CS771 Assignment-1

Group No. 71 - CodeCrafters

Madhur Bansal (210572)
Chirag Garg (210288)
Shashank Singh (210964)
Kumar Kanishk Singh (210544)
Vijay Soren (211163)
Krishna Paswan (210531)

1 Mathematical Derivation for CAR-PUF

In this section, we provide a detailed derivation to show that a CAR-PUF can be broken by a single linear model. We also provide a D (= 528) dimensional map $\phi : \{0,1\}^{32} \to \mathbb{R}^D$, mapping a 32-bit challenge vector **c** to D-dimensional feature vector.

1.1 Notations

We have used the following notations in the following sections:

- $\Delta = t^u - t^l$: Time difference between upper and lower signals.
- **c**: Challenge vector
- r: Response to a challenge
- τ: Secret threshold value
- Δ_w, Δ_r: Time difference experienced by the *working* PUF and the *reference* PUF respectively.

1.2 Derivation

We are given that for a challenge vector **c**, the response (r) is 0, if $|\Delta_w - \Delta_r| \leq \tau$ and is 1 if $|\Delta_w - \Delta_r| > \tau$, where $\tau > 0$.

As discussed in class notes, for a single arbiter-PUF, Δ can be represented as following:

$$\Delta = g_0 \cdot x_0 \; + g_1 \cdot x_1 \; + \cdots + g_{31} \cdot x_{31} \; + b_{31} = \mathbf{g}^T \mathbf{x} + b$$

where

$$x_i = d_i \cdot d_{i+1} \cdot \cdots \cdot d_{31}, \; x \in \{-1, 1\}$$
$$d_i = (1 - 2c_i), \; d \in \{-1, 1\}$$

and **g**, b are unknown parameters that depend on the arbiter-PUF.

Therefore,

$$\Delta_w = g_0^{(w)} \cdot x_0 \; + g_1^{(w)} \cdot x_1 \; + \cdots + g_{31}^{(w)} \cdot x_{31} \; + b_{31}^{(w)} = \mathbf{g}_{(w)}^T \mathbf{x} + b^{(w)}$$

$$\Delta_r = g_0^{(r)} \cdot x_0 \ + g_1^{(r)} \cdot x_1 \ + \cdots + g_{31}^{(r)} \cdot x_{31} \ + b_{31}^{(r)} = \mathbf{g}_{(r)}^T \mathbf{x} + b^{(r)}$$

$$\begin{aligned} \Delta_w - \Delta_r &= (g_0^{(w)} - g_0^{(r)}) \cdot x_0 \ + (g_1^{(w)} - g_1^{(r)}) \cdot x_1 \ + \cdots + (g_{31}^{(w)} - g_{31}^{(r)}) \cdot x_{31} \ + (b_{31}^{(w)} - b_{31}^{(r)}) \\ &= g_0' \cdot x_0 \ + g_1' \cdot x_1 \ + \cdots + g_{31}' \cdot x_{31} \ + b_{31}' \\ &= (\mathbf{g'})^T \mathbf{x} + b' = \Delta_g \end{aligned} \tag{1}$$

where $g_i' = g_i^{(w)} - g_i^{(r)}$ and $b' = b^{(w)} - b^{(r)}$

Given that response (r):

$$\begin{aligned} r &= \begin{cases} 0 & \text{if } |\Delta_w - \Delta_r| \leq \tau \\ 1 & \text{if } |\Delta_w - \Delta_r| > \tau \end{cases} \\ &= \begin{cases} 0 & \text{if } |\Delta_g| \leq \tau \\ 1 & \text{if } |\Delta_g| > \tau \end{cases} \\ &= \begin{cases} 0 & \text{if } \Delta_g \leq \tau \text{ and } \Delta_g \geq -\tau \\ 1 & \text{if } \Delta_g > \tau \text{ or } \Delta_g < -\tau \end{cases} \end{aligned}$$

The condition $\Delta_g \leq \tau, \Delta_g \geq -\tau \implies \Delta_g - \tau \leq 0$ and $\Delta_g + \tau \geq 0$

Therefore r = 0, iff both the above conditions are simultaneously satisfied

Since $\tau > 0$, $\Delta_g - \tau \leq 0$ and $\Delta_g + \tau \geq 0 \iff (\Delta_g - \tau)(\Delta_g + \tau) \leq 0$

