Introduction

We propose a technique called Side-channel Power Resistance for Encryption Algorithms using Dynamic Partial Reconfiguration or **SPREAD**

As a countermeasure to DPA, CPA and other types of side-channel attacks techniques referred to as **SCA**

SPREAD changes the *underlying hardware* as a mechanism to reduce data correlations that are leveraged by SCA techiques

Replicated primitives within AES, in particular, the SBOX, are synthesized to multiple implementations using a set of **implementation diversity** techniques

During encryption/decryption, SBOX components are randomly selected and replaced dynamically on-the-fly with one of these diverse implementations

Dynamic replacement on FPGAs is done via *dynamic partial reconfiguration* (**DPR**)

A state machine reconfigures regions of the FPGA while the FPGA continues to execute encryption operations at full speed

Introduction

DPA depends on the underlying circuit implementation remaining **invariant**

By changing the implementation characteristics (while preserving the functional behavior), the signal delays change

Power trace behavior is directly related to the delay characteristics of the underlying hardware

Implementation diversity can be introduced by:

- *Circuit Level*: A fine-grained approach adds capacitive loads to the existing wires of the encryption engine

  Circuit level technique change path delays in subtle ways, adding only 10's of picoseconds to delays

- *Synthesis Directed*: A course-grained approach changes components in the standard cell library and/or makes small inconsequential changes to the RTL or netlist

  CAD synthesis tools are then used to introduce diversity in the implementations

  Synthesis directed techniques create large, nanosecond level, changes in delay

SPREAD Architecture

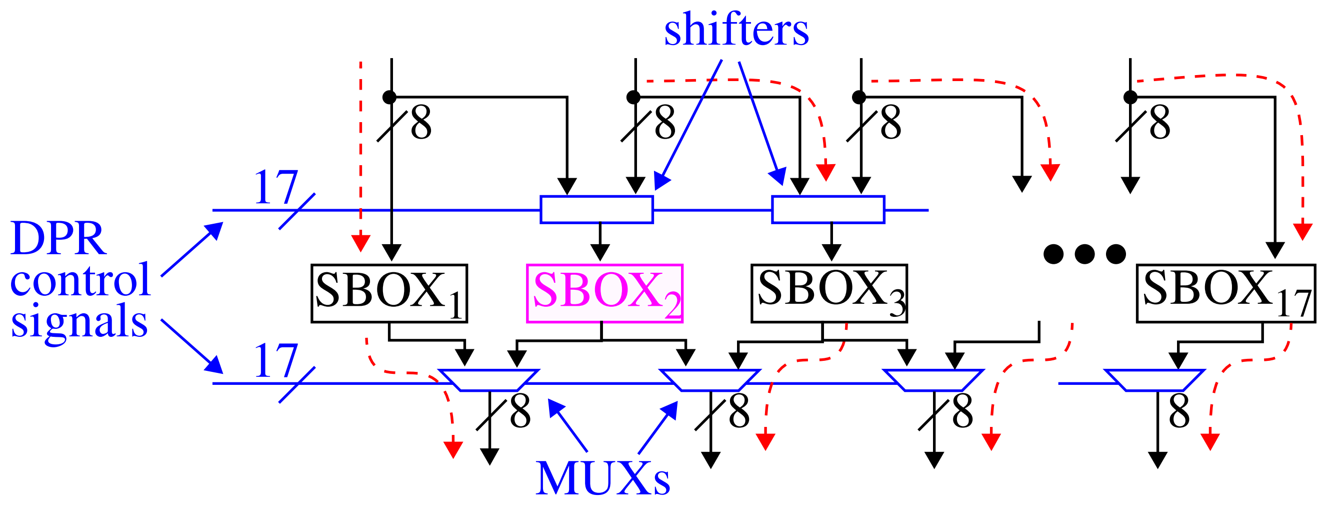
We propose a **moving target architecture** as a countermeasure to SCA

