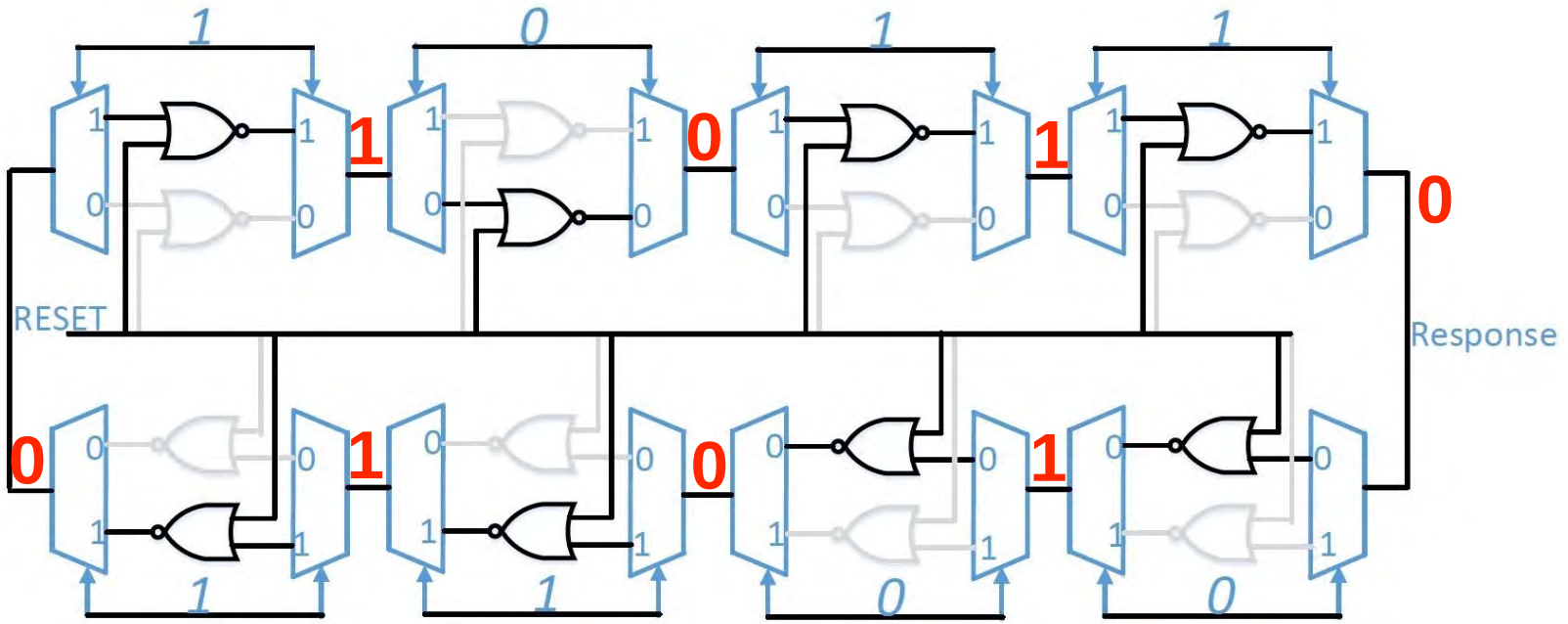
Example

- Challenge bits select weights, stage index determines signs
- Response tells whether sum is negative or positive
- Additive delay model (like Arbiter PUF)

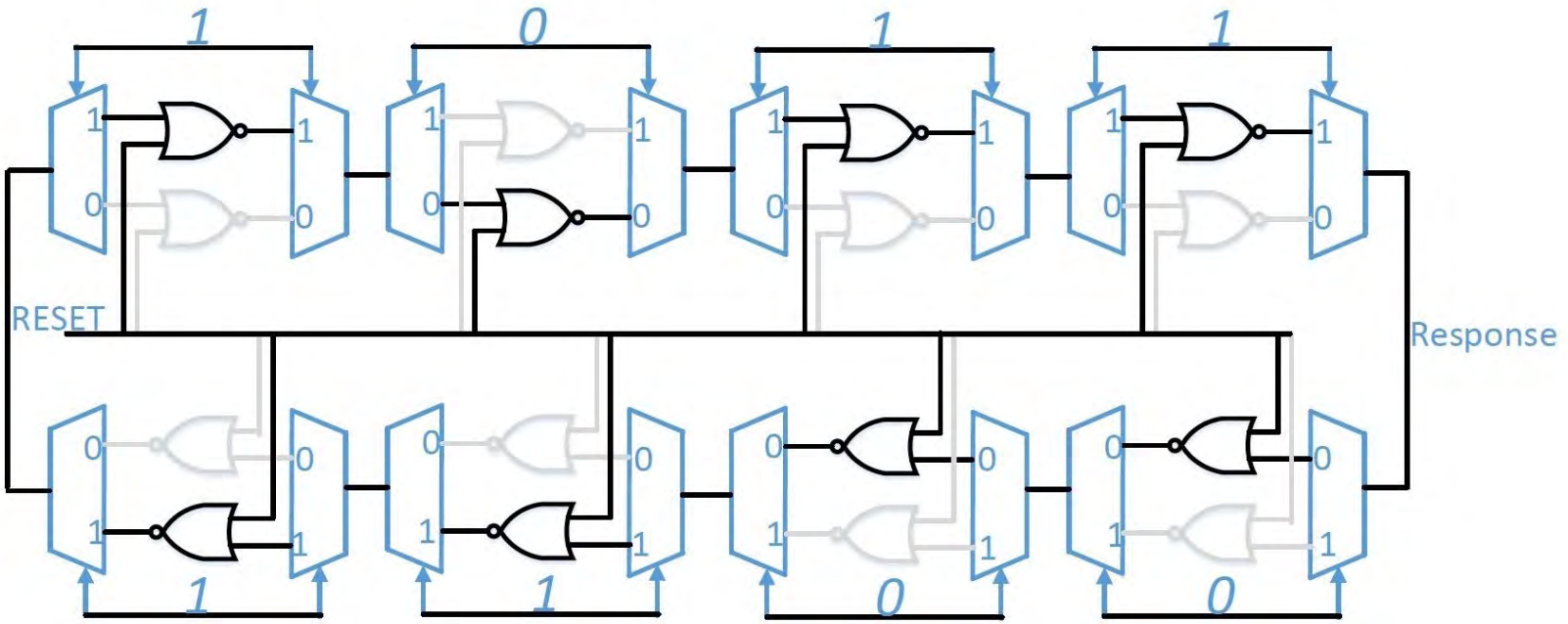
$$t_0 - b_1 + t_2 - t_3 + b_4 - b_5 + t_6 - t_7$$