We can conclude that,

$$r = \begin{cases} 0 & \text{if } \Delta_g^2 - \tau^2 \leq 0 \\ 1 & \text{if } \Delta_g^2 - \tau^2 > 0 \end{cases} \tag{2}$$

Now, using equation (1),

$$\begin{aligned} \Delta_g &= g_0' \cdot x_0 \ + g_1' \cdot x_1 \ + \cdots + g_{31}' \cdot x_{31} \ + b_{31}' \\ &= \Sigma_{i=0}^{31} \, g_i' \cdot x_i + b_{31}' \end{aligned}$$

$$\begin{aligned} \Delta_g^2 &= (\Sigma_{i=0}^{31} \, g_i' \cdot x_i + b_{31}') \cdot (\Sigma_{i=0}^{31} \, g_i' \cdot x_i + b_{31}') \\ &= \Sigma_{i=0}^{31} \Sigma_{j=0}^{31} \, g_i' g_j' \cdot x_i \cdot x_j + 2b_{31}' \cdot \Sigma_{i=0}^{31} \, g_i' \cdot x_i + (b_{31}')^2 + \Sigma_{i=0}^{31} \, (g_i')^2 \cdot x_i^2 \\ &= \Sigma_{j=i+1}^{31} \Sigma_{i=0}^{31} \, h_{ij} \cdot x_i \cdot x_j + \cdot \Sigma_{i=0}^{31} \, k_i \cdot x_i + b'' \\ &= \Sigma_{j=i+1}^{31} \Sigma_{i=0}^{31} \, h_{ij} \cdot z_{ij} + \cdot \Sigma_{i=0}^{31} \, k_i \cdot x_i + b'' \end{aligned} \tag{3}$$

where $z_{ij} = x_i \cdot x_j \ (i \neq j), \ h_{ij} = 2g_i' g_j', \ k_i = 2b_{31}' \cdot g_i', \ b'' = (b_{31}')^2 + \Sigma_{i=0}^{31} \, (g_i')^2 \cdot x_i^2$

We have merged x_i^2 terms in the constant as $x_i \in \{-1, 1\} \implies x_i^2 = 1 \, \forall \, i \in \{1, 2, ..., 31\}$

Using equation (2) and (3),

$$\begin{aligned} \Delta_g^2 - \tau^2 &= \Sigma_{j=i+1}^{31} \Sigma_{i=0}^{31} \, h_{ij} \cdot z_{ij} + \cdot \Sigma_{i=0}^{31} \, k_i \cdot x_i + b'' - \tau^2 \\ &= \Sigma_{j=i+1}^{31} \Sigma_{i=0}^{31} \, h_{ij} \cdot z_{ij} + \cdot \Sigma_{i=0}^{31} \, k_i \cdot x_i + b''' \end{aligned} \tag{4}$$

Equation (4) is the required linear equation. From it we can identify, the map ϕ, **W** and the bias.

$$\phi : \{0, 1\}^{32} \to \mathbb{R}^D$$

$$\phi((c_0, .., c_{31})') = (x_0, ..., x_{31}, z_{0,1}, z_{0,2}, ..., z_{30,31})' \tag{5}$$

where

$$(c_0, .., c_{31})' = \mathbf{c} \text{ is the challenge vector}$$
$$x_i = d_i \cdot d_{i+1} \cdot \dots \cdot d_{31},\ x \in \{-1, 1\}$$
$$d_i = (1 - 2c_i),\ d \in \{-1, 1\}$$
$$z_{ij} = x_i \cdot x_j, (i, j) \in \{(0, 1), (0, 2), ..., (30, 31)\}$$

The total dimension *D* of the map $\phi(.) = 32 + \frac{32.(32-1)}{2} = 32 + 496 = 528$

$$\mathbf{W} : D\text{-dimensional linear model}$$
$$\mathbf{W} = (h_{0,1}, h_{0,2}, \cdots, h_{30,31}, k_0, k_1, \cdots, k_{31})' \tag{6}$$

where h_{ij} and k_i are defined in equation (3).

$$\text{Bias } = b'''$$

2 Code

3 Experimental Outcomes

3.1 a) Effect of Loss function: Hinge Squared Loss vs Hinge Loss

3.1.1 Linear SVC

C: 1
Loss: Hinge Squared Loss
Tol: $1e^{-4}$
Penalty: *l*2

Table 1: Effect of Loss Function

Model Description	Training Time	Mapping Time	Accuracy
Hinge Squared Loss, iter = 20,000	53.612s	0.139s	0.9919
Hinge Loss, iter = 20,000	36.157s	0.119s	0.9897
Hinge Loss, iter = 1,60,000	172.047s	0.129s	0.9895

3.1.2 Observations:

- The model trained with Hinge Squared Loss achieves the highest accuracy which is better than the model with Hinge Loss
- The model with hinge loss is not converging even after increasing the iterations.

