CS771 Assignment I [Neural Ninjas 4]

Aakarsh Mishra
Department of Aerospace Engineering
210008
aakarsh21@iitk.ac.in

Anubhav Vashishtha
Department of Chemical Engineering
210164
anubhavv21@iitk.ac.in

Gandhi Khush Chandreshkumar
Department of Mechanical Engineering
210377
khushcg21@iitk.ac.in

Jatin Rastogi
Department of Chemical Engineering
210466
jatinr21@iitk.ac.in

Mihir
Department of Economics
210607
mihir21@iitk.ac.in

Utkarsh Kumar
Department of Mechanical Engineering
211130
utkarshk21@iitk.ac.in

Abstract

The Companion ArbiteR Physically Unclonable Function (**CAR-PUF**) is an innovative cryptographic primitive extending an Arbiter PUF's complexity. Consisting of a chain of switch PUFs, in an Arbiter PUF, each switch PUF introduces delays that are difficult to replicate. Selectively swapping signals based on challenger bits generates a unique response to challenges. The CAR-PUF, whereas **compares the time delay response** of the working Arbiter PUF with that of a reference PUF and produces a binary response based on a **secret threshold value**. This document presents a detailed analysis of the mathematical formulation of CAR-PUF, showing that it can be broken into **single linear model**.

QUESTION 1

Arbiter PUF

An arbiter PUF is a chain of K switch PUFs, each of which either swaps the signals or keeps them intact, depending on what the challenger bit fed into that switch PUF. If the challenger bit is 0, the switch simply lets the signal pass through, but if the select bit is 1, the switch swaps the two signal paths. The switch PUF each has delays that are hard to replicate but consistent.

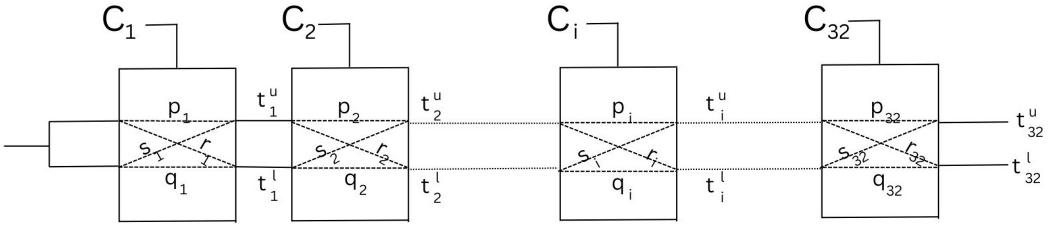


Figure 1: 32-bit mux arbiter PUF

C_i is the i-th challenger bit, where $C_i \in \{0, 1\}$

p_i = delay introduced by the i-th mux when the upper signal passes from above
q_i = delay introduced by the i-th mux when the lower signal passes from below
r_i = delay introduced by the i-th mux when the upper signal passes from below
s_i = delay introduced by the i-th mux when the lower signal passes from above

The time taken by the upper and lower signal to pass i-th mux depends only on the time taken to pass the previous mux, the delay introduced in the i-th mux and the challenger bit fed in the i-th mux as:

$$t_i^u = (1 - C_i)(t_{i-1}^u + p_i) + C_i(t_i^l + s_i)$$
$$t_i^l = (1 - C_i)(t_{i-1}^u + q_i) + C_i(t_i^l + r_i)$$

where
t_i^u = time at which the upper signal leaves the i-th mux
t_i^l = time at which the lower signal leaves the i-th mux

Let Δ_i denote the lags in the signals at i-th mux

$$\Delta_i = t_i^u - t_i^l$$
$$\Delta_i = (1 - C_i)(t_{i-1}^u + p_i - t_{i-1}^l - q_i) + C_i(t_{i-1}^l + s_i - t_{i-1}^u - r_i)$$

For $i = 1$, we have,

The equation

$$\Delta_1 = (1 - C_1) \cdot (\Delta_0 + p_1 - q_1) + C_1 \cdot (-\Delta_0 + s_1 - r_1),$$

which can be rearranged as:

$$\Delta_1 = (1 - 2C_1) \cdot \Delta_0 + \frac{(q_1 - p_1 + s_1 - r_1)}{2} + \frac{(p_1 - q_1 - r_1 + s_1)}{2},$$

therefore

$$\Delta_1 = \Delta_0 \cdot d_1 + \alpha_1 \cdot d_1 + \beta_1.$$

Hence

$$\Delta_i = \Delta_{i-1} \cdot d_i + \alpha_i \cdot d_i + \beta_i$$

where
$d_i = (1 - 2 * C_i)$,where $d_i \in \{-1, 1\}$
$\alpha_i = (p_i - q_i + r_i - s_i)/2$
$\beta_i = (p_i - q_i - r_i + s_i)/2$

