Self-Modifiable Reversible Circuits

Michael Stephen Fiske
HICSS-59

Aemea Institute

January 9, 2026

Discovery of Reversible Computing

- Landauer (1961): Computers are implemented with devices (gates) that are not 1-to-1.
- A function f is 1-to-1 if $x_1 \neq x_2$ implies $f(x_1) \neq f(x_2)$.
- AND, OR, NAND, NOR are not 1-to-1. OR(1, 0) = OR(0, 1).
- Non-invertible math operations executed in physical devices must increase entropy and produce heat.
- Bennett (1973): theoretically possible to build logically reversible computing machines.
- Fundamental relationship: physics, information & computation.

Contributions of Self-Modifiable Reversible Circuits

- Extends Reversible Computing. Circuits change the Boolean function they compute without physically altering connections or components.

- Unpredictable execution of machine instructions on different gates and at different physical locations on the circuit.

- Physics obfuscation based on Space-Time and Heisenberg uncertainty principle. Cryptography based on complexity theory relies on algebraic structure of the algorithm.[1]

[1] Our methods do not rely upon the existence of one-way functions.

Contributions of Self-Modifiable Reversible Circuits II

- Enables Hindering Cryptanalysis Attacks.
- Unpredictably moves the AES S-Box computation.
- Likely to be Applicable to Quantum Computing.
- Long-term Potential for Repairing Circuits.

Related Work

- "Conservative Logic" (Fredkin & Toffoli, 1982): a model of computation, based on fundamental principles in physics.

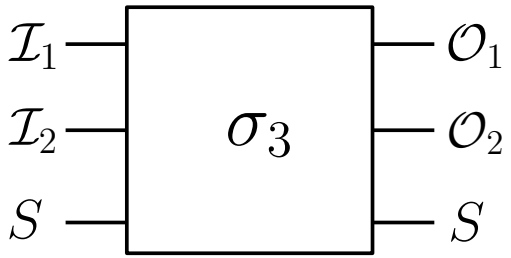
- Yao (1982): secure multi-party computation. Alice and Bob jointly compute a function on private inputs. Depends on one-way functions.

- Barak et al. (2010): impossible to obfuscate a program. Their math proof depends on one-way functions. There is no math proof that one-way functions exist.

Roadmap

- Reversible gate, called σ_3.
- Show σ_3 is universal: NAND, NOR.
- A self-modifiable circuit: function computed at a gate output can be changed without changing the physical connections of the input and output lines.
- AES S-box procedures.
- Build the AES S-box with reversible circuits.
- Obfuscate S-box circuits: unpredictably move the location of bit computations.

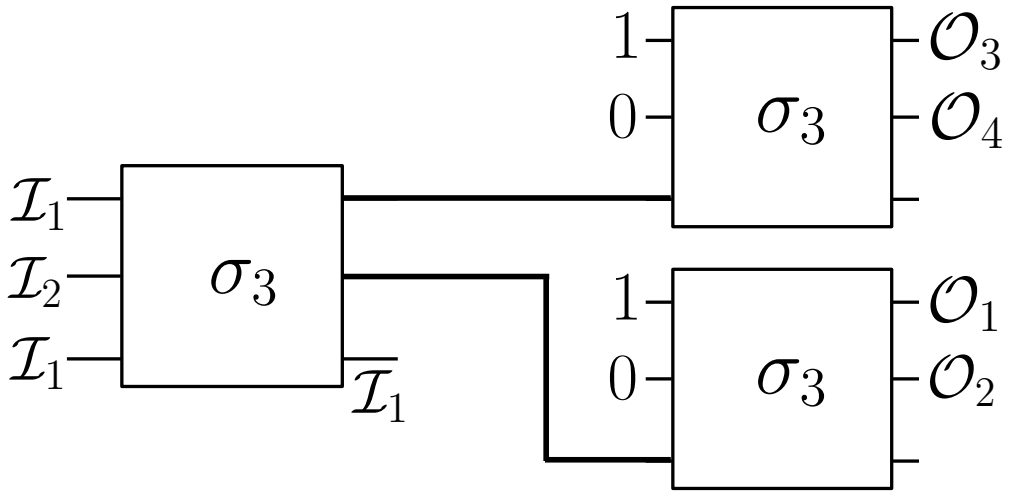
Reversible σ_3 Gate: If $S = 0$, Then Swap $\mathcal{I}_1$ & $\mathcal{I}_2$



3-bit Reversible Gate Logic

$\mathcal{I}_1$	$\mathcal{I}_2$	S	$\mathcal{O}_1$	$\mathcal{O}_2$	S
0	0	0	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	1	1	0
0	0	1	0	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	1

