Whitenoise Encryption Implementation with Increased Robustness to Side-Channel Attacks

Mihai SIMA and André BRISSON

University of Victoria – Whitenoise Laboratories
British Columbia, CANADA

Trusted Computing in Distributed and Hybrid Systems Workshop
IEEE Conference in Advanced and Trusted Computing (*ATC 2017*)

August 4th, 2017, San Francisco Bay Area, California, USA

Scope of the Research

- Whitenoise encryption algorithm is considered highly robust
- Application Specific Integrated Circuit (ASIC) implementation of the Whitenoise encryption algorithm
  - A **Whitenoise chip** (hardware implementation) is investigated
  - Software implementations are outside the scope of this work
  - Field-Programmable Gate Arrays (FPGA) represent the boundary between hardware and software – short discussion is provided
- Side-channel attacks are serious threats to any cryptosystem
  - A **secure Whitenoise hardware implementation** is investigated
  - A secure Whitenoise FPGA implementation is a different problem

Side-Channel Attacks

- The secret key influences all activity in a cryptosystem
- The measurements from a physical implementation provide side-channel information, which can be used to break the system
  - Power consumption
  - Electromagnetic emissions
  - Execution time
- Attacks such as Differential Power Analysis and Correlation Power Analysis exploit the relation between data, operations, and power consumption to derive the secret key
- Pioneer work: Paul Kocher, Joshua Jaffe, and Benjamin Jun, “Differential Power Analysis,” in Advances in Cryptology, 1999

Example: Mounting an Attack based on Power Consumption

- Assume a cryptosystem with a 256-bit key
- Power consumption is recorded during operation
- If the attacker correctly guesses *all* 256 bits of the key, then all the activities inside the cryptosystem will add up constructively
  $\Rightarrow$ **power consumption will exhibit a maximum**
- **Attacking strategy** that reduces the search space:
  - Try all possible combinations for the first N bits and use random values for the remaining $256 - N$
  - Record which combination (or combinations) of the first N bits exhibits the largest power consumption
  - Repeat these steps for the next N bits

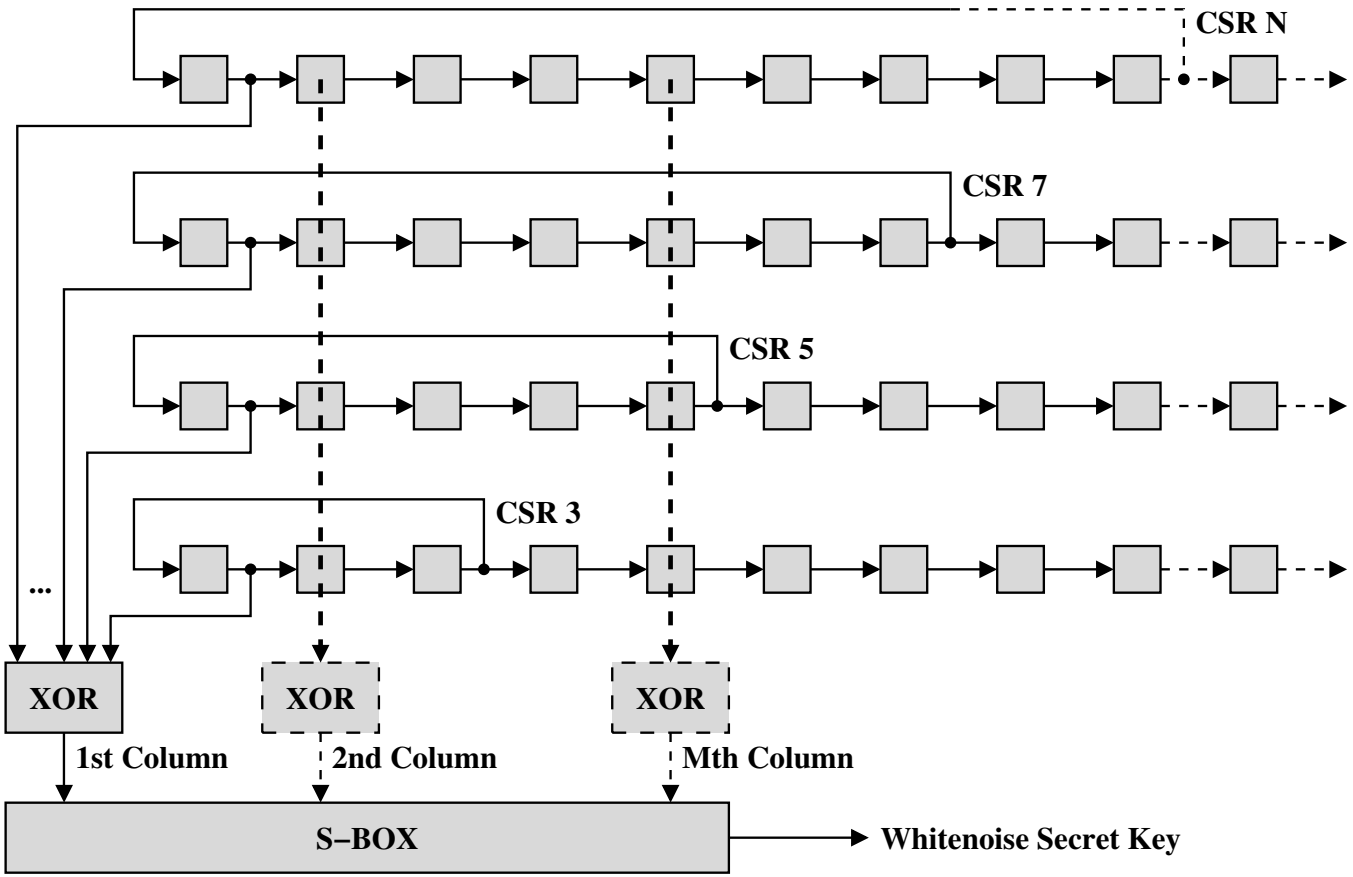
Techniques to Build Secure Implementations