Assuming $\Delta_0 = 0$, The difference in the timings of the upper and the lower signal after all the combinations of mux are expressed as:
$\Delta_1 = \alpha_1 * d_1 + \beta_1$
$\Delta_2 = (\alpha_1 * d_1 + \beta_1) * d_2 + \alpha_2 * d_2 + \beta_2$
$\Delta_2 = \alpha_1 * d_1 * d_2 + (\beta_1 + \alpha_2) * d_2 + \beta_2$
$\Delta_3 = \alpha_1 * d_1 * d_2 * d_3 + (\beta_1 + \alpha_2) * d_3 * d_2 + (\beta_2 + \alpha_3) * d_3 + \beta_3$
$\Delta_3 = b_1 * c_1 + b_2 * c_2 + b_3 * c_3 + \beta_3$
where
$c_1 = d_1 * d_2 * d_3$
$c_2 = d_2 * d_3$
$c_3 = d_3$
$b_1 = \alpha_1$
$b_i = \alpha_i + \beta_{i-1}$

In our case of 32-switch PUF, the difference in the time of reaching of both the signals at the end of 32 switches is given by:

$$\Delta_{32} = b_1 * c_1 + b_2 * c_2 + b_3 * c_3 + \ldots + b_{31} * c_{31} + b_{32} * c_{32} + \beta_{32} = \mathbf{b}^T \cdot \mathbf{c} + a$$
$$\Delta_{32} = \mathbf{b}^T \cdot \mathbf{c} + a$$

where
$b_1 = \alpha_1$ fixed for a given PUF
$c_i = d_i * d_{i+1} * .... * d_{32}$ where $c_i \in \{-1, 1\}$
$b_i = \alpha_i + \beta_{i-1}$ for i = 2,3,....,31,32 fixed for a given PUF

Companion ArbiteR PUF (CAR-PUF)

Companion ArbiteR PUF (CAR-PUF for short). A CAR-PUF uses 2 arbiter PUFs – a working PUF, a reference PUF, and a secret threshold value $\tau > 0$. Given a challenge, it is fed into both the working PUF and reference PUF, and the timings for the upper and lower signals for both PUFs are measured. Let Δ_w, Δ_r be the difference in timings experienced for the working PUF and reference PUF, respectively, on the same challenge.

The response to this challenge is 0 if $|\Delta_w - \Delta_r| \leq \tau$ and the response is 1 if $|\Delta_w - \Delta_r| > \tau$, where $\tau > 0$ is the secret threshold value.

