# Towards
# Security Without Secrets

**Srini Devadas**

**Massachusetts Institute of Technology**

**Joint work with**

**Charles Herder and Marten van Dijk**

# Modern Cryptography Requires Secrets

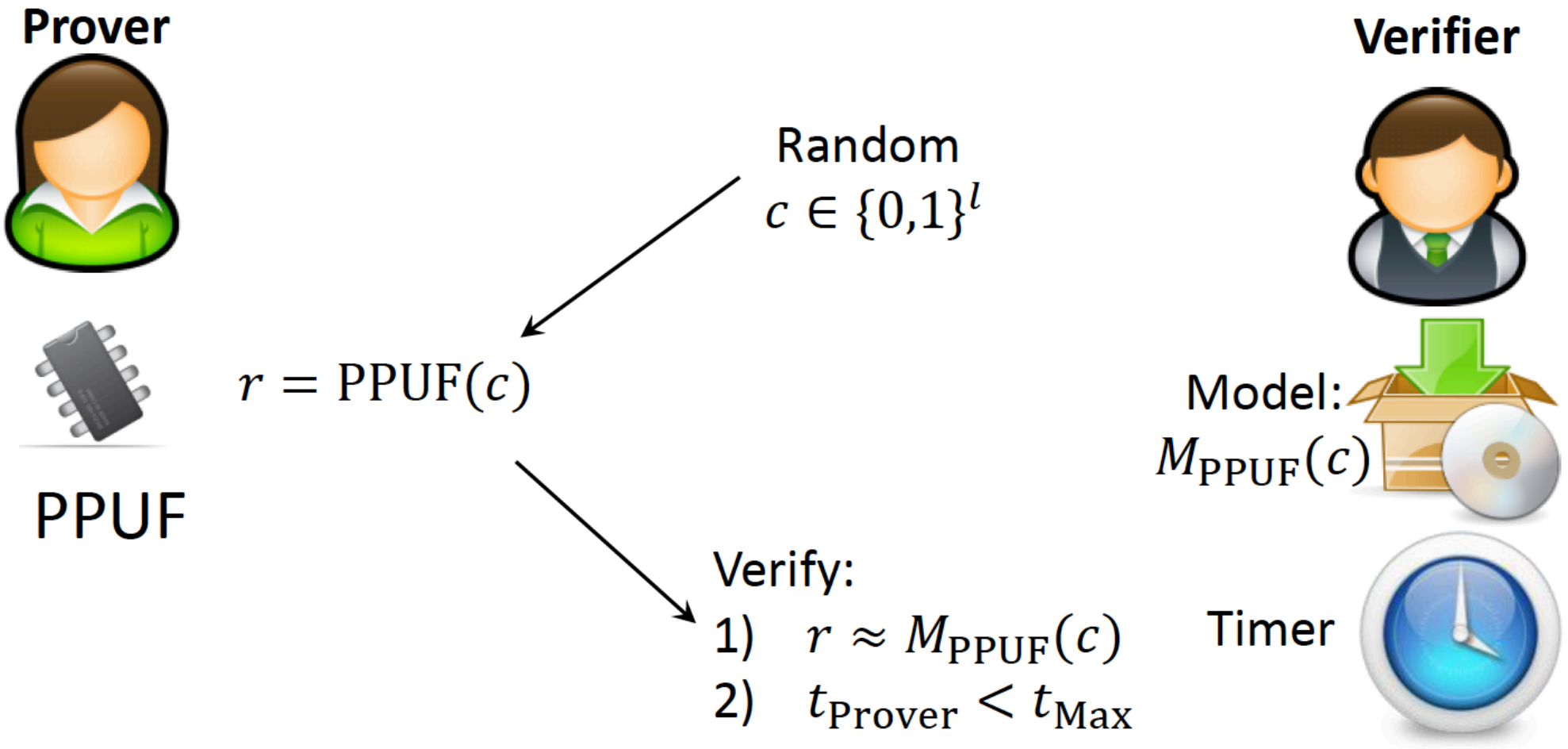
- Something you know/have/are
- Loss/Theft of secret is *permanent*
  - Can't use password/key/device/biometric *ever again*
- Is this the only way?

# Public-Model Physical Unclonable Functions

Notion due to

U. Ruhrmair, M. Potkonjak, F. Koushanfar

# Public Model PUFs

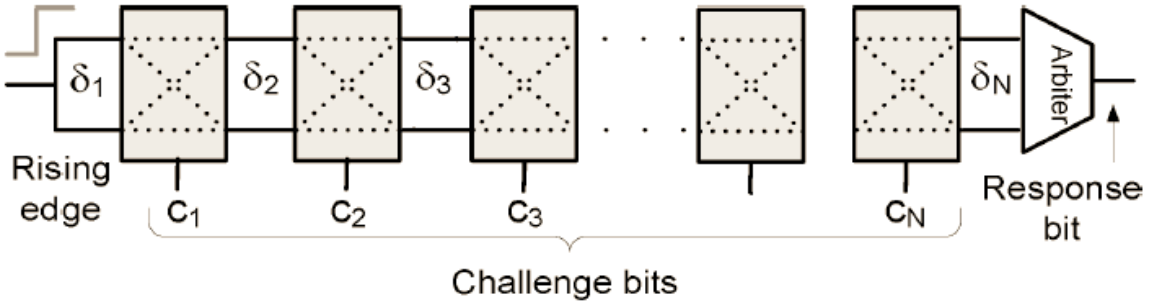



Key Recognition:

Compute time of software model > hardware PPUF

# PPUF Proposal: FPGA Time-Bounded-Authentication

- Standard Arbiter PUF
- Combinatorial Delay faster than software model

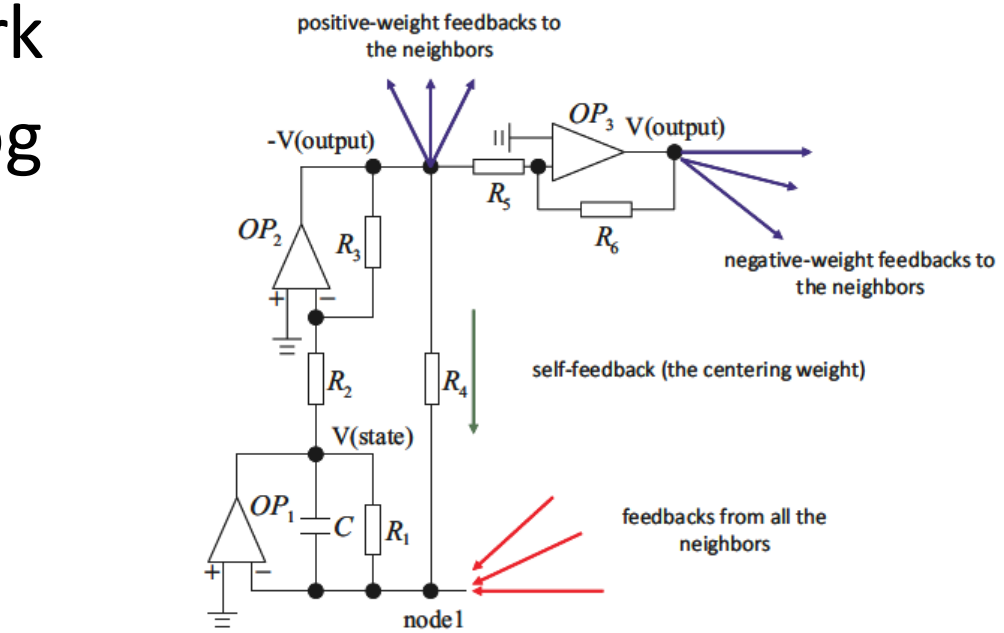



- FPGA with programmable delay lines breaks this.

# PPUF Proposal: SIMPL

Cellular Neural Network

- Square grid of Analog circuits

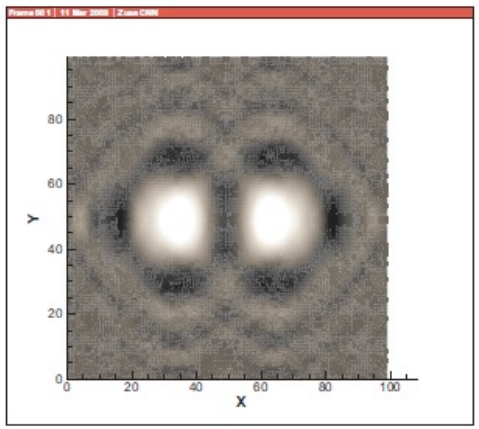


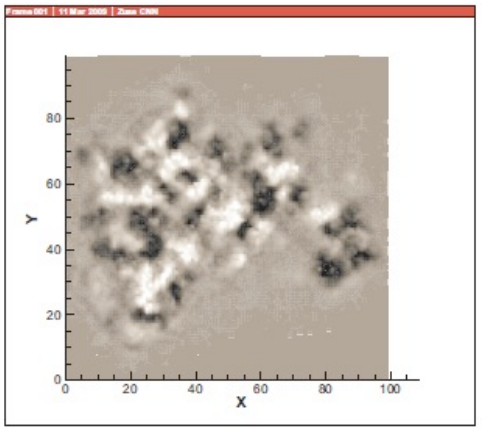

- Unreliable:

  Chaos

- Broken:

  Parallel CMOS model



# Informal Criteria for PPUFs - 1

- The PPUF hardware can compute **some** function with inputs and some measurable state
- The **non-CMOS** PPUF hardware will only evolve forward in time according to its physical laws
- The CMOS PPUF model will then be a direct physical simulation of the PPUF hardware itself but provably **slower**

# Classical 'Speedups'
## (quantum computers are far away!)

- Classical Complexity-Theoretic Church-Turing Thesis
  - "Classical Systems 'efficiently' simulatable by classical computers"

- A stronger statement:
  - Fully parallel CMOS computer can simulate a classical physical system with at most CONSTANT time slowdown

# Informal Criteria for PPUFs - 2

- Computational speedup can be derived from **internal parallelism** and the speed **of internal system dynamics**