σ_3 is a Universal, Reversible Gate

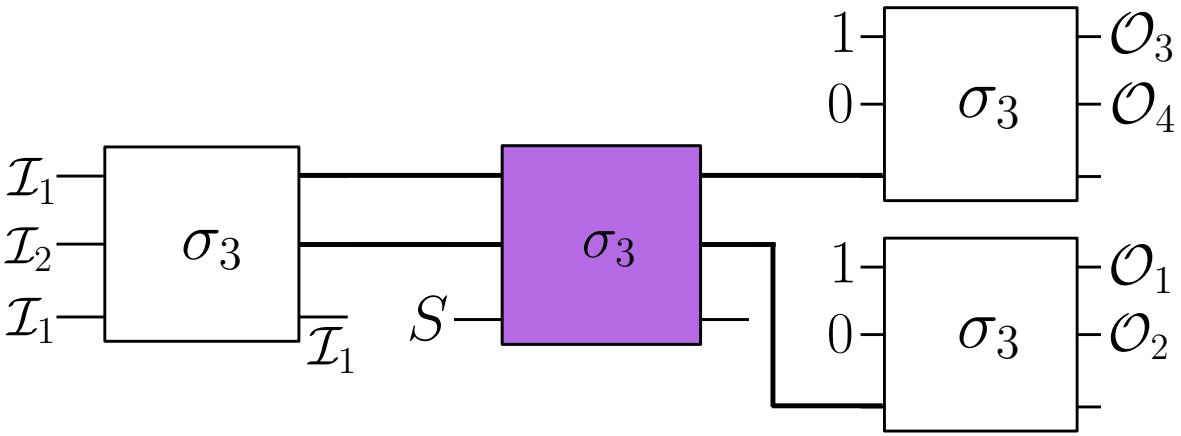


$$\mathcal{O}_4 = \mathsf{NOR}(\mathcal{I}_1, \mathcal{I}_2)$$

$$\mathcal{O}_2 = \mathsf{NAND}(\mathcal{I}_1, \mathcal{I}_2)$$

A Self-Modifiable Circuit

Self-modifiable hardware circuit: function computed at a gate output can be changed without changing the physical connections of the input and output lines.



When S flips from 1 to 0:

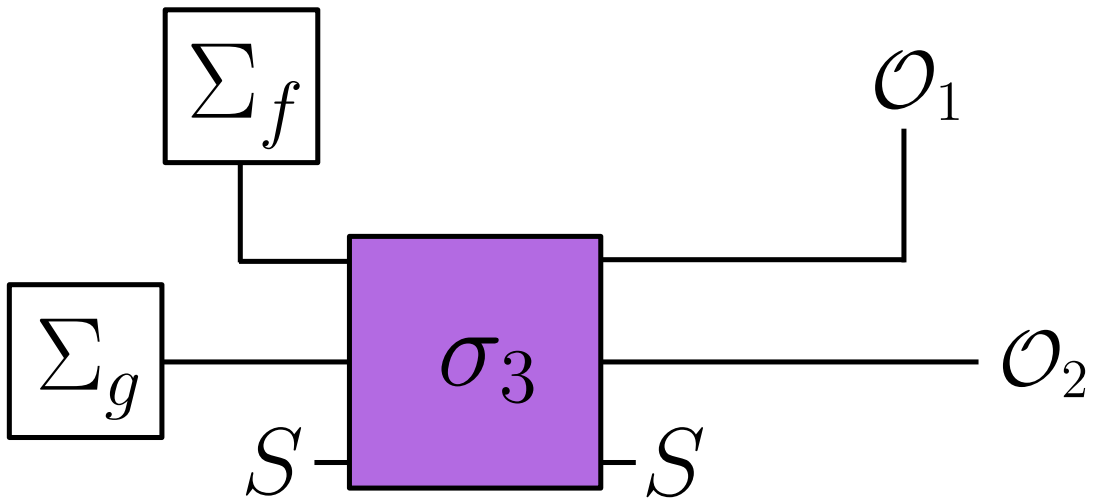
Output $\mathcal{O}_4 = \text{NOR}(\mathcal{I}_1, \mathcal{I}_2)$ is changed to $\mathcal{O}_4 = \text{NAND}(\mathcal{I}_1, \mathcal{I}_2)$.

Output $\mathcal{O}_2 = \text{NAND}(\mathcal{I}_1, \mathcal{I}_2)$ is changed to $\mathcal{O}_2 = \text{NOR}(\mathcal{I}_1, \mathcal{I}_2)$.

General Principle: Self-Modifiable Circuits

Subcircuit Σ_f computes Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$.

Subcircuit Σ_g computes Boolean function $g : \{0,1\}^n \rightarrow \{0,1\}$.



By routing outputs $\mathcal{O}_1$ and $\mathcal{O}_2$ to different physical locations in the circuit, and using randomness, computation can unpredictably move across a circuit.

Motivation for Unpredictable Circuits I

- Each time AES executes the physical location of the computation unpredictably changes.

- Advantageous: Kerckhoff's principle is violated. *The system must not require secrecy and can be stolen by the enemy without causing trouble*.

- Evolutionary biology. Red Queen hypothesis: organisms change to maintain survival fitness.

- Kerckhoff's lifetime: no notion of machine instructions executing.

Motivation for Unpredictable Circuits II

- Algorithmic transparency and execution predictability address different threat models.

- Acoustic, power (Kocher, 1999) and timing (Bonneau, 2006) cryptanalysis can capture keys. Eve relies on knowing the machine instructions, and order and timing of their execution.

- Mallory can bypass register machine execution of cryptographic instructions (Valbuena, 2020).

- Randomness in circuits: unpredictable S-box execution.

- Unpredictable execution can be applied to other algorithms.

AES S-Box: Why Not Use a Lookup Table

Scaling: a table for $\mathbb{F}_{2^n}$ requires $n * 2^n$ bits.

$n = 32$: a table uses $137,438,953,472$ bits of memory.

$n = 64$: a table uses $1,180,591,620,717,411,303,424$ bits.

Obfuscating a small table uses too many gates.

Our obfuscation techniques are applicable to other algorithms.

AES S-Box

$\mathbb{F}_2 = \{0, 1\}$. Addition is XOR. $1 + 1 = 0$. Multiplication is AND.