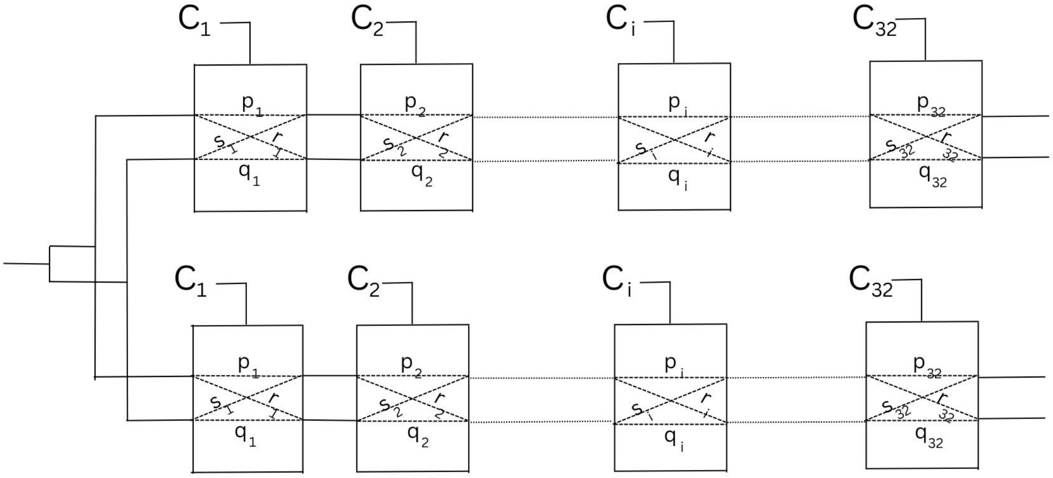


Figure 2: 32-bit CAR-PUF

For Working PUF
$\Delta_w = \mathbf{u}^T \cdot \mathbf{c} + p$

For Reference PUF
$\Delta_r = \mathbf{v}^T \cdot \mathbf{c} + q$

$\Delta = |\Delta_w - \Delta_r|$

Here, we will consider both cases simultaneously,

$\Delta < \tau \implies r = 1$
$|\Delta_w - \Delta_r| < \tau \implies r = 1$
$(\Delta_w - \Delta_r)^2 < \tau^2 \implies r = 1$
$(\Delta_w - \Delta_r)^2 - \tau^2 < 0 \implies r = 1$

$\Delta > \tau \implies r = 0$
$|\Delta_w - \Delta_r| > \tau \implies r = 0$
$(\Delta_w - \Delta_r)^2 > \tau^2 \implies r = 0$
$(\Delta_w - \Delta_r)^2 - \tau^2 > 0 \implies r = 0$

So, we need to figure out the sign of $(\Delta_w - \Delta_r)^2 - \tau^2$ such that the response of CAR-PUF can be expressed as

$$\frac{1 + \text{sign}((\Delta_w - \Delta_r)^2 - \tau^2)}{2} = r$$

Now
$(\Delta_w - \Delta_r) = (\mathbf{u}^T - \mathbf{v}^T) \cdot \mathbf{c} + (p - q)$
$(\Delta_w - \Delta_r) = \mathbf{q}^T \cdot \mathbf{c} + k$

$$\begin{aligned}(\Delta_w - \Delta_r)^2 &= (\mathbf{q}^T \cdot \mathbf{c} + k)^2 \\ &= (\mathbf{q}^T \cdot \mathbf{c})^2 + 2r(\mathbf{q}^T \cdot \mathbf{c}) + r^2 \\ &= (\mathbf{q}^T \cdot \mathbf{c})^2 + 2r\mathbf{q}^T \cdot \mathbf{c} + k^2 \\ &= (\mathbf{q}^T \cdot \mathbf{c})^2 + 2r(q_1c_1 + q_2c_2 + \ldots + q_{32}c_{32}) + k^2 \\ &= q_1^2c_1^2 + q_2^2c_2^2 + \ldots + q_{32}^2c_{32}^2 \\ &+ q_1c_1(q_2c_2 + q_3c_3 + \ldots + q_{32}c_{32}) \\ &+ q_2c_2(q_1c_1 + q_3c_3 + \ldots + q_{32}c_{32}) + \ldots \\ &+ q_{32}c_{32}(q_1c_1 + q_2c_2 + \ldots + q_{31}c_{31}) \\ &+ 2r(q_1c_1 + q_2c_2 + \ldots + q_{32}c_{32}) + k^2\end{aligned}$$

$$(\Delta_w - \Delta_r)^2 = \sum_{i=1}^{32}(q_ic_i)^2 + 2\sum_{j=i+1}^{32}\sum_{i=1}^{32}(q_ic_i)(q_jc_j) + 2r\sum_{i=1}^{32}(q_ic_i) + k^2$$

So

$$(\Delta_w - \Delta_r)^2 - \tau^2 = \sum_{i=1}^{32}(q_ic_i)^2 + 2\sum_{j=i+1}^{32}\sum_{i=1}^{32}(q_ic_i)(q_jc_j) + 2r\sum_{i=1}^{32}(q_ic_i) + k^2 - \tau^2$$

Now

$$\sum_{i=1}^{32}(q_ic_i)^2 \text{ is constant for a given PUF as } c_i^2 = 1 \text{ for all } i \in \{1,2,3....,31,32\}$$

So combining $\sum_{i=1}^{32}(q_ic_i)^2$,k^2 and τ^2 terms we get a constant value. Let this constant value be b. Hence,

$$(\Delta_w - \Delta_r)^2 - \tau^2 = 2\sum_{j=i+1}^{32}\sum_{i=1}^{32}(q_ic_i)(q_jc_j) + 2r\sum_{i=1}^{32}(q_ic_i) + b$$

From the above equation, we can infer that the feature map maps $\{0,1\}^{32}$ to $\mathbb{R}^D$, where $D = 32 + \binom{32}{2} = 528$.

