Power Side Channel Attacks on Hardware Wallet to Recover Pins

Supervisor : Prof. Urbi Chatterjee

Contents

1. Introduction 2
2. Background 2
3. Vulnerability in Trezor-One Firmware 3
4. Experimental Setup 4
5. Performing side-channel attack on an emulator for Trezor 4
6. ML Based Side Channel Attack on the Emulator 5
7. Pin Retrieval 5
8. Performing side-channel attack on Trezor one 6
9. Future Work 8

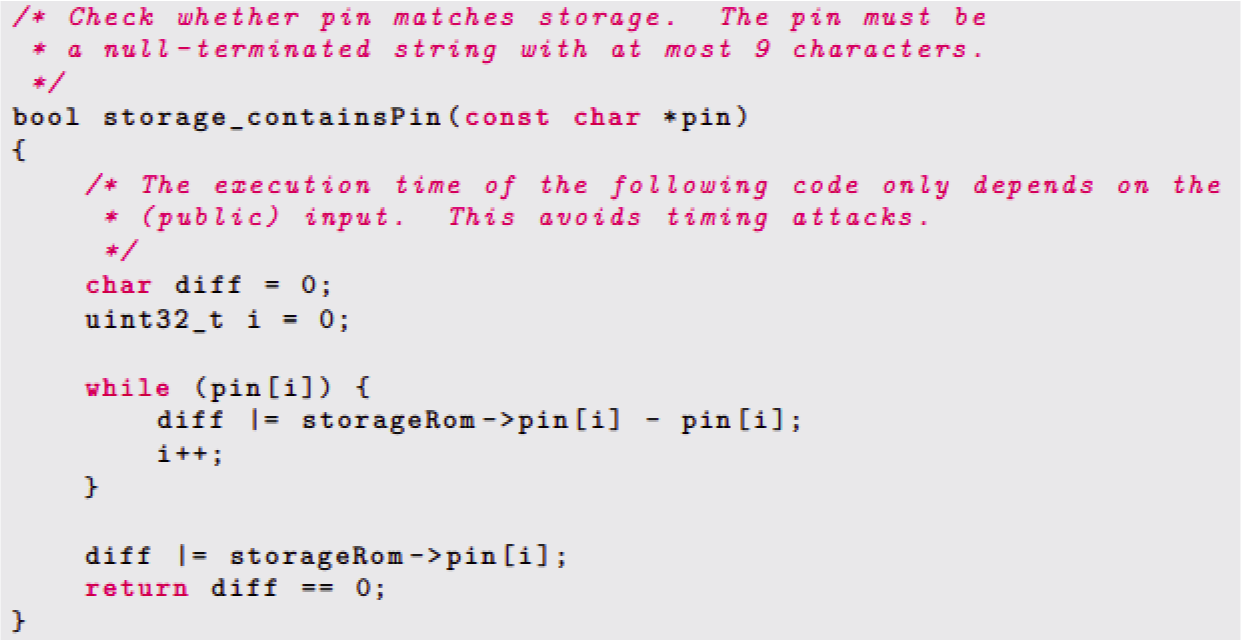
Introduction

Hardware wallets such as Trezor are used to store private keys of users so that they are not exposed to attackers, this makes them secure to store funds. Although keys are not directly exposed, there is a possibility of leakage in power consumption when some operations are performed in the wallet, which might make it vulnerable to side-channel attacks. In this work we have built an emulator for trezor firmware using the Rainbow Library and tried to extract its pin using ML attack on simulated power traces, further we have obtained real traces using oscilloscope, using which a similar attack can be performed.

Background

1. Side Channel Attack: Side-Channel Attack are attacks on devices which depend on the characteristics leaked by those devices (such as time, power, EM waves) during computation, which depend on the data which is manipulated. An attacker can use this leaked information to retrieve sensitive data.
2. Trezor-One: Trezor one is an open source hardware wallet which used a 4-6 digit PIN to safeguard the private keys stored inside it.