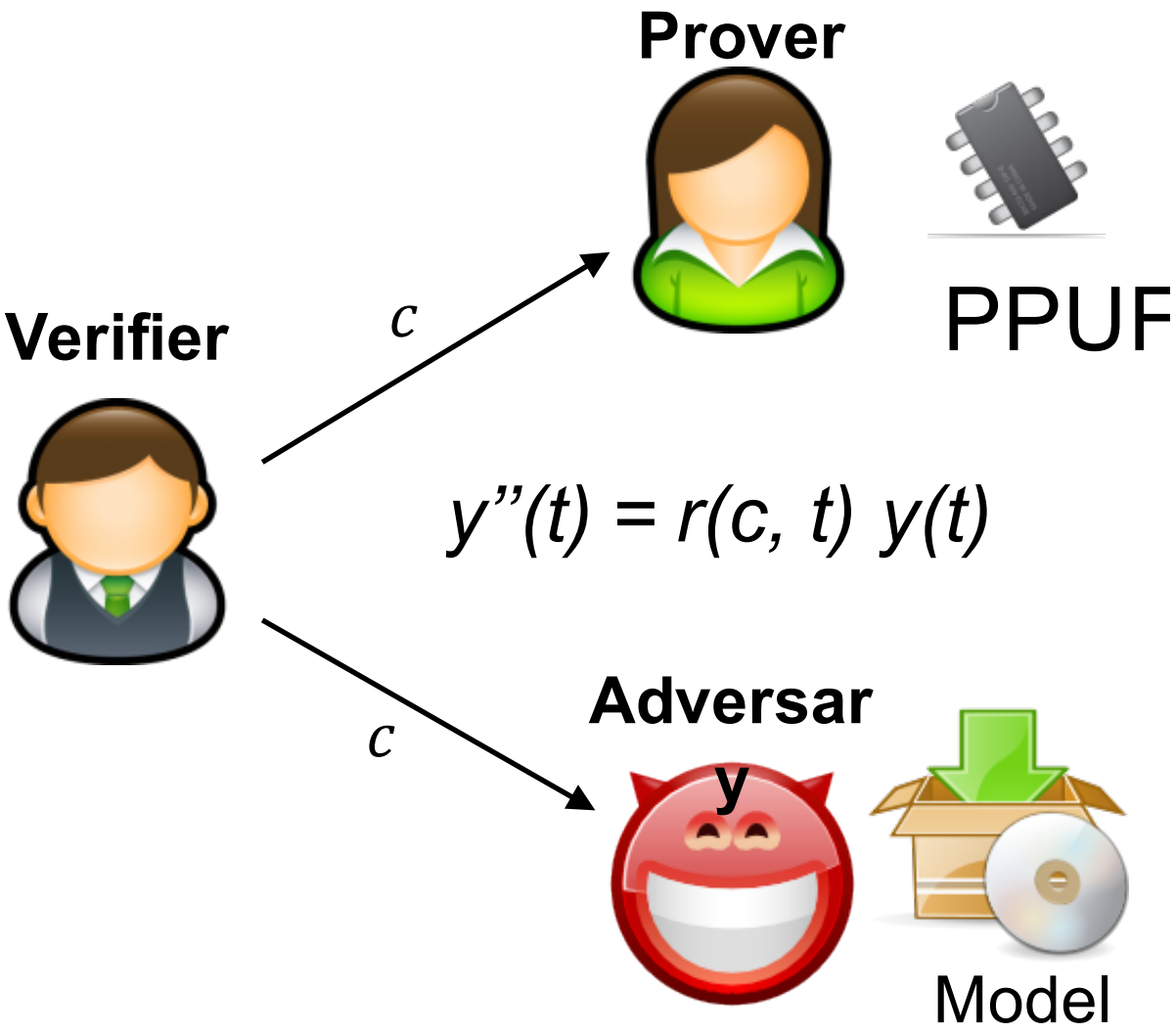
- Since CMOS provides abundant parallelism, the non-CMOS PPUF hardware internal dynamics must be faster than the dynamics of the CMOS model

