CS771 Assignment 1

Team Arjun
Team members : Abhishek Khandelwal (220040)
Dhruv Gupta (220361)
Pallav Goyal (220747)
Poojal Katiyar (220770)
Pragati Agrawal (220779)
Shahnawaj (220997)

1 Part I - Cracking the CAR PUF

Note: In this derivation, an n-dimensional vector $\mathbf{a}$ *is expressed as* $(a_0, a_1, \ldots, a_{n-1})$ *where* a_i*'s are components of the vector (*$i \in \{0, 1, \ldots, n-1$ *and* $n \in \mathbf{N}$*). This indicates a* $n \times 1$ *matrix and its transpose (denoted as* $\mathbf{a^T}$*) indicates a* $1 \times n$ *matrix.*

Let (**u**, p) and (**v**,q) be the two linear models that can exactly predict the output of the working PUF and the reference PUF respectively.

Let **c** = $(c_0, c_1, c_2, c_3, \ldots, c_{31})$ be the 32-dimensional challenge vector where for each $i \in \{0, 1, 2, \ldots, 31\}, c_i \in \{0, 1\}$.

Let Δ_w and Δ_r denote the difference in timings experienced for the working PUF and reference PUF respectively.

As discussed in class, we have:

$$\Delta_w = \mathbf{u^T x} + p \tag{1}$$

and

$$\Delta_r = \mathbf{v^T x} + q \tag{2}$$

where **x** = $(x_0, x_1, \ldots, x_{31})$ is a 32-dimensional vector with $x_i = d_i.d_{i+1}.d_{i+2}\ldots.d_{31}$ for each $i \in \{0, 1, 2, \ldots, 31\}$ and $d_j = 1 - 2 * c_j \; \forall \; j \in \{0, 1, \ldots, 31\}$.

The response given by the CAR-PUF for a challenge is 0 if $|\Delta_w - \Delta_r| \leq \tau$ and 1 if $|\Delta_w - \Delta_r| > \tau$.

Thus response r of the CAR-PUF can be written as:

$$r = \frac{1 + sign(|\Delta_w - \Delta_r| - \tau)}{2} \tag{3}$$

where $sign(k) = \begin{cases} -1 & \text{if } k \leq 0 \\ 1 & \text{if } k > 0 \end{cases}$

and $\tau > 0$ is the secret threshold value.

Since both $|\Delta_w - \Delta_r|$ and τ are non-negative, we observe that:

$sign(|\Delta_w - \Delta_r| - \tau) = sign(|\Delta_w - \Delta_r|^2 - \tau^2) = sign((\Delta_w - \Delta_r)^2 - \tau^2)$

Thus, we have:

Preprint. Under review.

$$r = \frac{1 + sign((\Delta_w - \Delta_r)^2 - \tau^2)}{2} \tag{4}$$

Let us now simplify $(\Delta_w - \Delta_r)^2$ using equations (1) and (2):

$$(\Delta_w - \Delta_r)^2 = (\mathbf{u^T x} + p - \mathbf{v^T x} - q)^2$$

$$(\Delta_w - \Delta_r)^2 = ((\mathbf{u^T} - \mathbf{v^T})\mathbf{x} + p - q)^2$$

$$(\Delta_w - \Delta_r)^2 = ((\mathbf{u} - \mathbf{v})^\mathbf{T}\mathbf{x} + p - q)^2$$

Denote $\mathbf{u} - \mathbf{v}$ with $\mathbf{s} = (s_0, s_1, \ldots, s_{31})$ and $p - q$ with t.

$$(\Delta_w - \Delta_r)^2 = (\mathbf{s^T x} + t)^2$$

$$(\Delta_w - \Delta_r)^2 = (s_0x_0 + s_1x_1 + \ldots + s_{31}x_{31} + t)^2$$

$$(\Delta_w - \Delta_r)^2 = s_0^2x_0^2 + s_1^2x_1^2 + \ldots + s_{31}^2x_{31}^2 + t^2 + 2(\alpha)$$

where α is the sum of pairwise products of all possible pairs using the terms $\{s_0x_0, s_1x_1, \ldots, s_{31}x_{31}, t\}$. Note that there are $\binom{33}{2} = 528$ distinct possible pairs.