Vulnerability in Trezor-One Firmware



```
/* Check whether pin matches storage.  The pin must be
 * a null-terminated string with at most 9 characters.
 */
bool storage_containsPin(const char *pin)
{
    /* The execution time of the following code only depends on the
     * (public) input.  This avoids timing attacks.
     */
    char diff = 0;
    uint32_t i = 0;

    while (pin[i]) {
        diff |= storageRom->pin[i] - pin[i];
        i++;
    }

    diff |= storageRom->pin[i];
    return diff == 0;
}
```

As noted by [1], in the actual pin comparison function of Trezor-one digits are processed one after other and the comparison with pin is done in a deterministic way. So basically, our task is to perform a side-channel attack on each digit of the pin. Further the leakage depends on the value of difference between input and actual pin. This claim is further proved during analysis of traces.

Experimental Setup

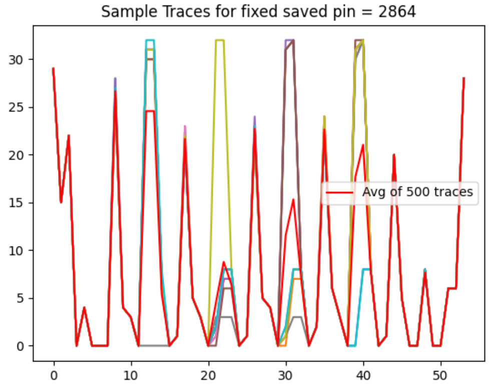
We conducted 2 experiments

1. Performing side-channel attack on an emulator for Trezor
2. Attempting side-channel attack on an actual Trezor-One

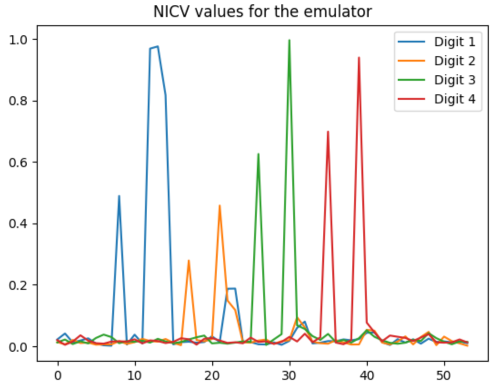
Performing side-channel attack on an emulator for Trezor

- We first built an emulator for Trezor.
- Next, we tried to extract some extract Traces from the emulator.
- Upon analysis of these traces we observe the following NICV values

Sample Traces



NICV values using difference as class metric



ML Based Side Channel Attack on the Emulator

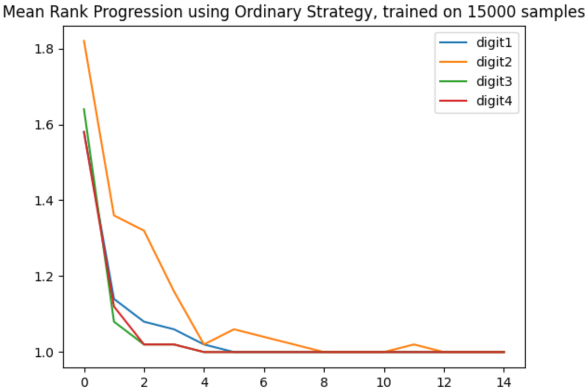
- Trained classifiers for each digit with SVM[2] using linear kernel
- Trained with 1500 traces
- Always correct predictions when the presented digit is the same as the actual stored pin digit.

Pin Retrieval

We use the SVM models that we trained for each digit to retrieve the actual pin, for which we explored three strategies.

1. Random Pin input strategy
2. Chosen Pin input strategy
3. Iterative Pin input strategy

Random Pin Input Strategy



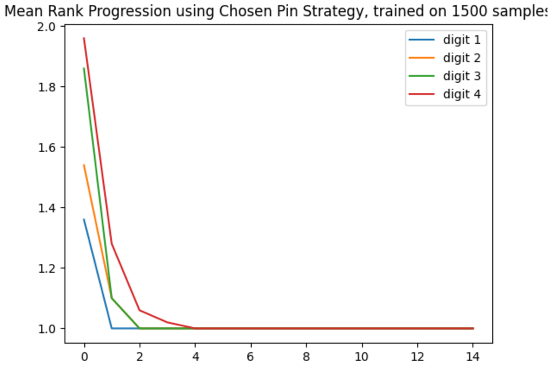
In each trial for Stored Pin:

1. Generate random input pin
2. Get Traces for the actual and input pin
3. Store the predicted probability of each digit
4. Repeat 1,2,3 for 14 rounds

Finally, Find the most probably digit at each place.

As can be seen, we can get the correct pin in 14 tries.