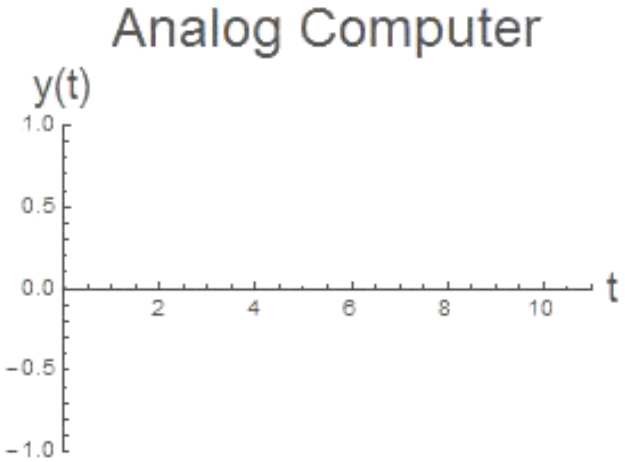
# Informal Criteria for PPUFs - 3

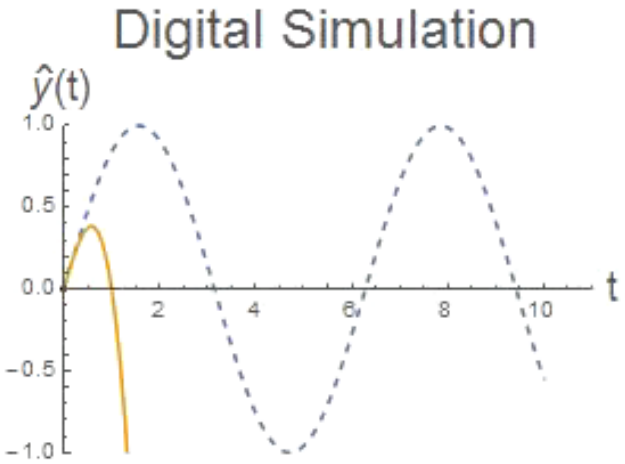
- Need to **restrict parallelization** of CMOS model simulation

- The computational problem that the PPUF hardware solves must have the property that **any digital circuit has an asymptotic lower bound on time complexity of $\Theta(t_f - t_s)$**

# PPUF Idea: Differential Equations



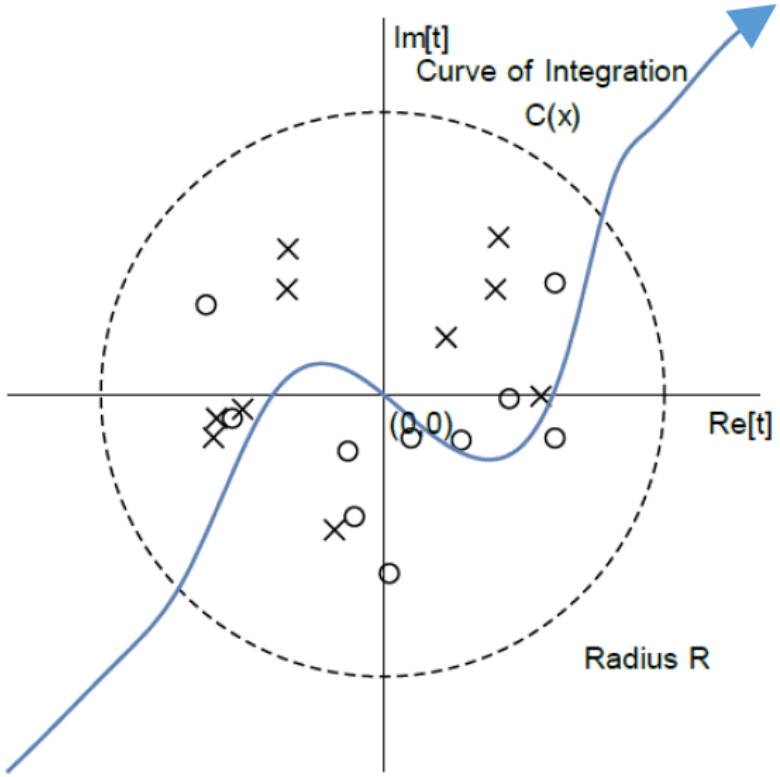



# ODE family

$$y''(t) = r(t)y(t)$$
$$r(t) \in \mathbb{C}(t)$$

- Kovacic algorithm determines if $y(t)$ has closed form (Liouvillian)

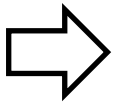
- Finite representation
  - Finite number of poles/zeroes: $m$
  - Inside disc of radius $R$
  - Measurement Precision $\pm\epsilon$

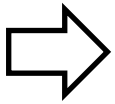


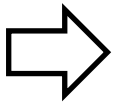
# Challenges

**What needs to be shown?**

- No closed-form solutions for $y(t)$
- Large challenge/response space
- Bound on step size of *all* numeric integrators