- **Attacking strategy**
  - Power consumption depends on data and/or operations
  - Get information on data/operations through power consumption
- **Defense strategy**
  - Power consumption independent of data and operations
  - Eliminate the correlation between power and data / operations
- **Hiding / Concealing**
  - Balances the power consumption into a constant value
- **Masking**
  - Randomizes the power consumption

The Power Consumption Attack Model

- The attacker has physical access to the chip
  - Physical access to every chip of a batch to be attacked
- The power signal can be collected from the pins only
  - Chip decapping is not possible or is too expensive
  - Internal probing is not possible or is too expensive
- Semiconductor plant where the chip is manufactured does not change its layout or alter it in any way
  - Intentional defects are not introduced
- Enough time to analyze the power signal is available.

Whitenoise Algorithm and Its Behavioral Implementation



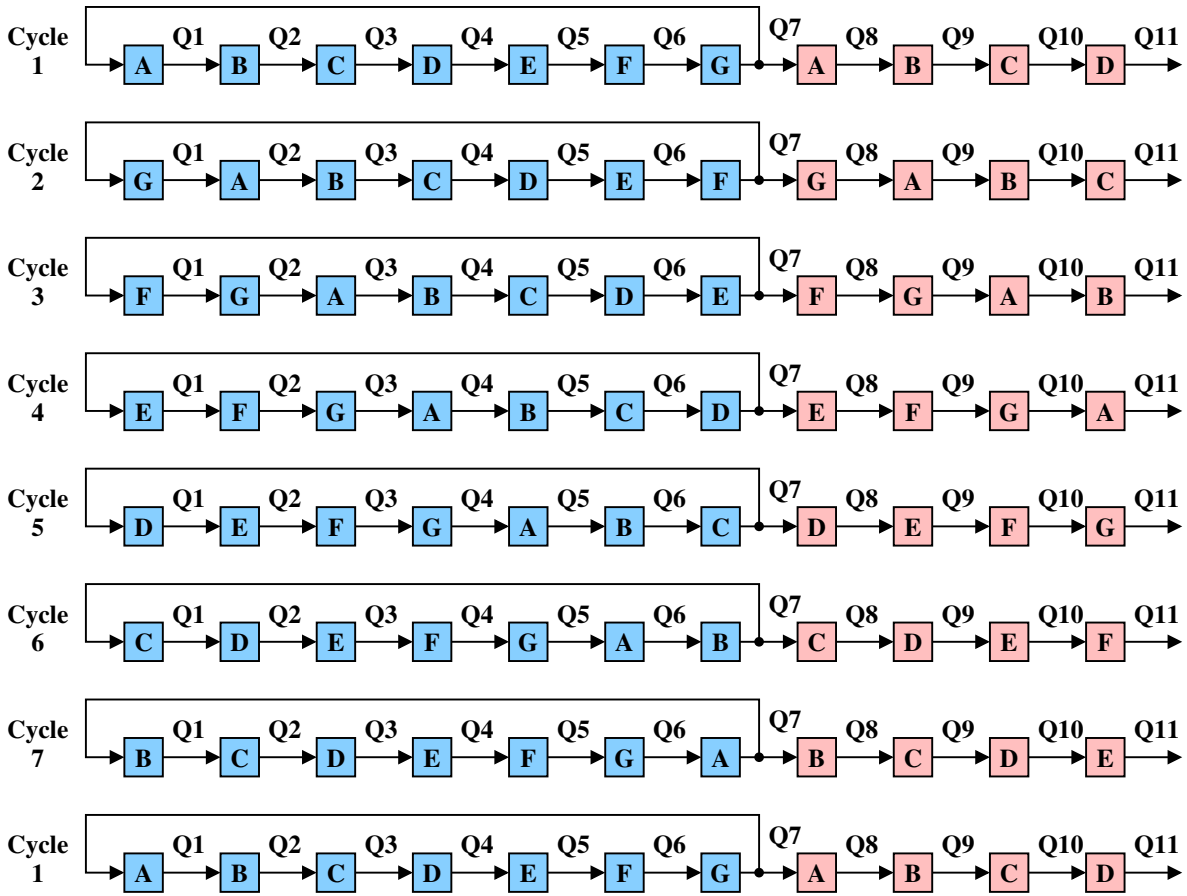
- Prime-length Circular Shift Registers
- Outputs are XOR'd
- Delinearization S-box