SPREAD can be applied to any type of encryption algorithm that incorporates replicated primitives, e.g., the SBOX within the AES algorithm

We use AES in our examples here

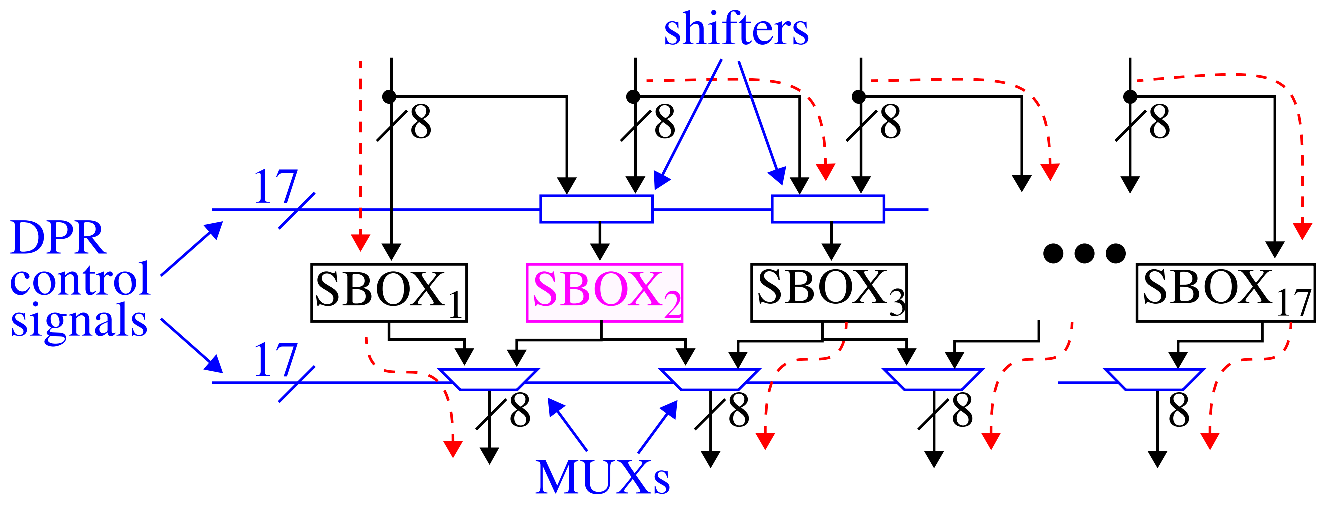
AES has 16 SBOXs in its datapath which operate in parallel in custom ASIC and FPGA implementations

SPREAD adds a 17th SBOX, and a set of shifters and MUXs



SPREAD Architecture

The 17th SBOX represents a *hole* in the datapath of AES that is dynamically reconfigured



A state machine controls the shifters and MUXs to move the hole to different locations, e.g., the figure shows SBOX$_2$ is configured as the hole

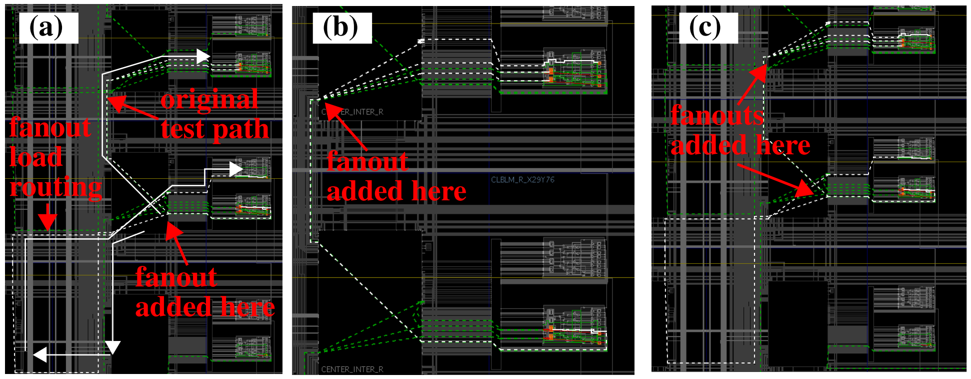
DPR is then used to replace SBOX$_2$ with, e.g., any one of 10 diverse implementations