However, we can observe that $x_i^2 = 1 \; \forall \; i \in \{0, 1, \ldots, 31\}$. Thus,

$$(\Delta_w - \Delta_r)^2 = T^2 + 2(\alpha)$$

where $T^2 = t^2 + s_0^2 + s_1^2 + \ldots + s_{31}^2$

We can now write:

$$(\Delta_w - \Delta_r)^2 = \mathbf{W^T X} + T^2$$

where $\mathbf{X} = (x_0x_1, x_0x_2, \ldots, x_0x_{31}, x_0, x_1x_2, x_1x_3, \ldots, x_1x_{31}, x_1, x_2x_3, \ldots, x_{30}x_{31}, x_{30}, x_{31})$
and $\mathbf{W} = (2s_0s_1, 2s_0s_2, \ldots, 2s_0s_{31}, 2s_0t, 2s_1s_2, 2s_1s_3, \ldots, 2s_1s_{31}, 2s_1t, 2s_2s_3, \ldots, 2s_{30}s_{31}, 2s_{30}t, 2s_{31}t)$

Note that the components of $\mathbf{X}$ and $\mathbf{W}$ correspond to those in α. Hence, one can see that the dimension of $\mathbf{X}$ and $\mathbf{W}$ is 528.

Further, we can do the following simplification:

$$(\Delta_w - \Delta_r)^2 - \tau^2 = \mathbf{W^T X} + T^2 - \tau^2$$

Denote $T^2 - \tau^2$ by b.

$$(\Delta_w - \Delta_r)^2 - \tau^2 = \mathbf{W^T X} + b$$

From equation (4), we have:

$$r = \frac{1 + sign(\mathbf{W^T X} + b)}{2} \tag{5}$$

We can clearly see that there exists a linear model that always predicts the correct response, thus breaking the CAR-PUF.

Using the same way in which $\mathbf{X}$ has been obtained from $\mathbf{c}$ in the above derivation, we can define a map $\phi : \{0,1\}^{32} \rightarrow \mathbb{R}^{528}$, that maps a 32-bit 0/1-valued challenge vector $\mathbf{c}$ to a 528-dimensional feature vector $\mathbf{X}$. Note that ϕ does not depend on any PUF-specific constants like $\mathbf{u}$,$\mathbf{v}$,p,q,τ. (We can observe from the above derivation that $\mathbf{X}$ does not depend on such constants, rather, it depends only on $\mathbf{c}$ and universal constants). Now, for any CAR-PUF, there exists a 528-dimensional linear model $\mathbf{W} \in \mathbb{R}^{528}$ and a bias term b in $\mathbb{R}$, such that for all challenge-response pairs ($\mathbf{c}$,r), with $\mathbf{c} \in \{0,1\}^{32}$ and $r \in \{0,1\}$, we have:

$$\frac{1 + sign(\mathbf{W^T}\phi(\mathbf{c}) + b)}{2} = r \tag{6}$$

2 Part III - Report

- a) On keeping the 'loss' hyperparameter as 'squared hinge' for LinearSVC, the training time was found out to be 56.75 and the test accuracy was found out to be 99.19%.

  On keeping the 'loss' hyperparameter as 'hinge' for LinearSVC the training time was found out to be 24.94 and the test accuracy was found out to be 98.948%.

  In the above cases the 'max_iter' hyperparameter was set to be 10000 and other parameters were taken to be default values.

- d) On keeping the 'penalty' hyperparameter as 'l1' for LinearSVC the training time was found out to be 155.10 and the test accuracy was found out to be 99.124%.

  On keeping the 'penalty' hyperparameter as 'l2' for LinearSVC the training time was found out to be 5.01 and the test accuracy was found out to be 99.19%.

  In the above cases 'dual' hyperparameter was kept to be False and the 'max_iter' were set at 1000 and the 'loss' was taken 'squared hinge'.

  On keeping the 'penalty' hyperparameter as 'l1' for LogisticRegression the training time was found out to be 196.72 and the test accuracy was found out to be 99.18%.

  On keeping the 'penalty' hyperparameter as 'l2' for LogisticRegression the training time was found out to be 1.42 and the test accuracy was found out to be 99.07%.

  In the above cases 'dual' hyperparameter was kept to be False and the 'max_iter' were set at 10000 and the 'solver' was taken as 'liblinear'.

  Note: All the above mentioned time are in seconds.