Whitenoise Algorithm Implementation

Practical Details

- The maximum number of CSRs as allowed by the algorithm and cost of the chip is deployed
- Each CSR is manufactured with the largest number of cells allowed by the algorithm and cost of the chip
- Based on randomized data from the master key, a set of selectors (not figured) will define the length and byte values of each CSR.
- Assume, for the sake of presentation, that:
  - Maximum configurable length of a CSR / subkey is 11
  - One of the CSRs has seven cells, whose configured byte values are A, B, C, D, E, F, and G

Operation of a Circular Shift Register with Seven Cells

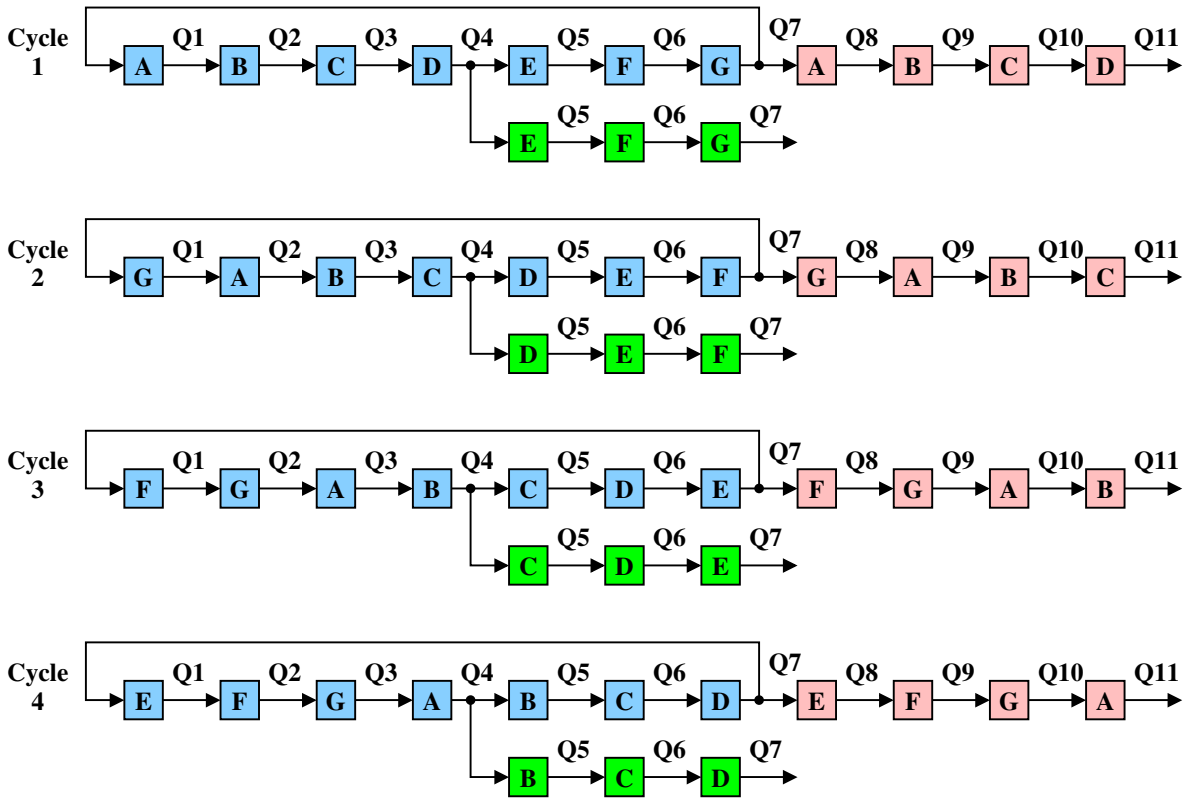


- Position of the bytes stored into main (blue) loop changes, not their values
- Main (blue) loop's total power does not change in time