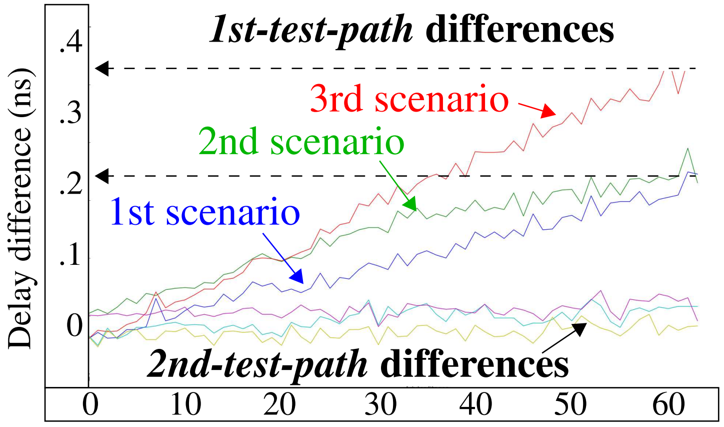
Note that the MIXCOLs component of AES is connected *combinationally* to the SBOX outputs, which deals with downstream data correlations

AES runs at full speed during DPR -- hole configuration adds only one stall cycle

SPREAD Implementation Diversity Techniques

Circuit Level: Adds capacitive loads to the P&R design (using FPGA *implemented design view*)





Capacitive loads are introduced by adding *wire stubs* to existing routes in the design

The clock strobing technique described in the PUF screencasts is used to measure changes in path delays as wire stubs are added

Each wire stub adds only approx. 3.3 ps to an existing route, so to be effective, many copies of wire stubs need to be added

SPREAD Implementation Diversity Techniques

Synthesis Directed: Uses CAD synthesis tools to add diversity as the design is synthesized to a netlist (via changes to the standard cell library) and/or implementation

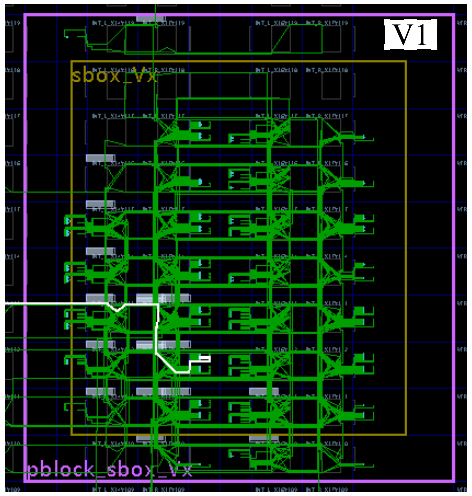
Synthesis Directed diversity methods include:

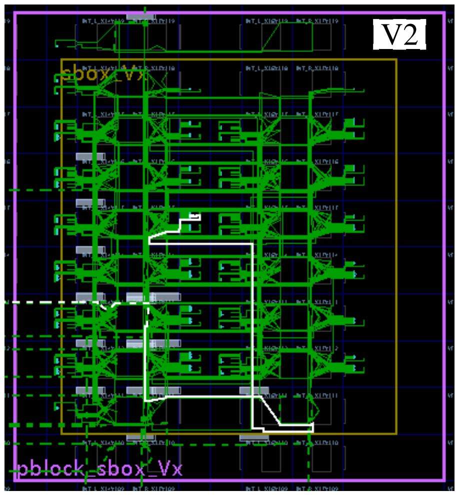
- Adding/removing logic gates from the standard cell library
- Changing the timing constraints
- Making small, inconsequential changes to the RTL