Total Number of terms are 528 + 1 = 529 Terms.
Now we can represent the response of CAR-PUF as

$$\frac{1 + \text{sign}(\mathbf{W}^T\phi(\mathbf{c}) + b)}{2} = r$$

where
$\phi(\mathbf{c})$ is the feature map, W is the linear model, and b is the bias term

$$\phi(\mathbf{c}) = \begin{bmatrix} c_1 & c_2 & \ldots & c_{32} & c_1c_2 & c_1c_3 & \ldots & c_1c_{32} & c_2c_3 & c_2c_4 & \ldots & c_2c_{32}\ldots\ldots & c_{31}c_{32}\end{bmatrix}$$

$$\mathbf{W} = 2\begin{bmatrix} kq_1 & kq_2 & \ldots & kq_{32} & q_1q_2 & q_1q_3 & \ldots & q_1q_{32} & q_2q_3 & q_2q_4 & \ldots & q_2q_{32}\ldots\ldots & q_{31}q_{32}\end{bmatrix}$$

where
$k = p - q$
$q_i = u_i - v_i$

$$b = \textstyle\sum_{i=1}^{32}((u_i - v_i)c_i)^2 + (p - q)^2 - \tau^2$$

QUESTION 3

Loss hyperparameter in LinearSVC & corresponding training time and test accuracy

The data is tabulated below:

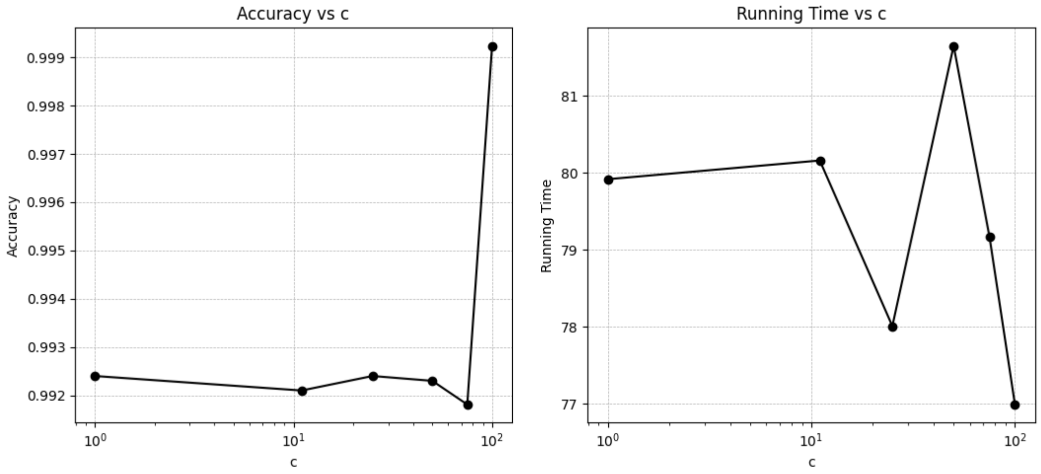
Loss Hyperparameter	**Train Time(in s)**	**Test Accuracy (in %)**
Hinge	78.411	0.9922
Squared Hinge	87.0765	0.992801

Table 1: Tabulated for C = 11, tolerance = 0.001, maximum iteration to be 10000

Training Time and Test Accuracy Dependence on *C* hyperparameter

Linear SVC

This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $tol = 10^{-3}$, $penalty =' l2'$, and maximum iterations to be 10000.



(a) Test Accuracy vs. C

(b) Train Time vs. C

Figure 3: Comparison of Test Accuracy and Train Time vs. C

Logistic Regression

This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $tol = 0.001$, $penalty =' l2'$, and maximum iterations to be 10000.