$\mathbb{F}_2[x]$: polynomials with coefficients that are 0 or 1.

$x^n + x^n = 0$ because $1 + 1 = 0$ in $\mathbb{F}_2$.

AES uses $m(x) = x^8 + x^4 + x^3 + x + 1$. $m(x)$ is irreducible in $\mathbb{F}_2[x]$.

$\mathbb{F}_2[x]/\big(m(x)\big)$ are polynomials in $\mathbb{F}_2[x]$ with degree ≤ 7.

AES S-box is built from polynomial multiplication in $\mathbb{F}_2[x]$.

An S-Box Example

Field $\mathbb{F}_2[x] \,/\, \big(m(x)\big)$ where $m(x) = (x^8 + x^4 + x^3 + x + 1)$.

Polynomials $p(x) = x^6 + x^4 + x$ and $q(x) = x^2 + 1$ are multiplied.

$$p(x) * q(x) = (x^6 + x^4 + x) + (x^8 + x^6 + x^3) \mod m(x).$$

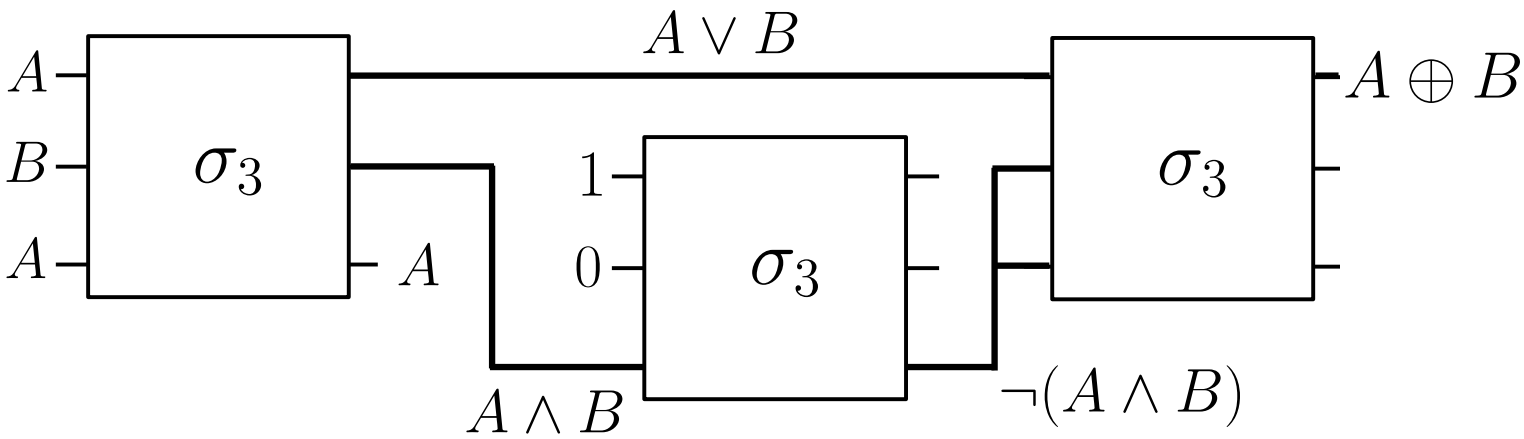
$$= (x^8 + x^6 + x^6 + x^4 + x^3 + x) \mod m(x).$$

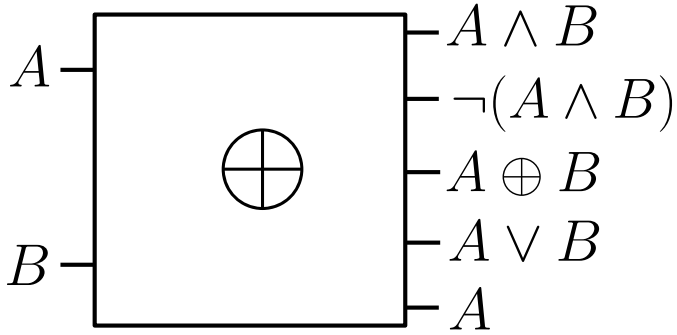
$$= (x^8 + x^4 + x^3 + x) + (x^8 + x^4 + x^3 + x + 1).$$