3.2 b) Effect of C:

3.2.1 Linear SVC

Loss: Hinge Squared Loss
Tol: $1e^{-4}$
Penalty: $l2$

Table 2: Effect of C: Linear SVC

Model Description	Training Time	Mapping Time	Accuracy
C = 0.01, iter = 20,000	5.841s	0.123s	0.9865
C = 0.1, iter = 20,000	16.636s	0.139s	0.9899
C = 1, iter = 20,000	53.612s	0.139s	0.9919
C = 10, iter = 20,000	58.619s	0.127s	0.9931
C = 10, iter = 100,000	196.126s	0.118s	0.993
C = 100, iter = 100,000	53.146s	0.128s	0.9915

3.2.2 Logistic Regression

Tol: $1e^{-4}$
Penalty: $l2$

Table 3: Effect of C: Logistic Regression

Model Description	Training Time	Mapping Time	Accuracy
C = 0.01, iter = 20,000	1.079s	0.139s	0.9635
C = 0.1, iter = 20,000	2.089s	0.199s	0.9871
C = 1, iter = 20,000	1.310s	0.127s	0.9907
C = 10, iter = 20,000	1.729s	0.156s	0.9922
C = 100, iter = 20,000	2.085s	0.133s	0.9931

3.2.3 Observations:

- Increasing the value of C in linear SVC from (0.01 to 10) generally leads to higher accuracy but after C=10 the model is not converging.
- For the logistic regression case, increasing the value of C (from 0.001 to 100) leads to higher accuracy for the model.

3.3 c) Effect of Tolerance:

3.3.1 Linear SVC

Loss: Hinge Squared Loss
C: 1
Penalty: $l2$

Table 4: Effect of tol: Linear SVC

Model Description	Training Time	Mapping Time	Accuracy
tol = $1e^{-6}$, iter = 20,000	78.951s	0.168s	0.9919
tol = $1e^{-4}$, iter = 20,000	53.612s	0.139s	0.9919
tol = $1e^{-2}$, iter = 20,000	36.731s	0.162s	0.9919
tol = $1e^{-1}$, iter = 20,000	26.339s	0.164s	0.9921

3.3.2 Logistic Regression

C: 1
Penalty: $l2$

Table 5: Effect of tol: Logistic Regression

Model Description	Training Time	Mapping Time	Accuracy
tol = $1e^{-6}$, iter = 20,000	1.974s	0.198s	0.9907
tol = $1e^{-4}$, iter = 20,000	1.310s	0.127s	0.9907
tol = $1e^{-2}$, iter = 20,000	1.459s	0.169s	0.9907
tol = $1e^{-1}$, iter = 20,000	1.522s	0.224s	0.9907

3.3.3 Observations:

- The accuracy remains relatively stable across different values of tolerance for both models (linear SVC and Logistic Regression).
- This suggests that the choice of tolerance does not significantly impact the model's predictive performance.
- In the context of training time, the model with logistic regression is better but in the context of accuracy, the linearSVC is better.

3.4 d) Effect of Penalty:

3.4.1 Linear SVC

C: 1
Loss: Hinge squared loss
Tol: $1e^{-4}$

Table 6: Effect of Penalty: Linear SVC

Model Description	Training Time	Mapping Time	Accuracy
penalty = $l2$, iter = 20,000	53.612s	0.139s	0.9919
penalty = $l1$, dual = F, iter = 1000	155.896s	0.119s	0.9909

3.4.2 Logistic Regression

C: 1
Tol: $1e^{-4}$

Table 7: Effect of Penalty: Logistic Regression

Model Description	Training Time	Mapping Time	Accuracy
penalty = $l1$, solver = liblinear, iter = 20,000	196.016s	0.125s	0.9918
penalty = $l2$, solver = liblinear, iter = 20,000	7.569s	0.126s	0.9906

3.4.3 Observations:

- In the context of accuracy both penalties does not drastically affect the model
- The penalty term '$l2$' takes less time to train for both models.
- The choice between '$l1$' and '$l2$'penalties impacts training time significantly