- The stub of four (red) cells exhibits periodic power consumption: **period equals CSR length** $\equiv$ **S.C. Info**

How to Make Power Consumption Constant

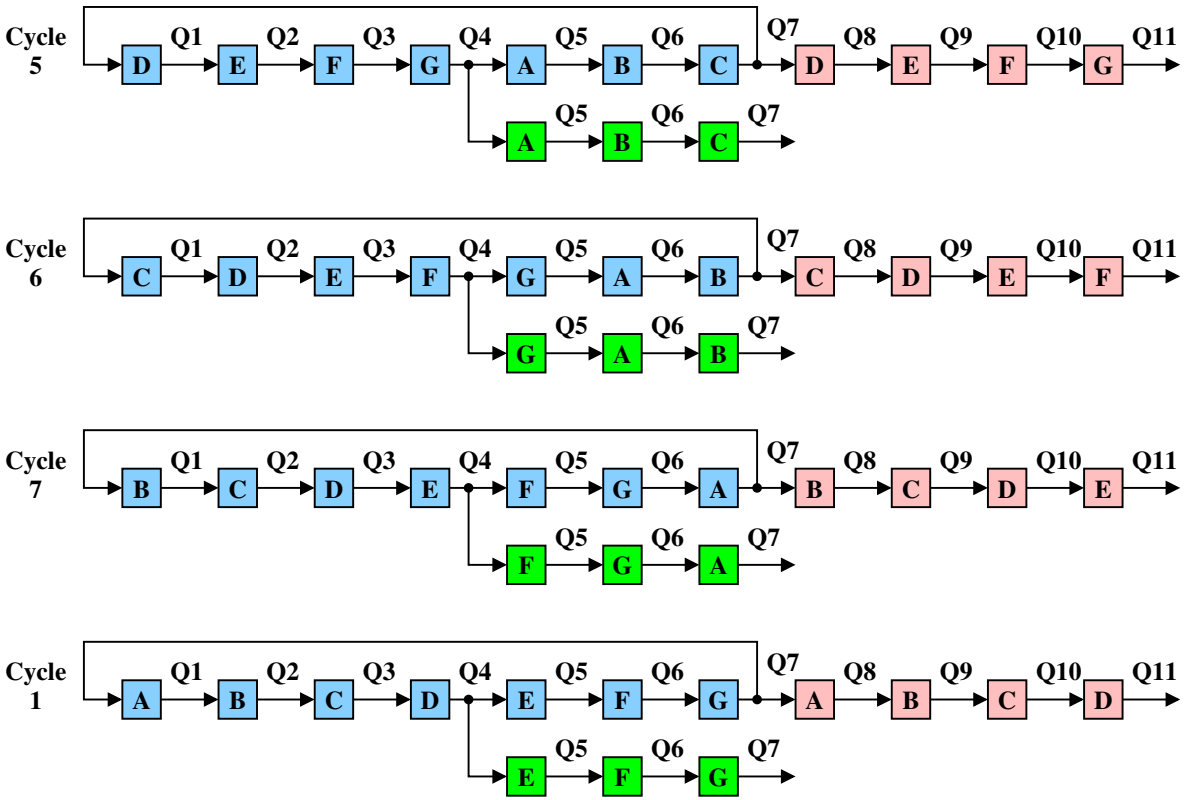
- **Remove the stub (red) cells**
  - **Removing cells during manufacturing** renders fixed-length CSRs
  - **Deactivating cells by forcing them idle** seems promising
- The power of the main (blue) part depends on Hamming distance
  - *Hamming distance between two bit strings of equal length is the number of positions at which the corresponding bits are different*
  - Hamming distance may change when the subkey is reloaded
  - Power attacks (notably leakage attacks) based on Hamming distance have been reported very effective
- **Add new cells** to make the total power consumption constant
  - This technique is presented next