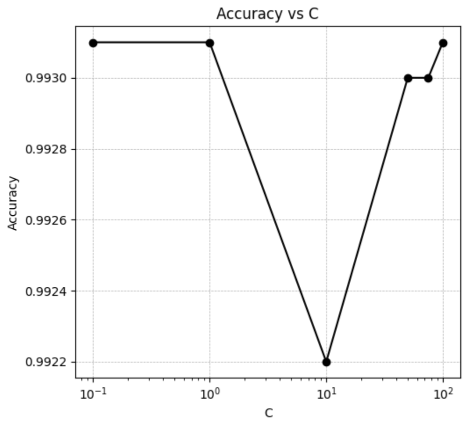


Figure 4: Test Accuracy vs. C

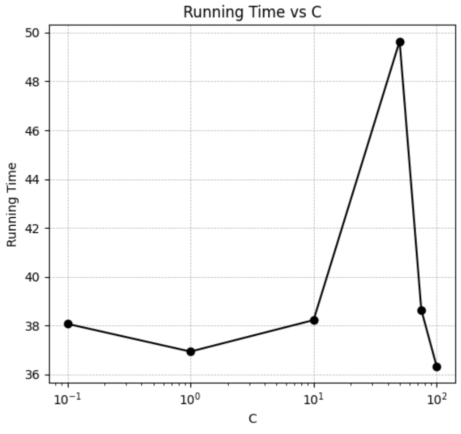


Figure 5: Train Time vs. C

Figure 6: Comparison of Test Accuracy and Train Time vs. C

Training Time and Test Accuracy Dependence on *tol* hyperparameter

Linear SVC
This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $penalty =' l2'$, $C = 11$ and maximum iterations to be 10000.

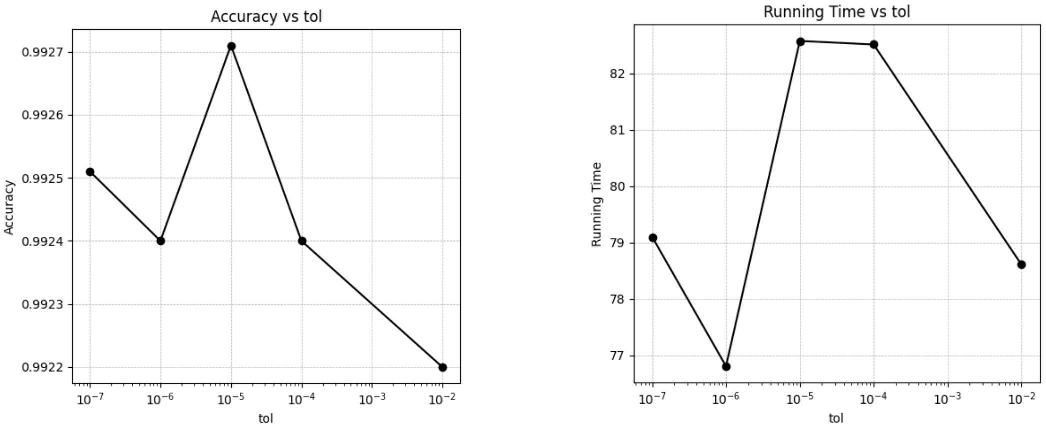


Figure 7: Test Accuracy vs. tol

Figure 8: Train Time vs. tol

Figure 9: Comparison of Test Accuracy and Train Time vs. tol

Logistic Regression
This data is set by fixing the following hyperparameters:
$loss =' hinge'$, $penalty =' l2'$, $C = 100$ and maximum iterations to be 10000.

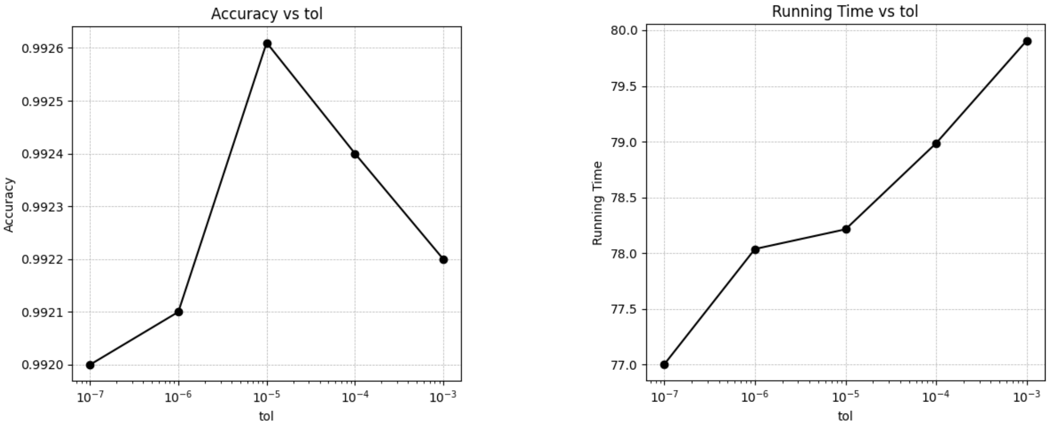


Figure 10: Test Accuracy vs. tol

Figure 11: Train Time vs. tol

Figure 12: Comparison of Test Accuracy and Train Time vs. tol