Example

- Challenge bits select weights, stage index determines signs
- Response tells whether sum is negative or positive
- Additive delay model (like Arbiter PUF)

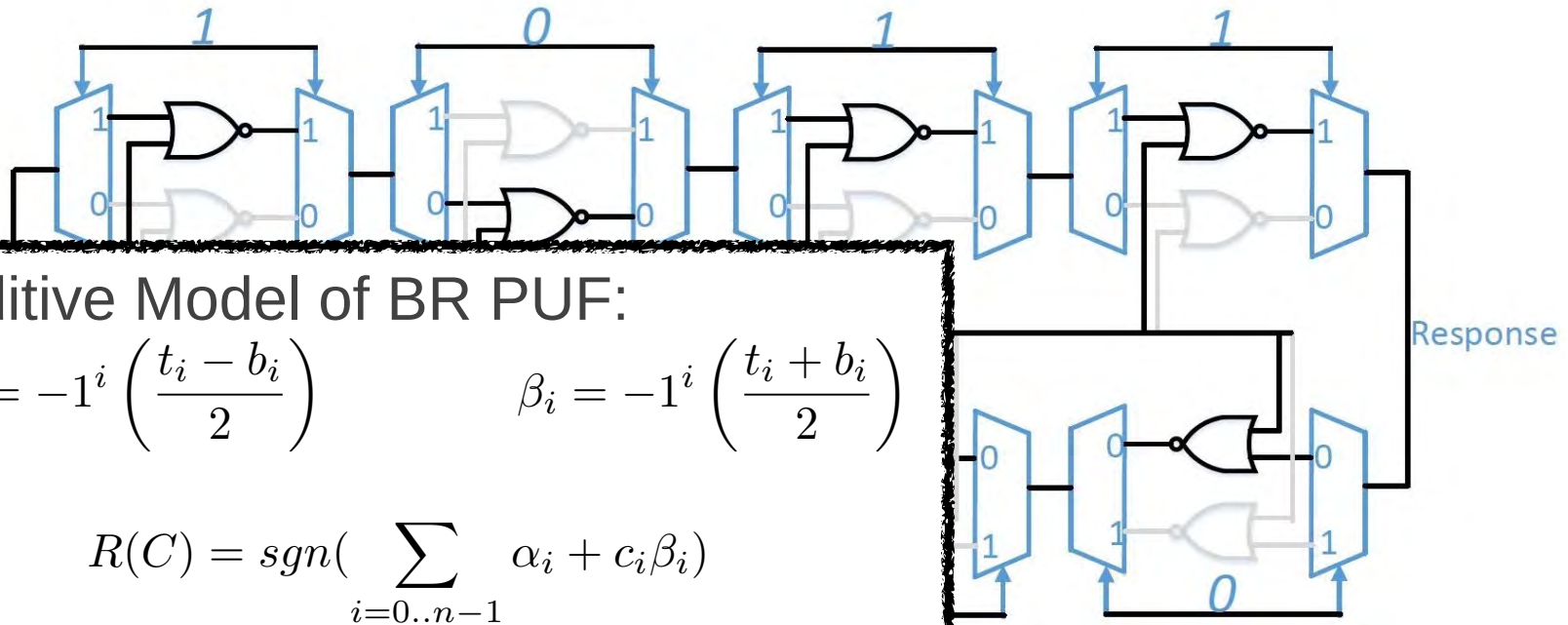
$$t_0 - b_1 + t_2 - t_3 + b_4 - b_5 + t_6 - t_7$$



Example

- Challenge bits select weights, stage index determines signs
- Response tells whether sum is negative or positive
- Additive delay model (like Arbiter PUF)

$$t_0 - b_1 + t_2 - t_3 + b_4 - b_5 + t_6 - t_7$$



Additive Model of BR PUF:

$$\alpha_i = -1^i \left(\frac{t_i - b_i}{2} \right) \qquad \beta_i = -1^i \left(\frac{t_i + b_i}{2} \right)$$

$$R(C) = sgn(\sum_{i=0..n-1} \alpha_i + c_i \beta_i)$$

see also Schuster et al. Trust 2014

Implementation of SVM Modeling Attacks

- Modeling with Support Vector Machines classification
- CRPs from FPGA implementation, SVM attacks use Matlab
- CRPs divided into training and validation datasets:

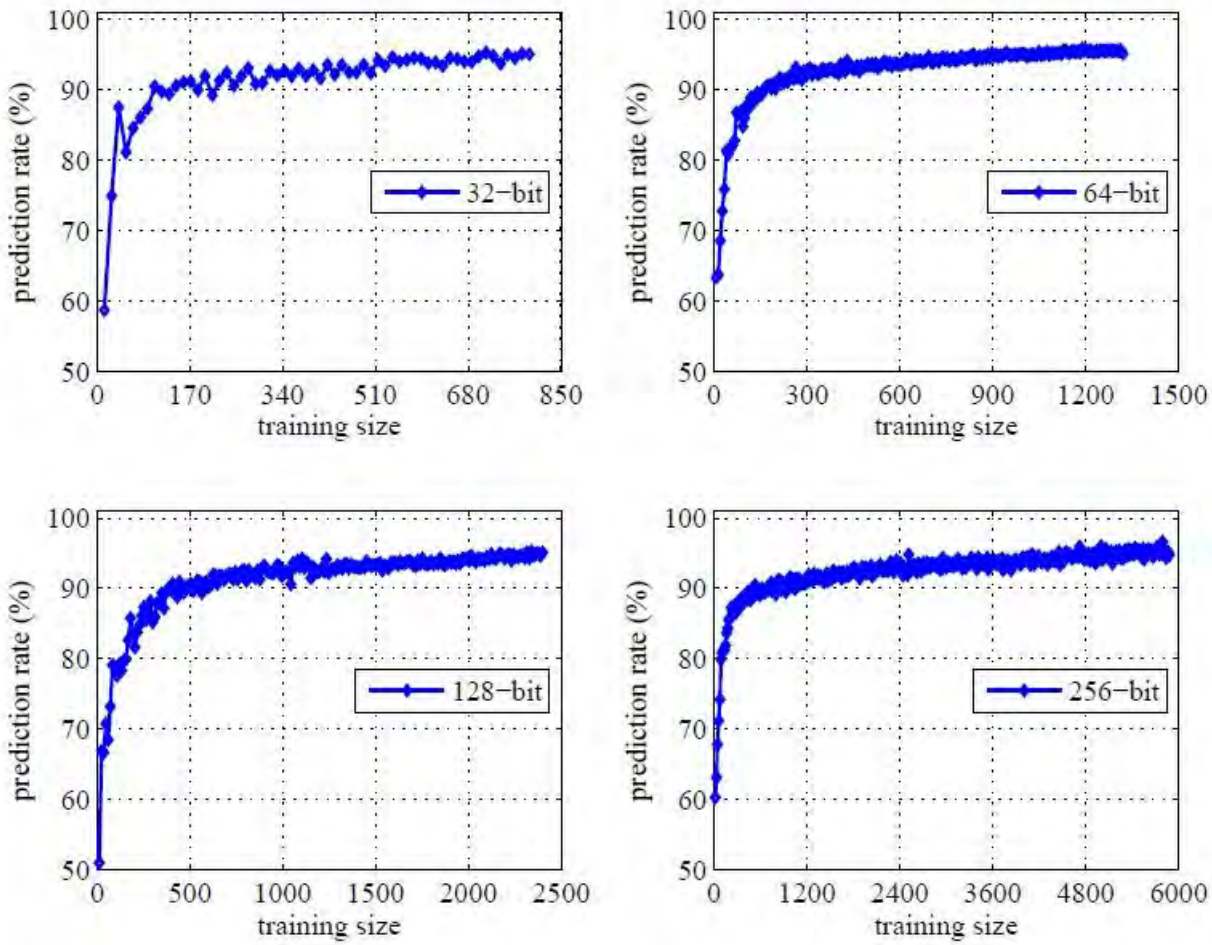
- **Train** the PUF model:

```
PUF_model=svmtrain(training_input, training_output, 'options',
'kernel_function', 'polynomial', 'polyorder', number_of_XOR); %%
polynomial kernel is used, while the polyorder is the XOR complexity, i.e.,
for a single BR PUF, number_of_XOR=1
```

- **Validate** the PUF model:

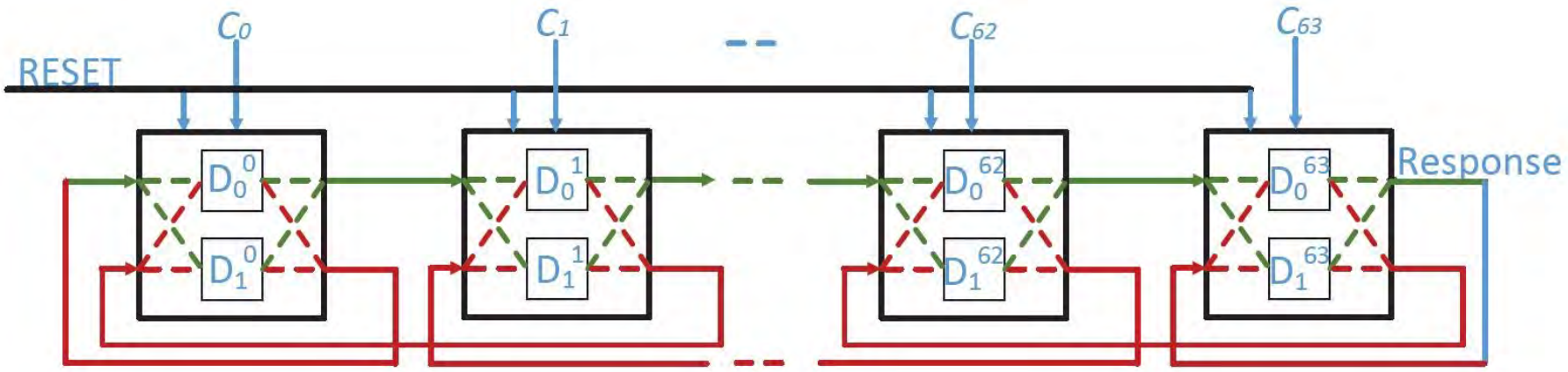
```
model_output= svmclassify(PUF_model,validation_input);
prediction_rate=(model_output==validation_output); %% predication rate
is the percentage of model_output equals with that of validation output
```