Circular Shift Register with Constant Power I



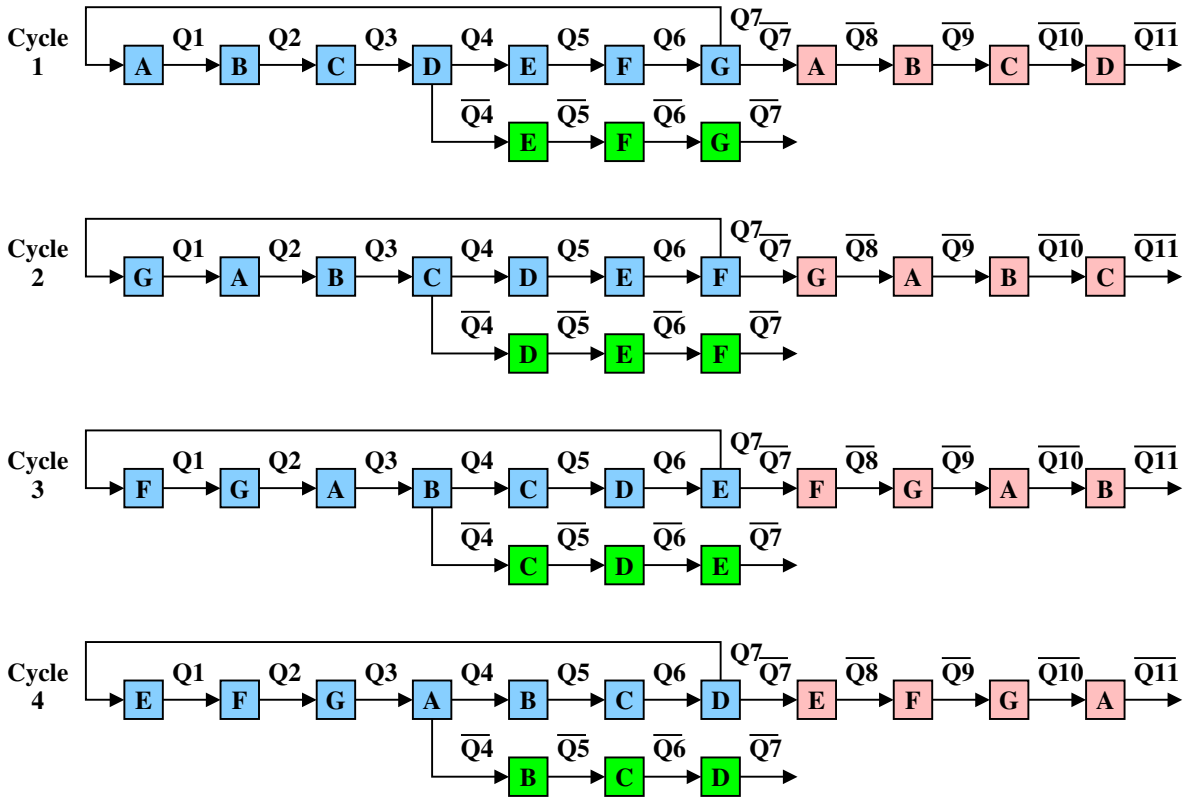
- Extra (green) stub deployed / activated
- # of blue cells = # of red+green cells
- Byte values in the blue part is identical to the red+green part
- Power is constant
- Power depends on Hamming distance