**How do we show it?**

- Kovacic Algorithm
- ODE Formalism
- Conjectural Statement + Reduction

**Asymptotic Statement:**

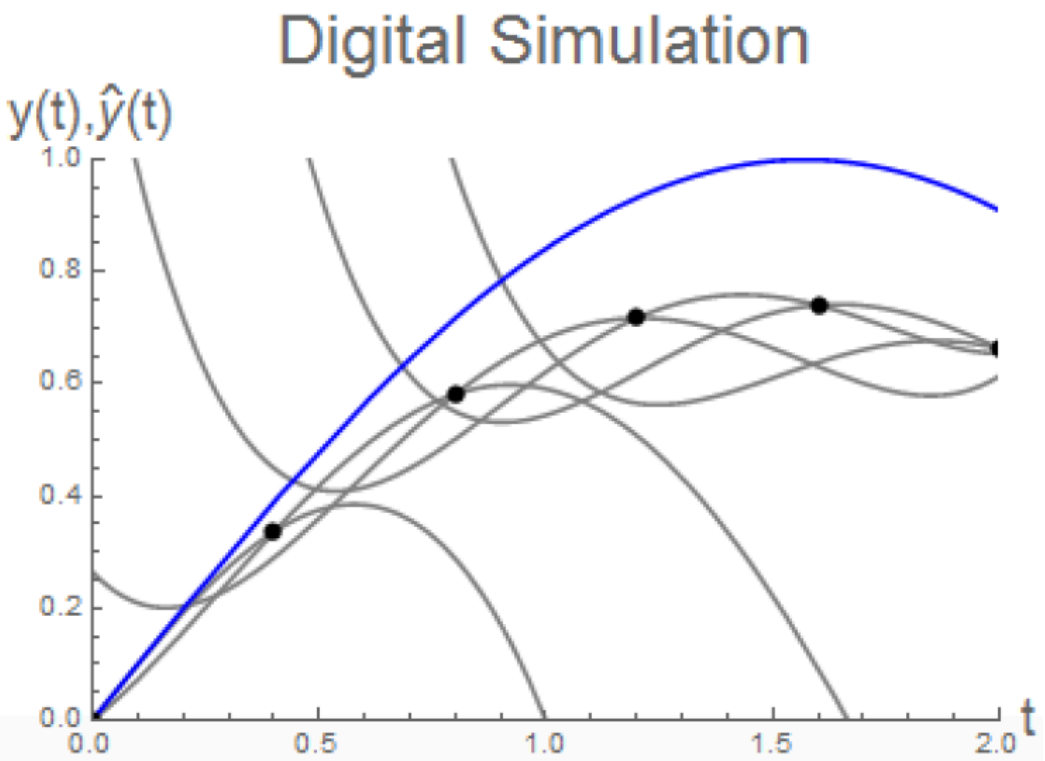
PPUF runs for time $t$ implies $\Theta(t)$ simulator steps

# Primary Conjecture (informal)

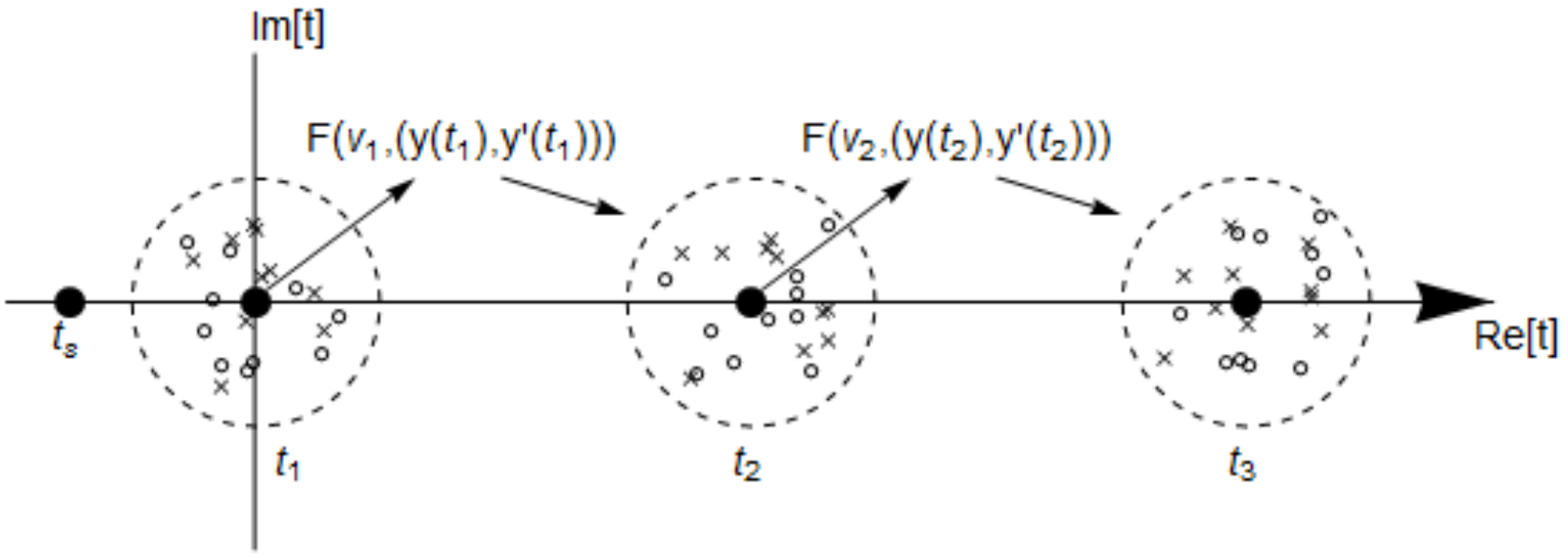
- All numeric integrations are iterated asymptotic expansions.
- Expansions are to order $k$
  - $k-2$ derivatives of $r(t)$ at the point of expansion.
  - Estimates higher order based on problem description.
- Expansions *don't use* a global knowledge of $r(\mathrm{t})$.
  - $r(t)$ is a random variable with increasing entropy far away from the point of expansion
  - Similar for $y(t)$

E.g., Forward Euler ($k = 2$)

$$\hat{y}(t+\Delta t) = \hat{y}(t) + \hat{y}'(t)\Delta t$$
$$\hat{y}'(t+\Delta t) = \hat{y}'(t) + r(t)\hat{y}(t)\Delta t$$