Chosen Pin Strategy



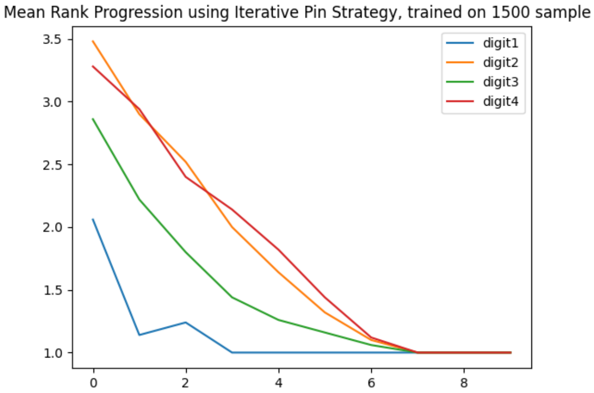
Use First input pin as “5555”

In each trial for Stored Pin:

1. Get Traces for the actual and input pin
2. Store the predicted probability of each digit
3. Use the most probable digits as next input pin.
4. Repeat 1,2,3 till input == stored pin

As can be seen, we can get the actual pin within 5 tries using the chosen pin strategy

Iterative Pin Strategy



Use First input pin as “1111”

In each trial for Stored Pin:

1. Get Traces for the actual and input pin
2. For each index check if the input pin digit is the most probable predicted digit.
   a. If Yes, keep the input pin digit same
   b. Else increment the digit by 1.
3. Repeat 1,2,3 till input == stored pin (for max 9 trials)

As can be seen, we can get the actual pin within 9 tries using the iterative pin strategy

Attempting Side-Channel Attack on Trezor-One

Next, we decided to perform side-channel attack on an actual Trezor-One Hardware wallet

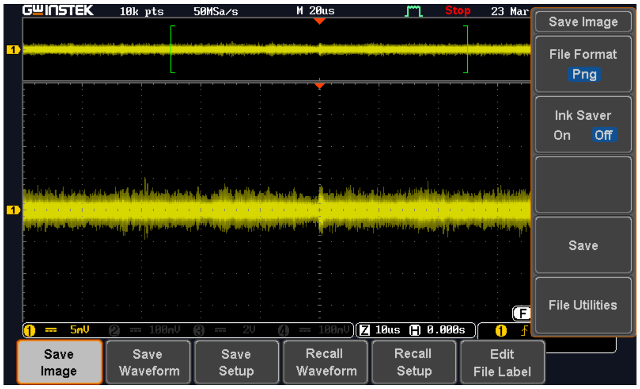
- First we removed the casing from our wallet
- We then connected our wallet to an oscilloscope and took some sample readings.
- We finally created a method to take periodic readings from the oscilloscope, and save them as csv. Using this we can extract multiple traces after passing a command

Setup for Trezor-One

An Oscilloscope. 200MHz Frequency, 1GS/a

Trezor-One Hardware wallet

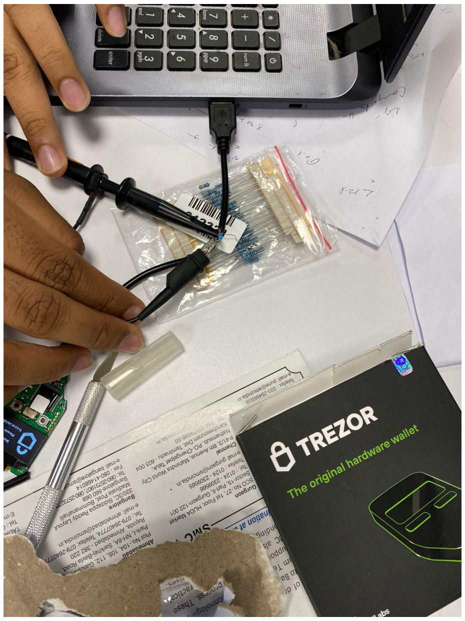
Sample readings from Trezor



File Format Png

Ink Saver On Off

Taking readings



Future Work

- Trigger based trace collection from the oscilloscope
- Automating sending triggers to wallet.
- Performing similar analysis and checking for possible side channel leakage in the latest Trezor Firmware.

References

1. https://eprint.iacr.org/2019/401.pdf
2. https://ieeexplore.ieee.org/document/708428