Circular Shift Register with Constant Power II



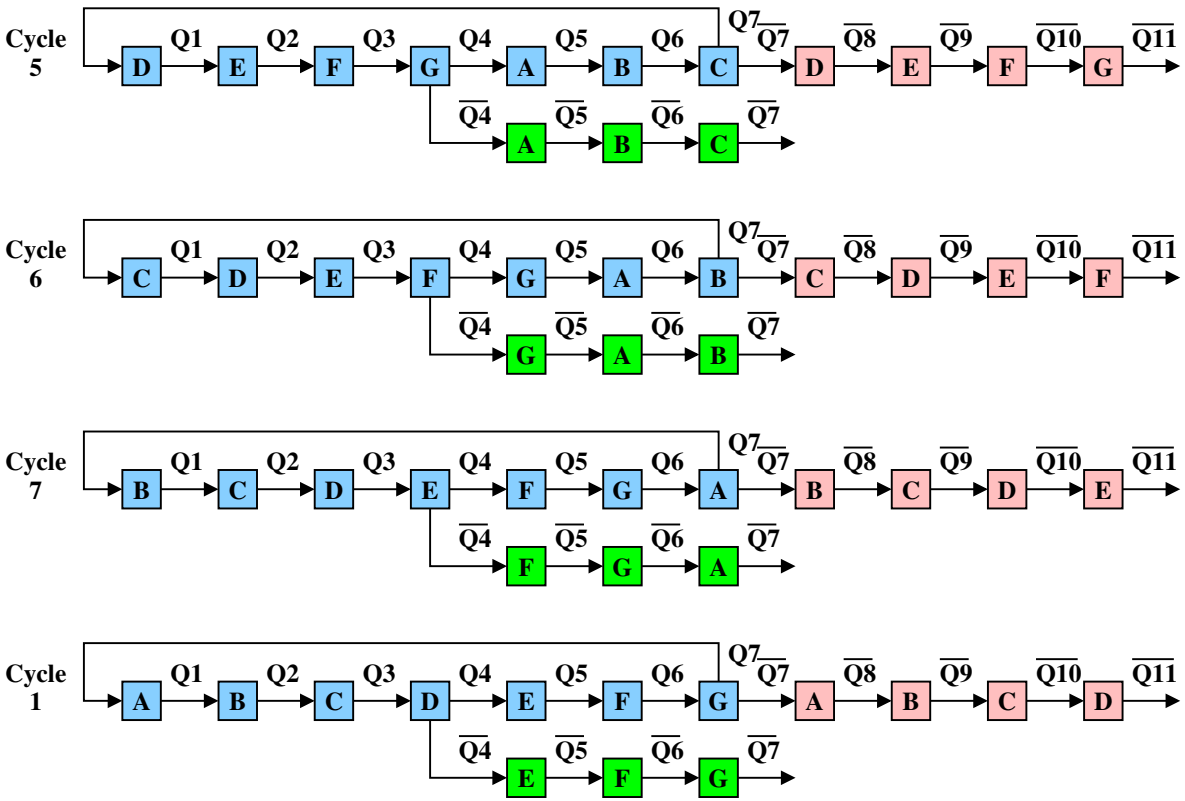
- Operation during Cycles 5, 6, 7, and 8 is shown here
- Power dependence on Hamming distance needs to be removed
- This problem is addressed next

