and the position of the plaintext bits that enabled clock jitter

We collected **30,000** traces for each of the 12 implementations and used a **CPA attack** and a **Hamming weight model**

Power Trace Characteristics and CPA

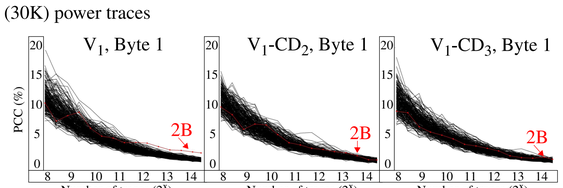
10,000 raw power traces from first two rounds of AES



Pearson's correlation coefficient used in CPA attack:

$$PCC = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\left[\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2\right]^{1/2}}$$

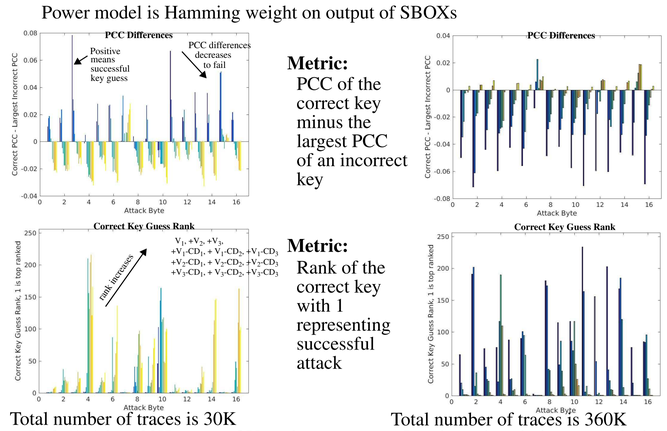
PCC evolution using Hamming weight on SBOX output using from 256 to 30,000 (30K) power traces



Logic diversity model V_1 (left), and with Clock jitter models J2 and J3

Correlation Power Analysis Results

Power model is Hamming weight on output of SBOXs



Metric: PCC of the correct key minus the largest PCC of an incorrect key

Metric: Rank of the correct key with 1 representing successful attack

Total number of traces is 30K
Left (1st) bar, 30K traces of V_1 only
Right bar (12th) bar, all Vs

Total number of traces is 360K
Interleaved evolutionary analysis with stop points at 8K, 16K, ... 360K

ELECTRICAL & COMPUTER ENGINEERING

UNC CHARLOTTE ELECTRICAL AND COMPUTER ENGINEERING
THE WILLIAM STATES LEE COLLEGE OF ENGINEERING

UMBC
Department of Computer Science and Electrical Engineering