$$= (x^8 + x^8) + (x^4 + x^4) + (x^3 + x^3) + (x + x) + 1.$$

$$= 1.$$

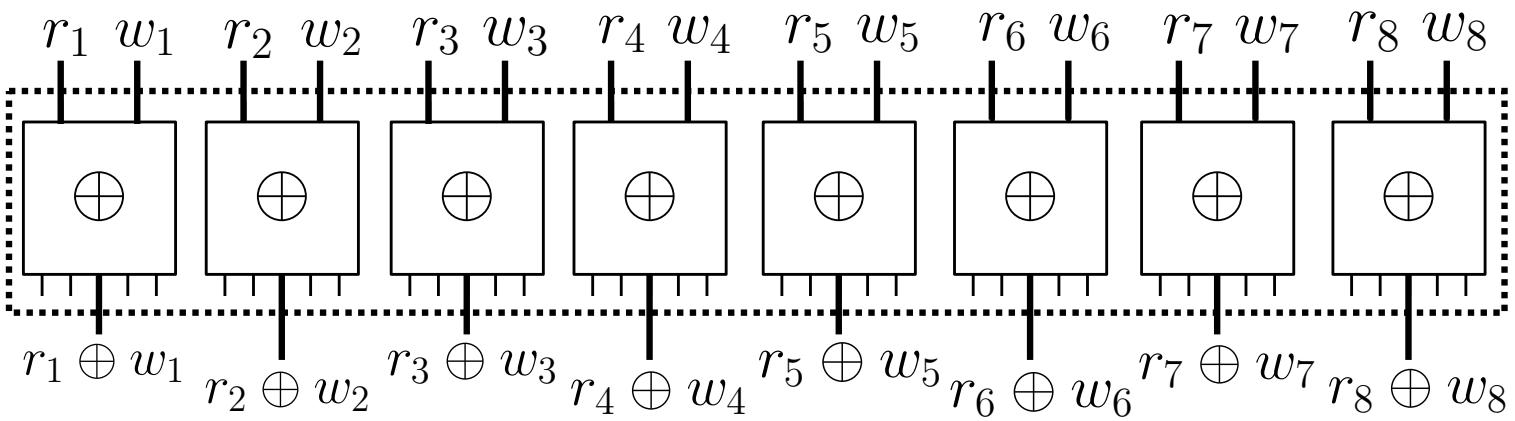
XOR Reversible Circuit $A \oplus B = (A \vee B) \wedge \neg(A \wedge B)$





Circuit Addition: Polynomials $r(x) + w(x)$

```
function  add_polynomials(r, w)
  return  [r1 ⊕ w1, r2 ⊕ w2, r3 ⊕ w3, r4 ⊕ w4, r5 ⊕ w5, r6 ⊕ w6, r7 ⊕ w7, r8 ⊕ w8]
end
```



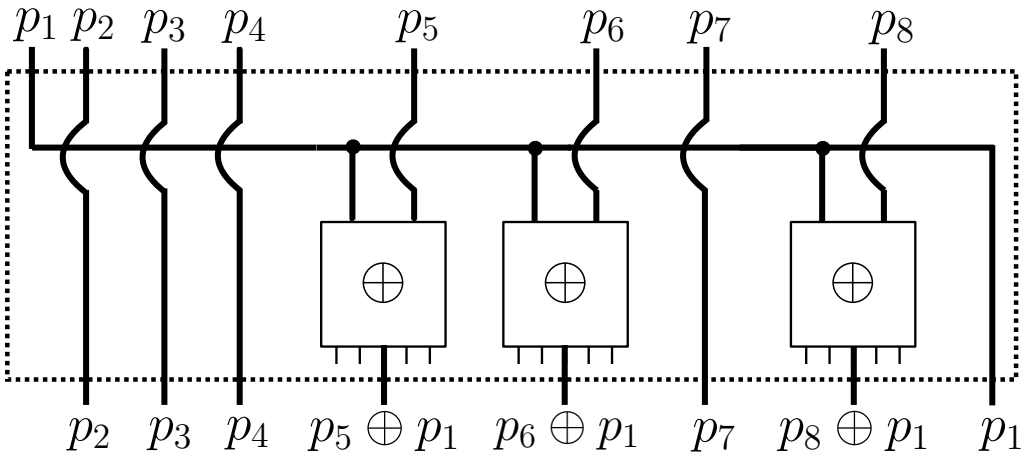
Circuit Multiplication: $x * p(x) \mod m(x)$

$p(x) = x^7 + x^5 + x^2 + 1$. $p_1 = p_3 = p_6 = p_8 = 1$. $p_2 = p_4 = p_5 = p_7 = 0$.

$m(x) = x^8 + x^4 + x^3 + x + 1$.

$x * p(x) \mod m(x) = (x^8 + x^6 + x^3 + x) + m(x)$

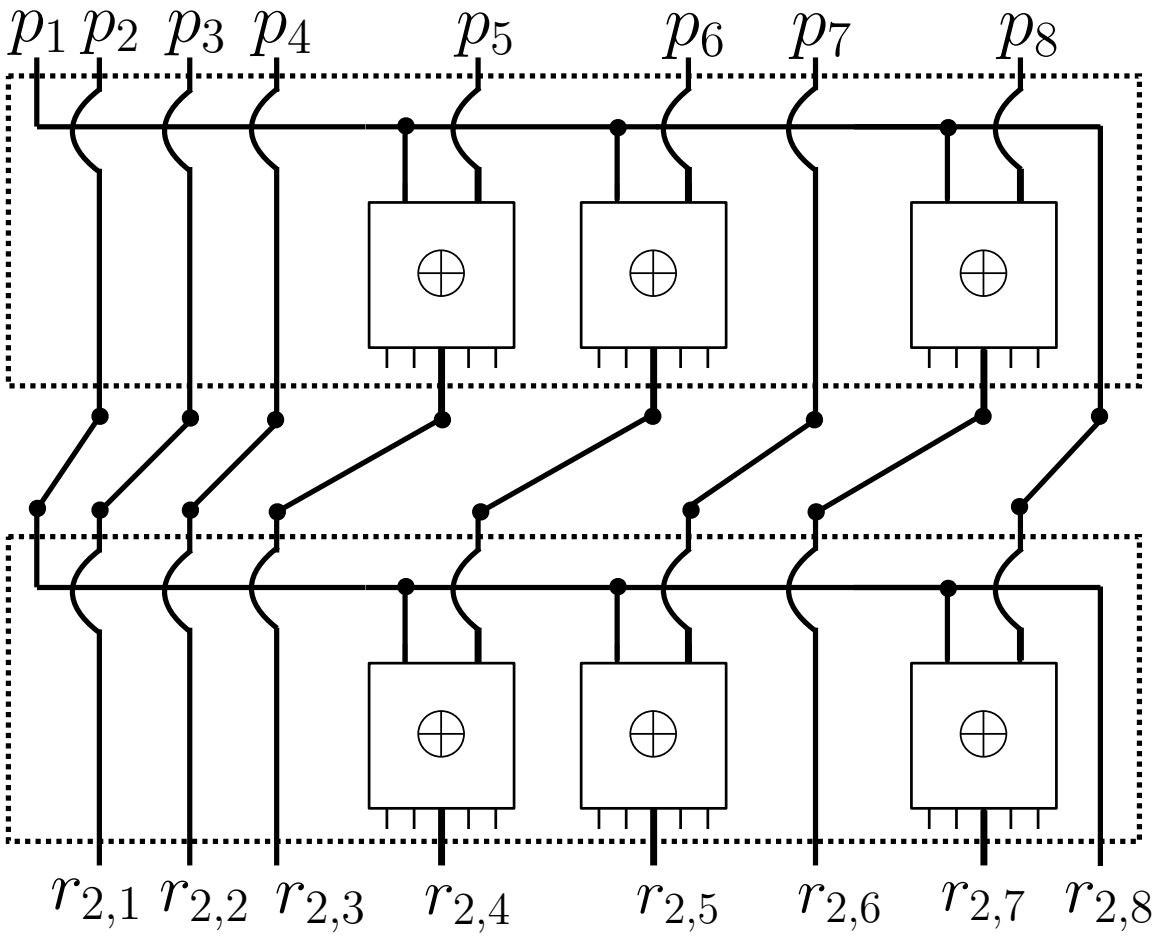
$= (x^8 + x^8) + x^6 + x^4 + (x^3 + x^3) + (x + x) + 1 = \boxed{x^6 + x^4 + 1}$