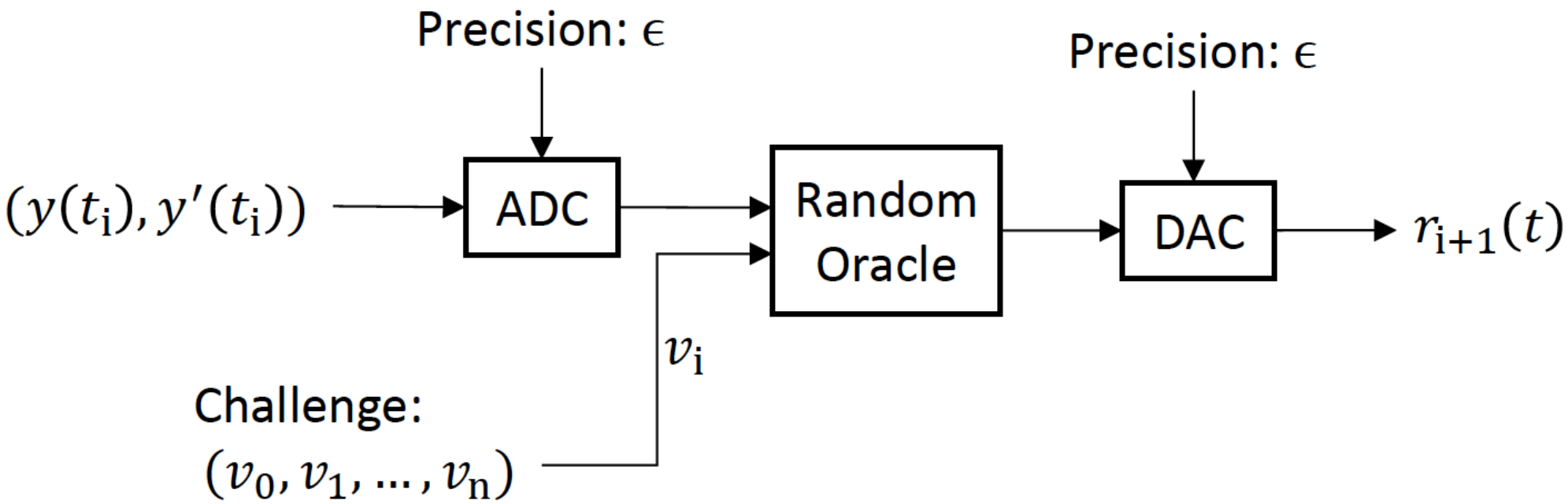
# PPUF Construction




- Multiple pole/zero "clusters" around $t_1$, $t_2$, …
- Integrate along $t_s$ along real axis through n clusters