BR PUF is Not Secure



Twisted BR PUF

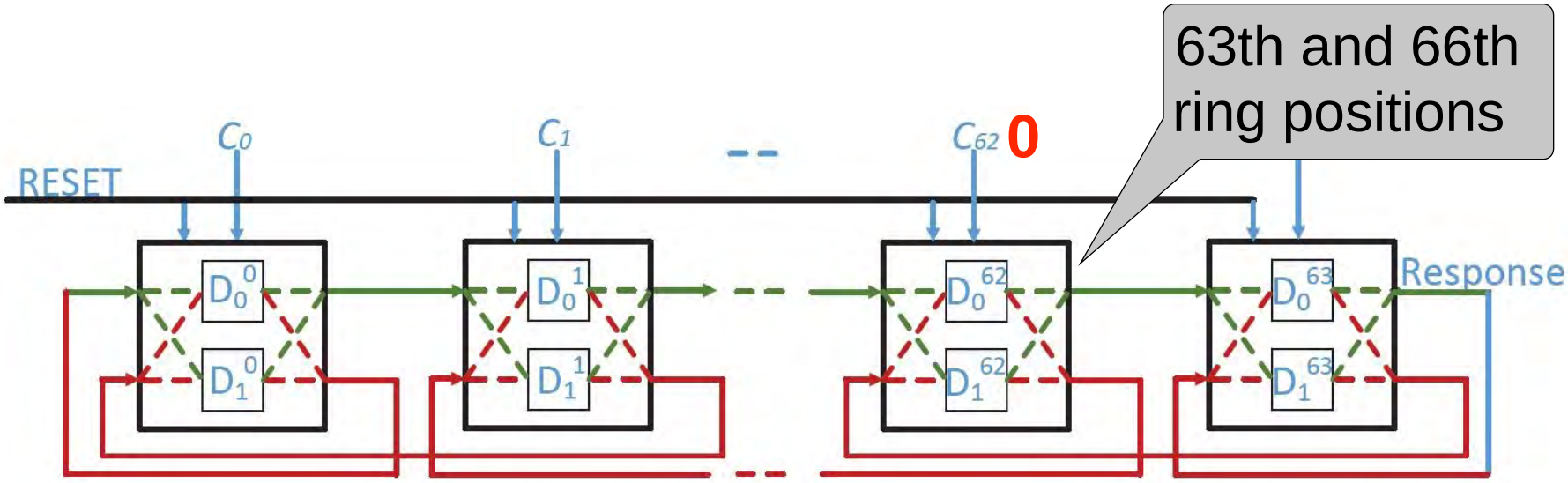
- TBR-PUF(6) has a more compact design
- All 2n inverting elements used in each ring
- Challenge bit determines whether ring position of each inverting element is even or odd
- Additive model still applies and is simpler than regular BR PUF



(6) D. Schuster, et al. *Trust and Trustworthy Computing* 2014

Twisted BR PUF

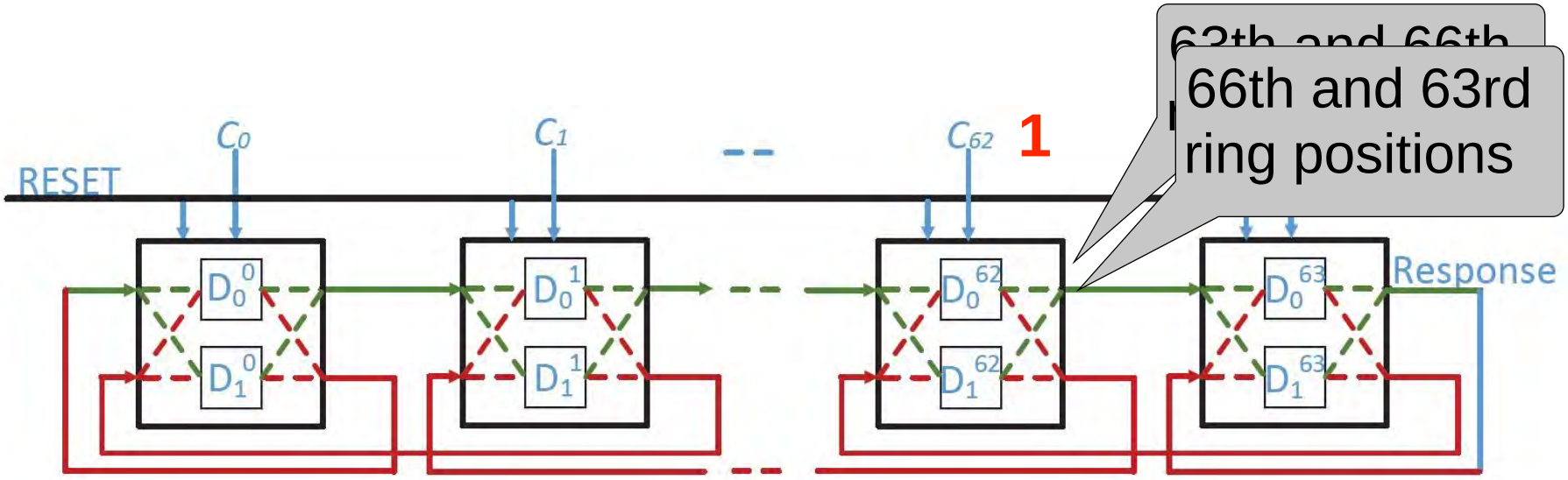
- TBR-PUF(6) has a more compact design
- All 2n inverting elements used in each ring
- Challenge bit determines whether ring position of each inverting element is even or odd
- Additive model still applies and is simpler than regular BR PUF



(6) D. Schuster, et al. *Trust and Trustworthy Computing* 2014

Twisted BR PUF

- TBR-PUF(6) has a more compact design
- All 2n inverting elements used in each ring
- Challenge bit determines whether ring position of each inverting element is even or odd
- Additive model still applies and is simpler than regular BR PUF