$p_2 x^7 + p_3 x^6 + p_4 x^5 + (p_5 \oplus p_1) x^4 + (p_6 \oplus p_1) x^3 + p_7 x^2 + (p_8 \oplus p_1) x + p_1$

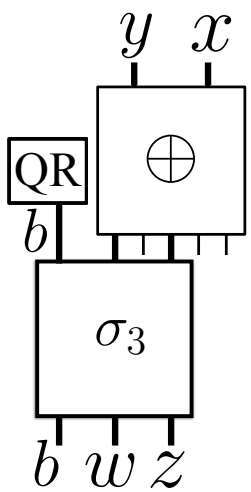
$= x^6 + (0 \oplus 1) x^4 + (0 \oplus 0) x^3 + (1 \oplus 1) x + 1 = \boxed{x^6 + x^4 + 1}$

Circuit Multiplication: $x^2 * p(x) \mod m(x)$ $n = 1, 2$



Random XOR Circuit

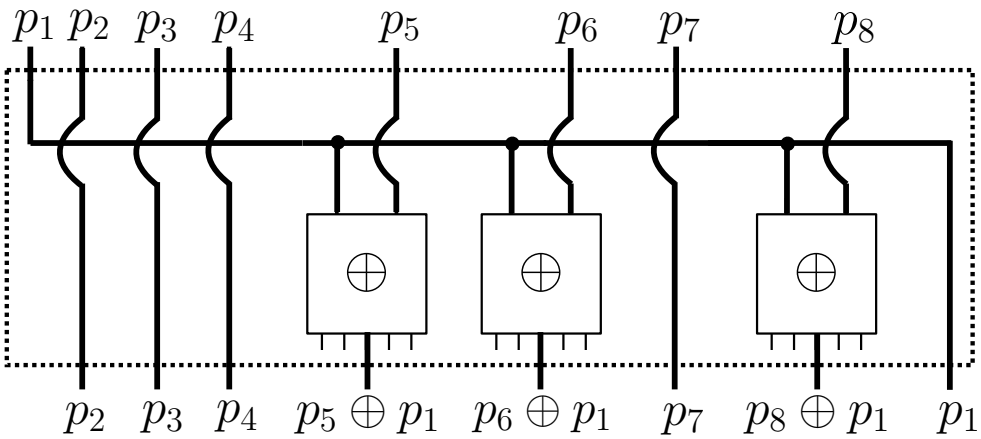
This circuit unpredictably moves output bit $x \oplus y$ via random bit b.

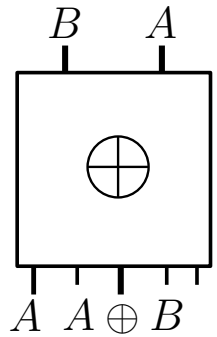


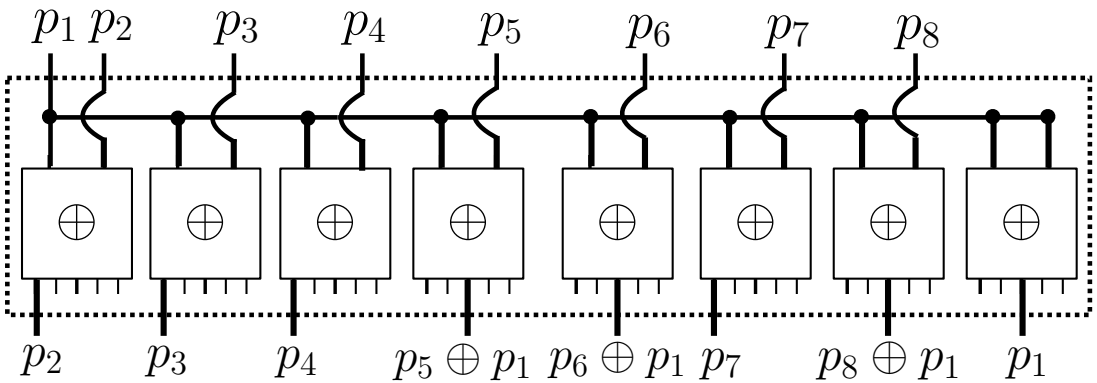
If $b = 1$, then $z = x \oplus y$ and $w = x$.