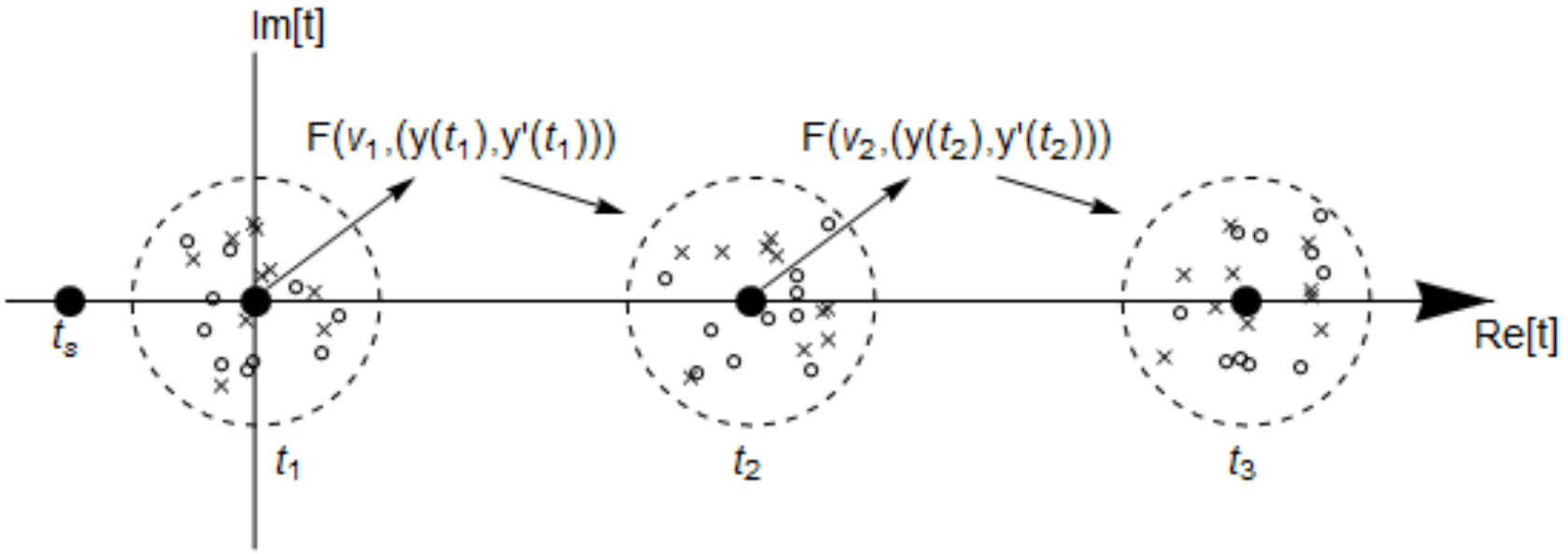
# Feed-Forward Nonlinearity

Feedforward Function: $F\left(v_i, \left(y(t_i), y'(t_i)\right)\right)$




- Hybrid analog/digital

# PPUF Construction




- Multiple pole/zero "clusters" around $t_1$, $t_2$, …
- Integrate along $t_s$ along real axis through n clusters
- **Lemma: "Can't jump over clusters."**

# Physical Requirements

$$|t_{i+1} - t_i| + t_{\text{ADC}} + t_{\text{DAC}} < \text{Runtime(Expand)}$$

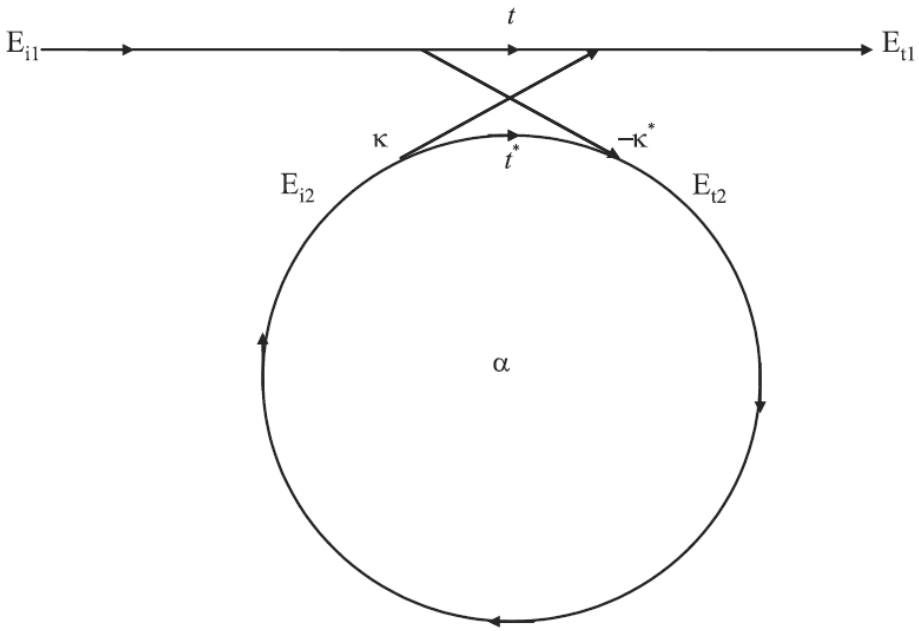
- Lower-bound Runtime(Expand) by cost of arithmetic in CMOS

# Thoughts on Implementation: Optical Ring Resonators

Coupling relationship:

$$\begin{pmatrix} E_{t1} \\ E_{t2} \end{pmatrix} = \begin{pmatrix} t & \kappa \\ -\kappa^* & t^* \end{pmatrix} \begin{pmatrix} E_{i1} \\ E_{i2} \end{pmatrix}$$

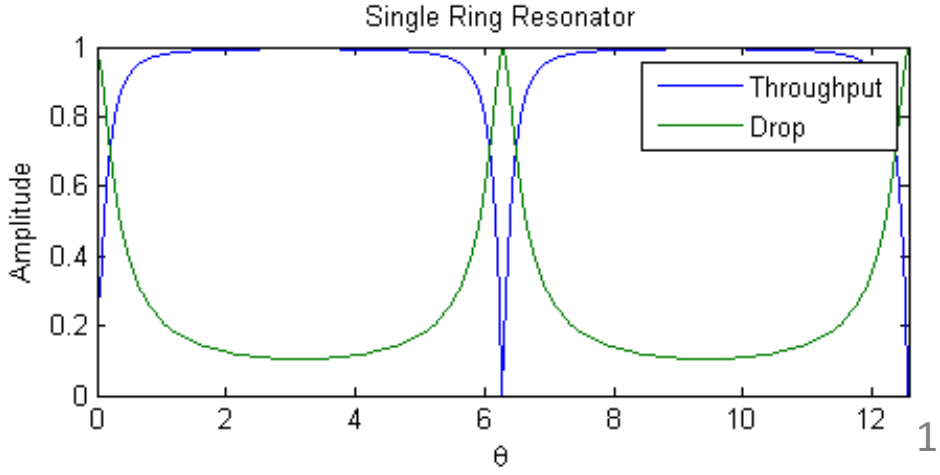
$$E_{i2} = \alpha E_{t2}$$




Recurrence relationship:

$$E_{i2}[n] = \alpha t^* E_{i2}[n-1] - \alpha\kappa E_{i1}[n-1]$$

$$E_{t1}[n] = \kappa E_{i2}[n] + t E_{i1}[n]$$



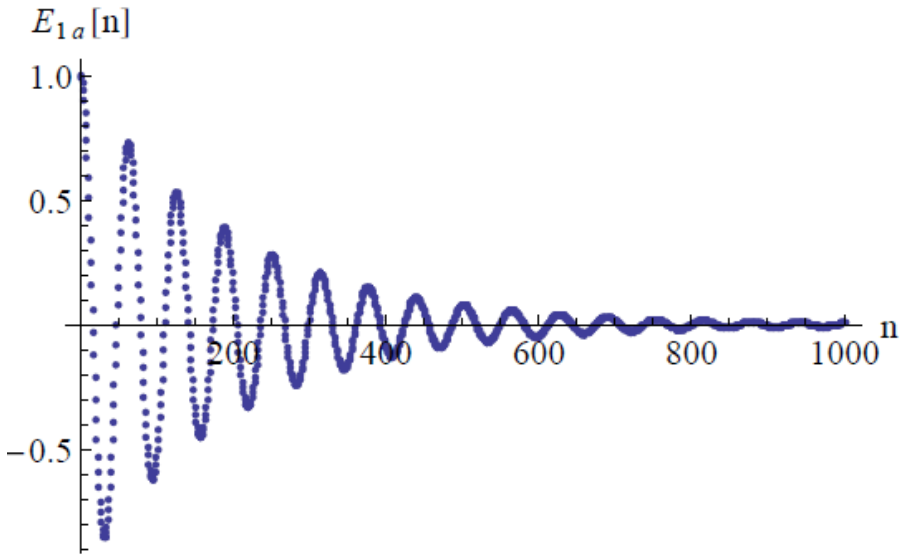
# Coupled Ring Resonator

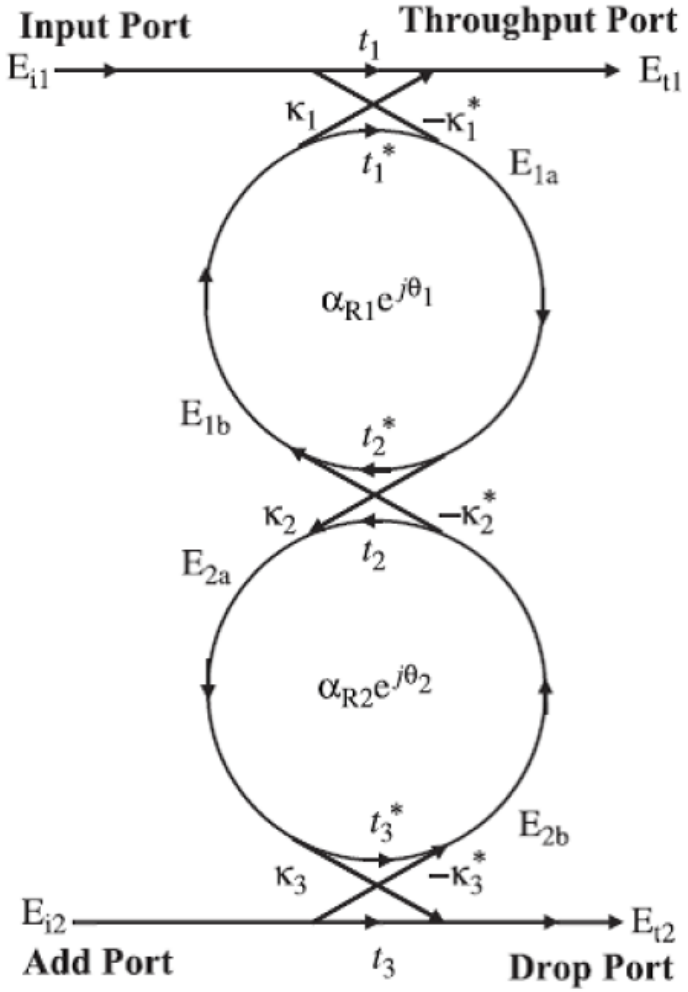
## Recurrence Relation

$$E_{1a}[n] = t_1^* t_2^* \alpha_1 e^{j\theta_1} E_{1a}[n-1] - t_1^* \sqrt{\alpha_1} e^{j\theta_1/2} \kappa_2^* \sqrt{\alpha_2} e^{j\theta_2/2} E_{2b}[n] - \kappa^* E_{i1}[n]$$

$$E_{2b}[n] = t_3^* \sqrt{\alpha_2} e^{j\theta_2/2} \kappa_2 \sqrt{\alpha_1} e^{j\theta_1/2} E_{1a}[n-1] + t_3^* t_2 \alpha_2 e^{j\theta_2} E_{2b}[n-1]$$

## Discrete Impulse Response



# Implementation Challenges

**Implementation**

- CMOS Speedup
  - Readout speed for feed-forward (ADC)
  - Modulation speed (DAC)
- Noise!

**Proving Security of Physical Implementation**

- Generality of Theoretical Model
- Physical Limitations

# Summary

- PPUF Formalism
  - ODE as a computational problem for PPUFs
  - Formal Conjecture
  - Formal Security Definition
- Analog Computing (Not presented)
  - Application of PPUF formalism in broader context of analog computing