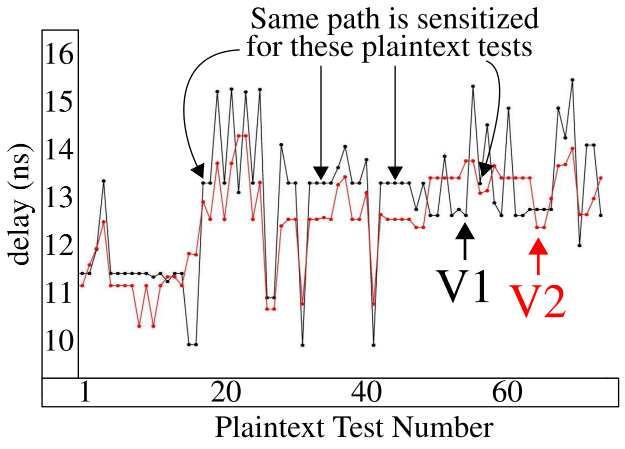
The implementations are functionally equivalent but differ dramatically in terms of their netlist, P&R and corresponding delay characteristics

SPREAD Implementation Diversity Techniques

The RTL for SBOX implementations *V1* and *V2* add two *dummy* input wires to the port list, which are connected directly in *V1* and through an inverter in *V2*







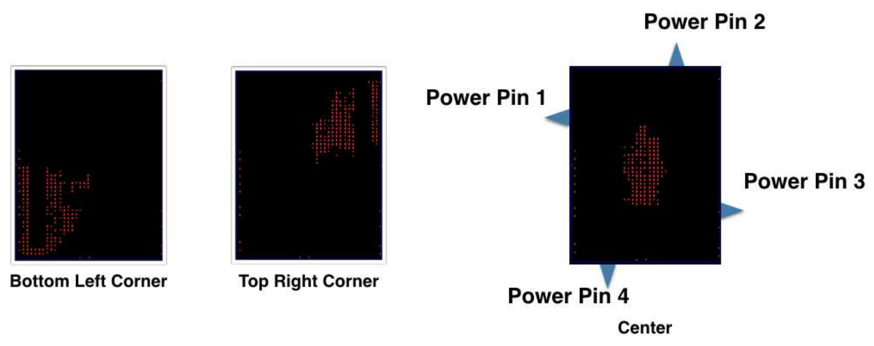
Xilinx Vivado creates different implementations, which produce different delays, as shown for a subset of the paths tested using the same plaintexts

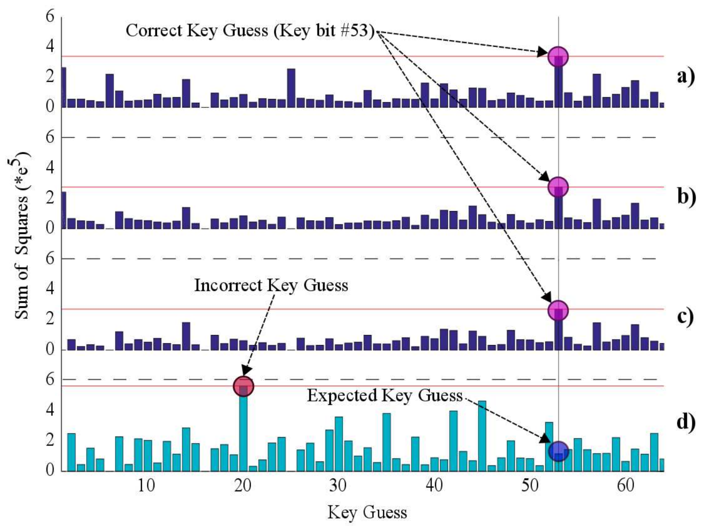
Note that paths with the same delay represent *frequently* tested paths and are likely to contribute significantly to correlations that DPA leverages

Synthesis directed diversity is likely to significantly reduce these correlations

Preliminary Results

DPA experiments using a Xilinx Spartan 3E FPGA, with 1,000,000 plaintexts to a 64-bit version of DES