If $b = 0$, then $z = x$ and $w = x \oplus y$.

$$\text{Multiplication } x * p(x) = x \sum_{i=1}^{8} p_i x^{8-i} \mod m(x)$$







Circuit Obfuscation: $x * p(x)$ Swaps p_4 and p_5

$b = 1$: $u = p_4$. $v = p_4 \oplus p_1$. $w = p_5$. $z = p_5 \oplus p_1$.

$b = 0$: $u = p_5 \oplus p_1$. $v = p_5$. $w = p_4 \oplus p_1$. $z = p_4$.

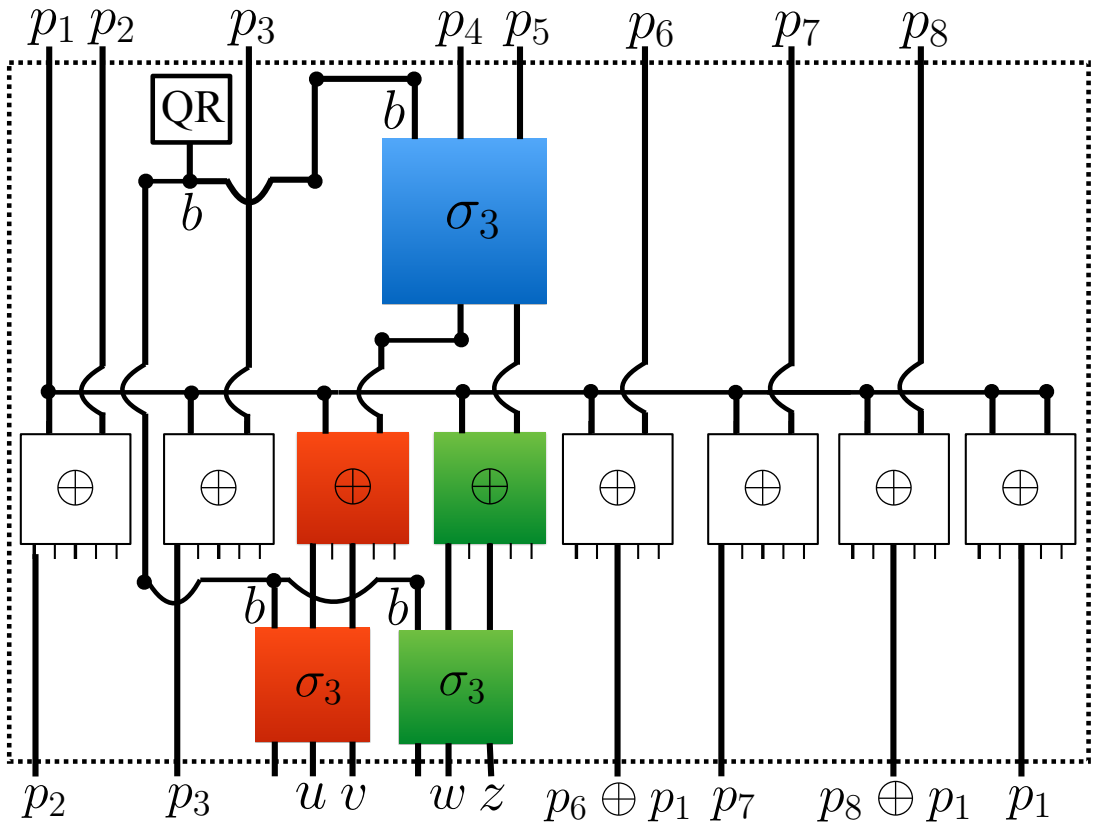


Fig. 21 Circuit Deobfuscation: $x * p(x)$ with u and z

$b = 1$: $u = p_4$. $z = p_5 \oplus p_1$. $b = 0$: $u = p_5 \oplus p_1$. $z = p_4$.