Secure Circular Shift Register I



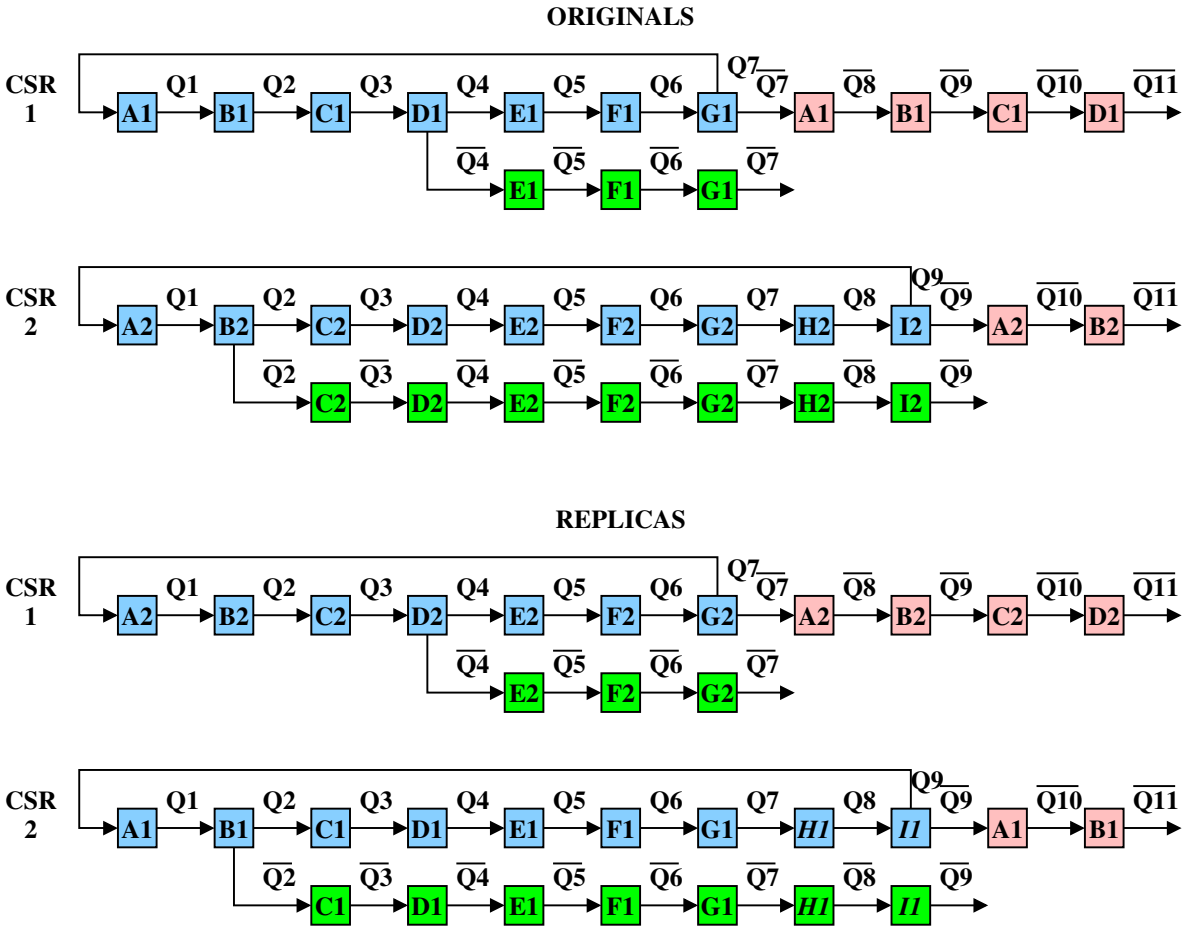
- The red+green part is driven with inverted signals (no extra HW!)
- # of cells in Logic '1'
  =
  # of cells in Logic '0'
- Power dependence on Hamming distance has been removed

Secure Circular Shift Register II



- Operation during Cycles 5, 6, 7, and 8 is shown here
- Technological variation can affect this symmetry
- Next technique randomizes power consumption

Implementation with Randomized Power Consumption I



- Original CSRs
- Each CSR has been replicated N-1 times
- The replicas run the subkeys of the other CSRs to produce fake randomized data
- CSR 1 replica runs the original Subkey 2
- CSR 2 replica runs the original Subkey 1

Implementation with Randomized Power Consumption II

- $N \times N - N$ registers running fake data
- Numerical example: $N = 10$ registers
  - 90 registers run fake data
  - 10 registers run real data
  - 90% fake side-channel information
- Improvement in robustness through randomizing power consumption is achieved at **quadratic cost**
- Improvement in robustness through balancing power consumption into a constant value is achieved at **linear cost**
- No fancy circuit techniques

Secure Implementation on FPGA – Brief Discussion

- Would the proposed techniques lead to a secure Whitenoise implementation on FPGAs?
- FPGA computing blocks are very similar (if not identical)
- The programmer does not have control of the routing
- The routing is heterogenous
- Pessimism that the presented techniques are effective on FPGAs

Application Scenarios

- **Behavioral (non-secured) implementation**
  - Cheapest one with the lowest level of robustness
  - Phonecards, bus passes
- **Power concealment protection**
  - Linear cost with medium level of robustness
  - Smartphones, wearable electronics, IoT devices
- **Power masking protection**
  - Quadratic cost with high level of robustness
  - Hard-drives, USB memory keys

Review of Contributions

- A design technique based on cell replication, which provides a high degree of concealment of the CSR power consumption
- A design technique based on signal polarity inversion, which further removes the power consumption dependence on the Hamming distance
- A design technique based on fake keys, which randomizes the power consumption of the CSRs

Conclusions

- Side-channel attacks are a major threat of any cryptosystem
  - **Implementation** is the weak chain, not the **algorithm**
- The proposed techniques are conceptual
  - Independent of (and orthogonal to!) the physical circuits used
  - Cheaper than circuit-level protection techniques
- **Future work**:
  - Investigate defense measures to attacks based on electromagnetic radiation, temperature, fault attacks, etc.
- **Questions** are welcome!