(6) D. Schuster, et al. *Trust and Trustworthy Computing* 2014

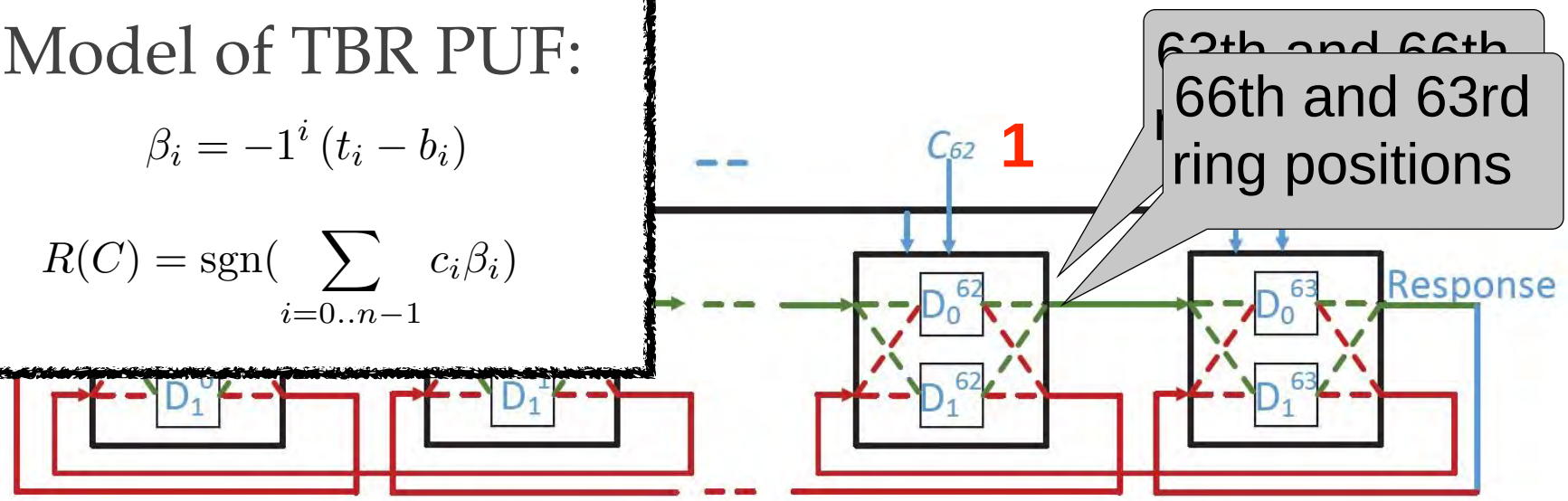
Twisted BR PUF

- TBR-PUF[6] has a more compact design
- All 2n inverting elements used in each ring
- Challenge bit determines whether ring position of each inverting element is even or odd
- Additive model still applies and is simpler than regular BR PUF

Model of TBR PUF:

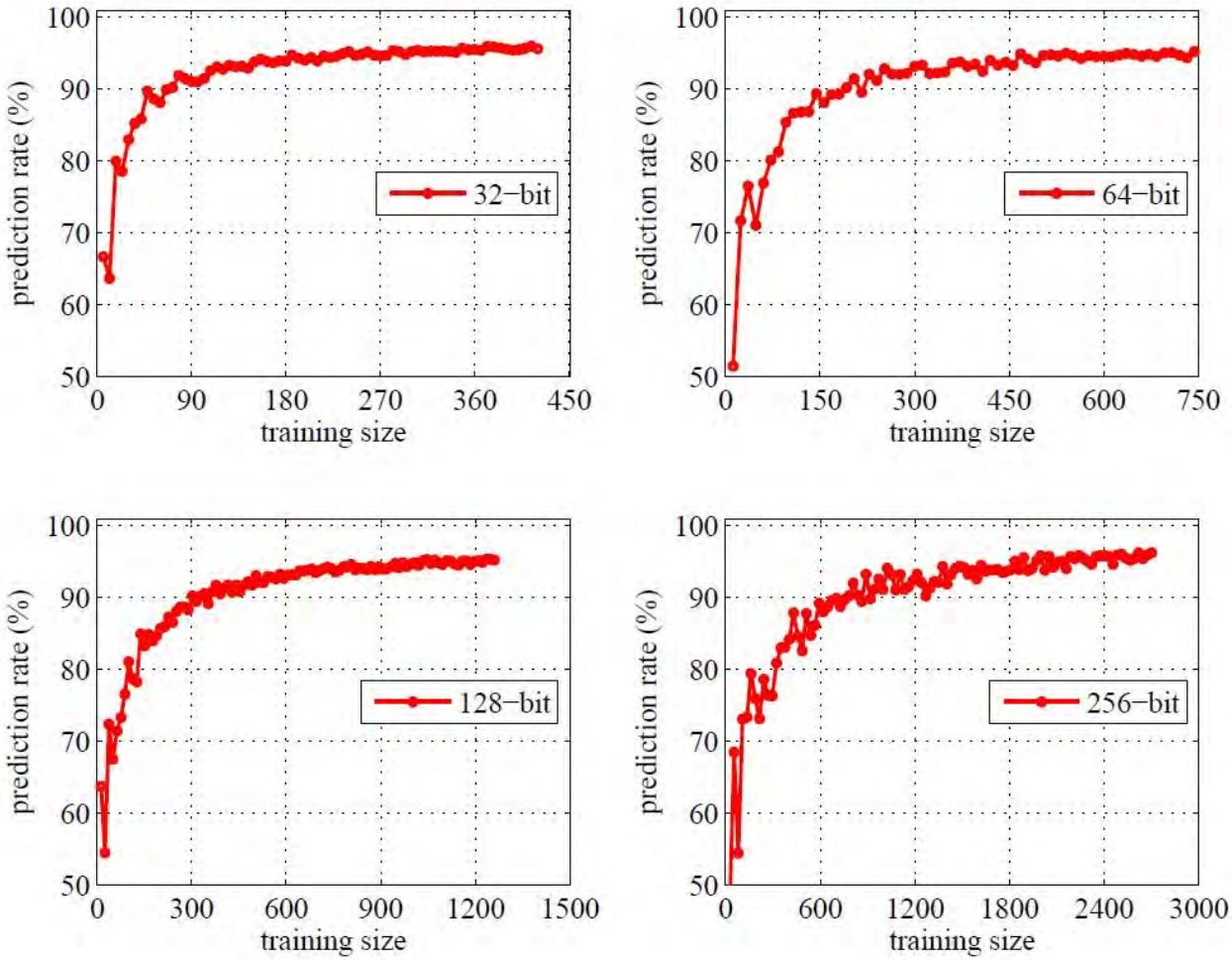
$$\beta_i = -1^i (t_i - b_i)$$

$$R(C) = \mathrm{sgn}(\sum_{i=0..n-1} c_i \beta_i)$$

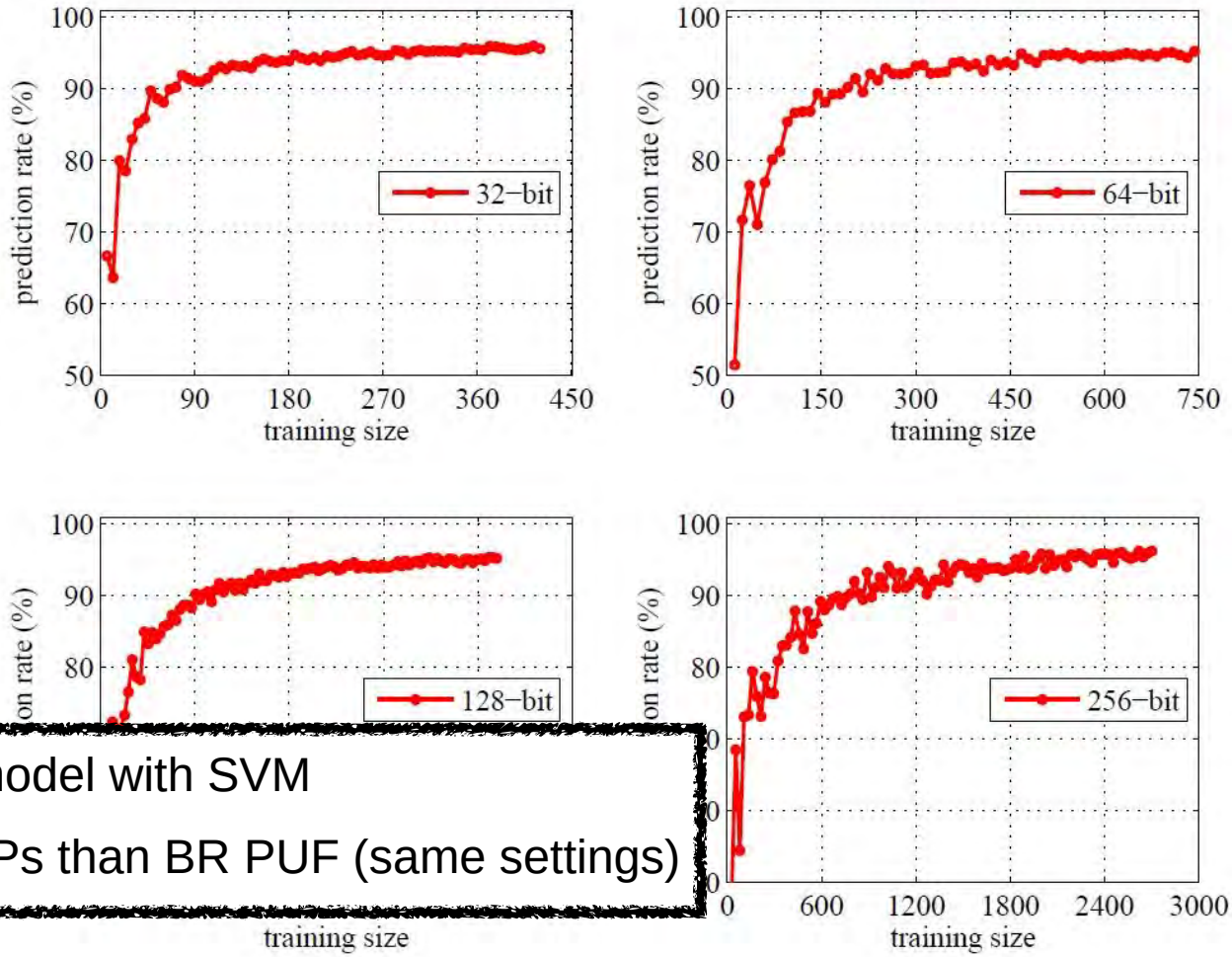


(6) D. Schuster, et al. *Trust and Trustworthy Computing* 2014

TBR PUF is Not Secure



TBR PUF is Not Secure



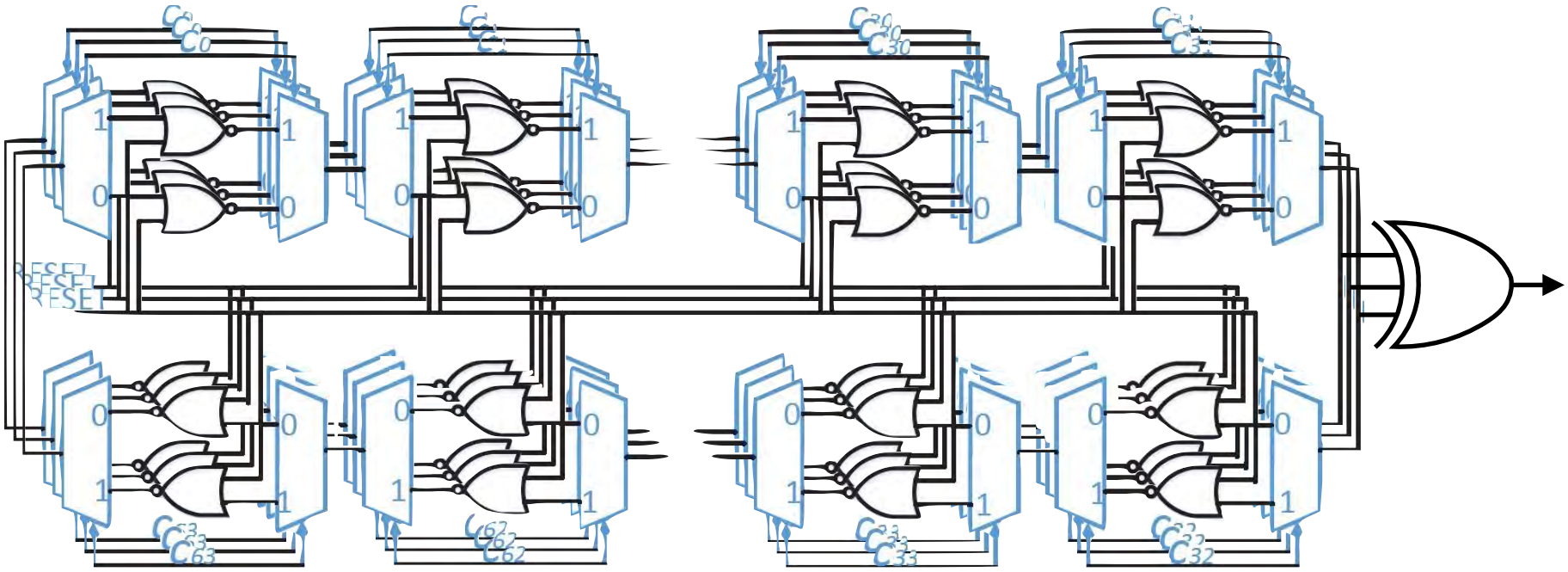
- Trivial to model with SVM
- Fewer CRPs than BR PUF (same settings)

Outline

- Background
  - PUFs
  - Modeling attacks on PUFs
  - Bistable Ring PUF
- Security Evaluation of BR PUFs
  - Modeling the BR PUF
  - Results against BR PUF and variants
- Security Enhancement of BR PUFs
  - XORing BR PUFs to enhance the security
  - Impact on other PUF parameters
- Conclusion and future work