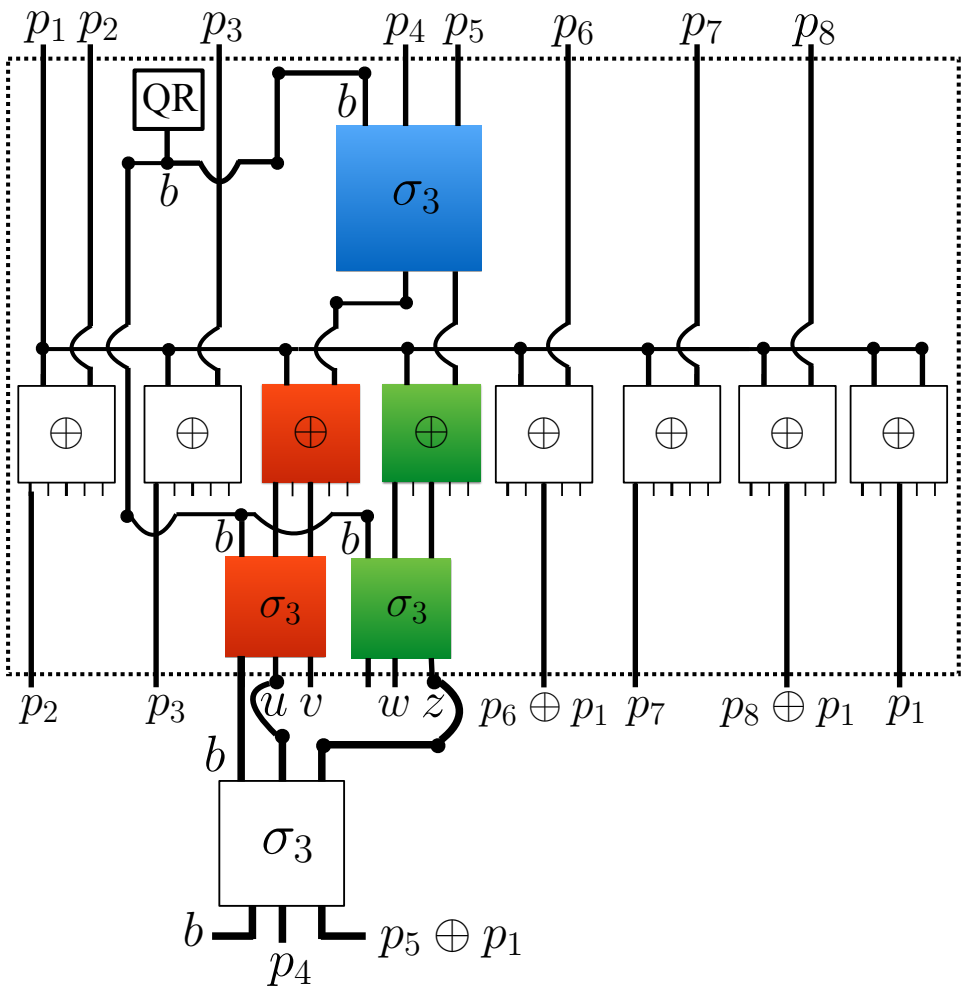
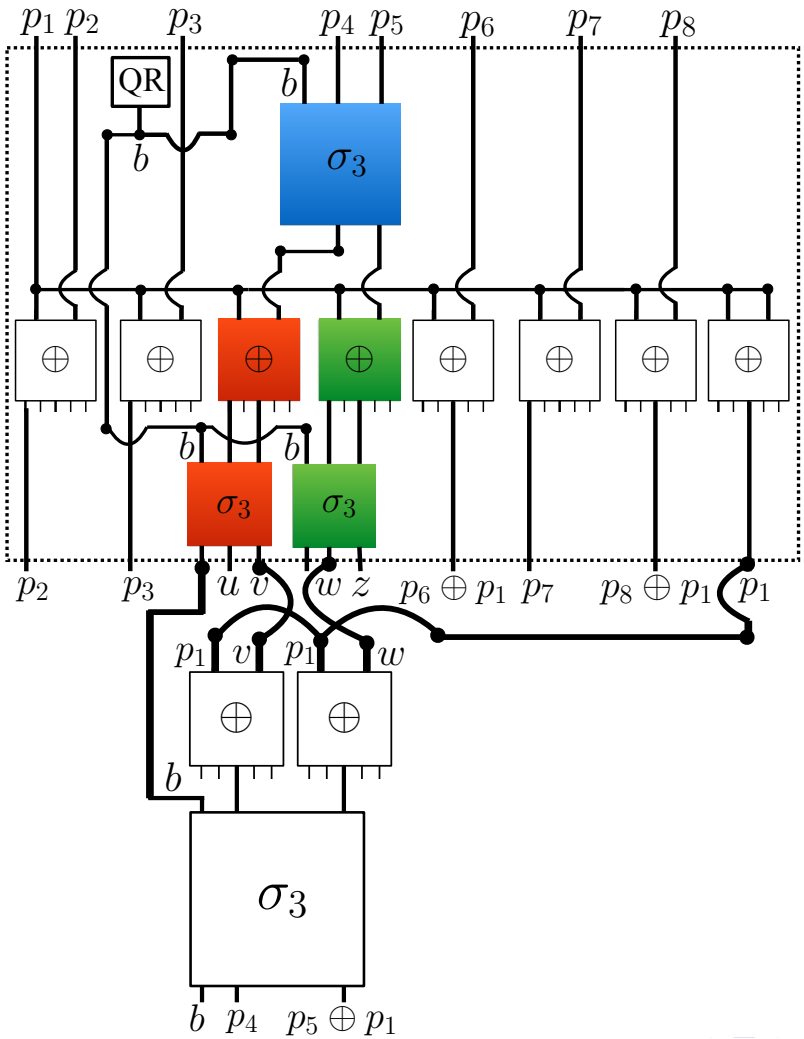
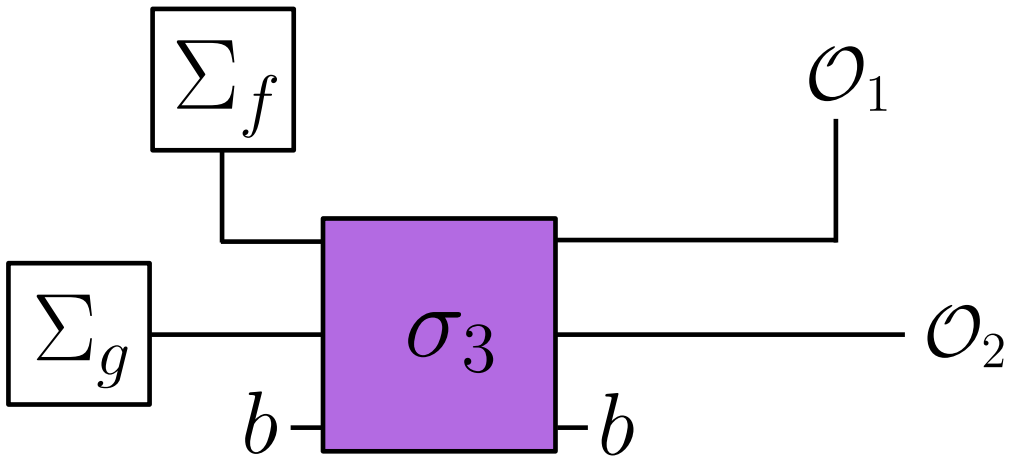