The top three bar graphs are constructed from the DPA results of the 3 individual placements

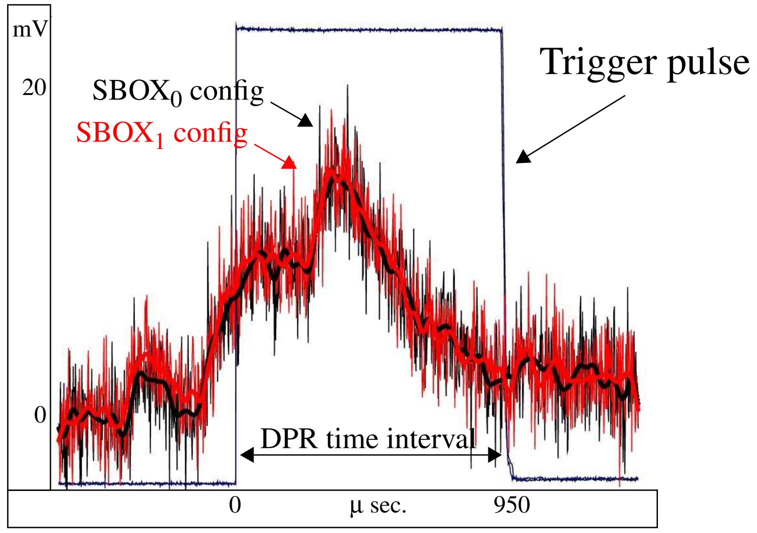
The correct key guess (decimal 53) is highlighted and is clearly the largest among the set of 64 possible key guesses in each of these graphs

Bottom bar graph mixes 1/3 of the power traces from the individual experiments

Peak associated with correct key significantly reduced and *ghost* peaks appear

DPR Leakage

Note that it is important that the adversary is not able to ’track’ which SBOX implementation is being removed/installed

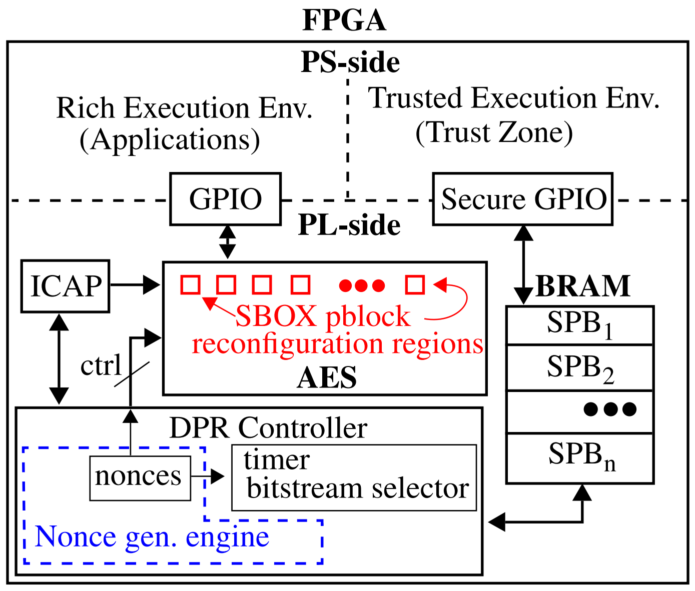


The traces shown above are produced by the DPR operation using two different SBOX implementations

Note that the number of possible configurations is exponential

For example, if any of 5 different implementations can be placed into any of the 16 positions of SBOX with AES than 1 million configurations are possible

SPREAD Controller



Operations

- PS-side loads SBOX partial bitstreams labeled SPB$_x$ into BRAM
- DPR Controller starts nonce generation engine
- DPR Controller uses the nonces to randomize the selection of the SPB$_x$, the reconfiguration region and the time interval between DPR operations.
- DPR Controller synchronizes with AES, asserts the appropriate control signals for reconfiguration of the randomly selected SBOXs and executes the transfer protocol with the ICAP controller