XOR BR PUFs to Enhance Security

- XOR responses to harden against SVM modeling attacks
- Prevent direct observation of CRP relation of single PUFs
- Standard technique in many PUF protocols



Security of XOR BR PUFs

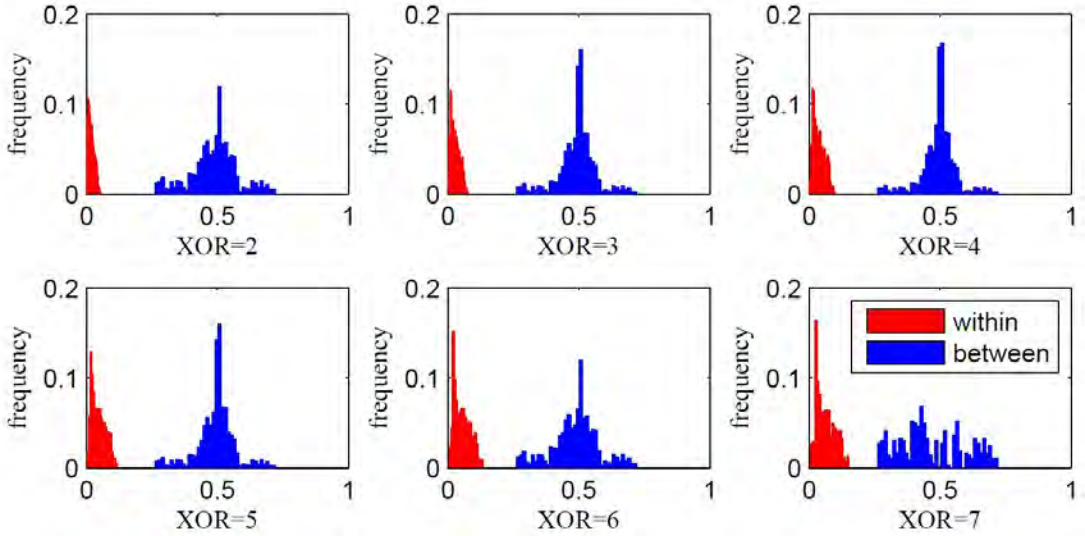
- Resists SVM modeling attacks when >4 XORs used
  - Similar to findings with Arbiter PUFs(1)
- Polynomial kernel; polynomial order set equal the number of XORs
- Stronger machine learning attacks may succeed

No. of XORs	Bit Length	CRPs ($\times 10^3$)	Predict. Rate	Training* Time
2	32	0.8	95%	3 sec
	64	4	95%	10 sec
	128	18	95%	6 mins
	256	—	50.8%	—
3	32	1.2	95%	5 sec
	64	7.2	95%	24 sec
	128	—	50.1%	—
	256	—	50.1%	—
4	32	—	50.1%	—
	64	—	50.3%	—
	128	—	49.8%	—
	256	—	50.1%	—