Fig. 22 Circuit Deobfuscation: $x * p(x)$ with v and w

$b = 1$: $v = p_4 \oplus p_1$. $w = p_5$. $b = 0$: $v = p_5$. $w = p_4 \oplus p_1$.



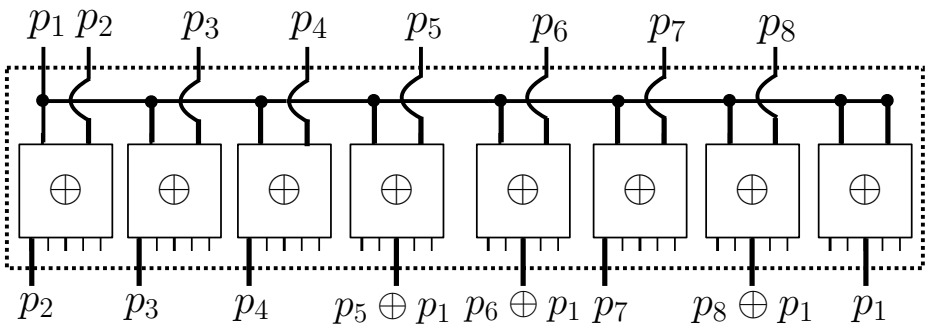
General Method of Unpredictably Moving Computation



A proprietary algorithm can be hidden in the obfuscated circuit.

Obfuscation Complexity of AES S-box

Seven circuit layers multiply two polynomials in $\mathbb{F}_2/\big(m(x)\big)$



An arbitrary permutation on n bits needs at most $n-1$ swaps.

AES S-box: $n = 8$. A σ_3 gate swaps when $S = 0$.

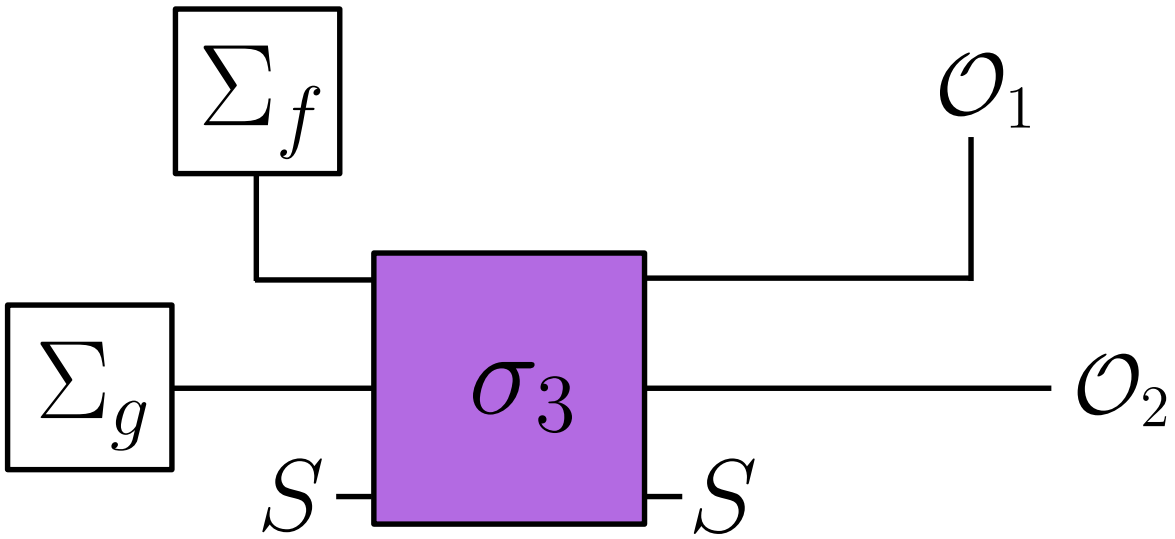
Reversible XOR gate requires three σ_3 gates.

1 swap: Fig. 21 and 22 uses 4 and 10 extra σ_3 gates, respectively.

Complexity: 7 layers $*$ (7 swaps/layer) $*$ 10 $=$ 490 σ_3 gates.

Summary I

Extends Reversible Computing. Reconfigures circuits.



Randomness and reversible gates can unpredictably move the physical location of computation across the circuit.

Summary II

New type of obfuscation: uses physics and space-time.

Doesn't rely on computational complexity and one-way functions.[2]

S-box techniques are applicable to any algorithm as σ_3 is universal.

Hinders theft of a proprietary algorithm.

Hinders cryptanalysis attacks: acoustic, power, timing attacks.

[2] No math proof that one-way functions exist.