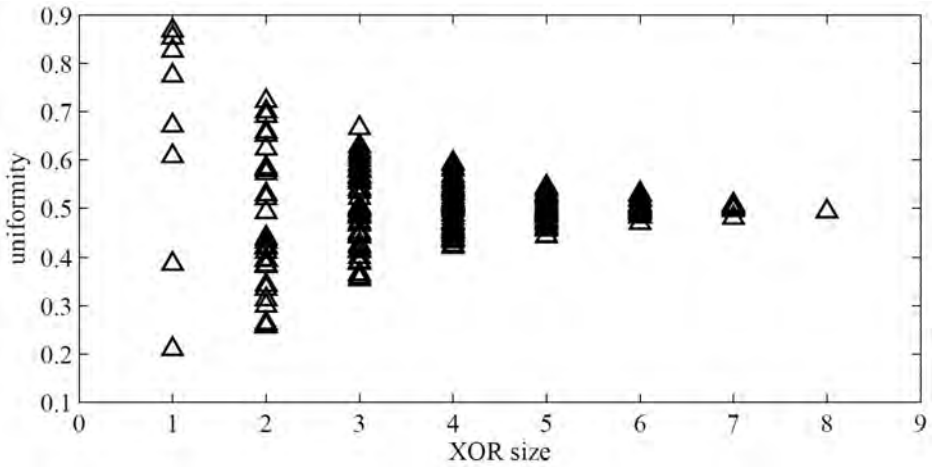
(1) U. Rührmair, et al, CCS, 2010.

Impact of XOR on Uniqueness and Uniformity

- XOR increases within-class Hamming Distance
- Within-class and between-class HD remain separable

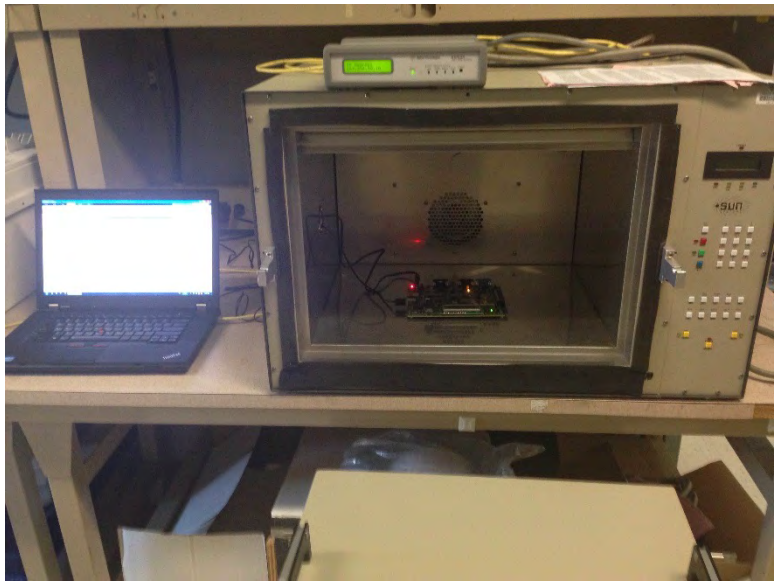


- Single PUFs have poor uniformity
- Uniformity improves with XOR

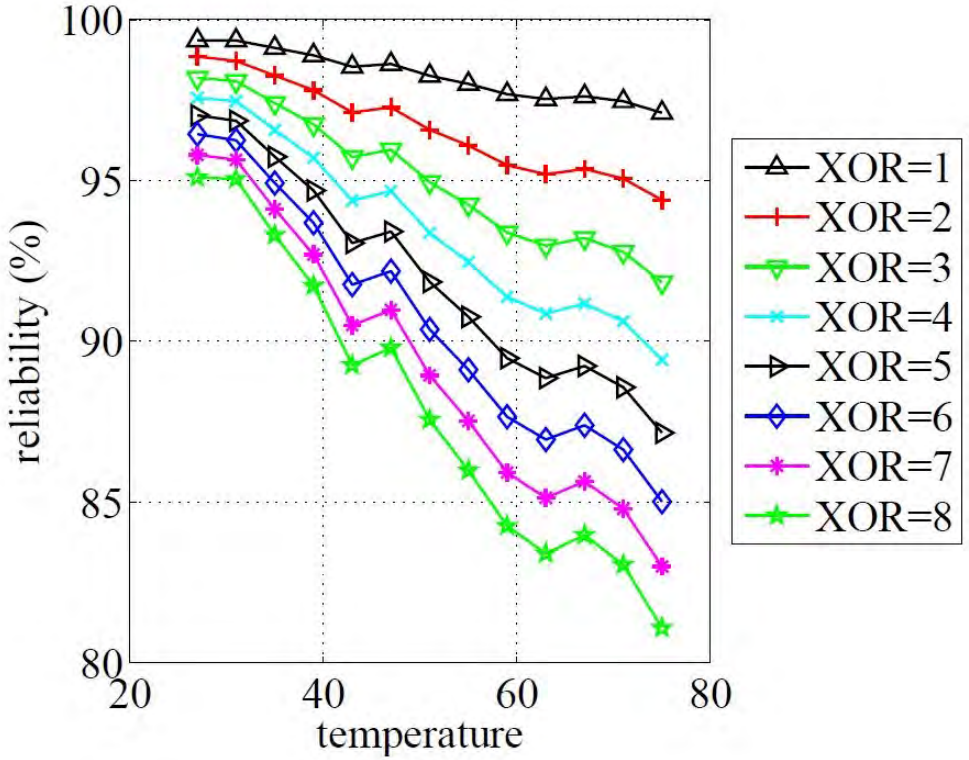


Impact of XOR on Reliability

- XOR degrades reliability
- Any single PUF response flip will change response parity



Sun Electronics EC12 Environmental Chamber



Conclusion and Future Work

- BR PUF and TBR PUF are vulnerable to machine learning modeling attacks
  - 95% accurate prediction surpasses capabilities of ANN-based attacks(1)
  - Reasonable runtime and fewer than 10k CRPs
- XORing four or more BR PUFs produces a behavior that is beyond the modeling capability of the applied SVM attacks
  - XOR function improves uniformity but degrades reliability
- Future work will explore the effectiveness of other modeling attacks including evolutionary strategies and logistic regression

Thank you for your attention

(1) Schuster et al. TRUST 2014