CS5435: Security and Privacy concepts in the wild

Fall2014

Homework Assignment 2 Write-Up

Jai Chaudhary

jc2855@cornell.edu

Problem 1

VeriChip was a bad idea as an authentication mechanism as:

- Security
  - The emitted secret key that was only a 16 digit and could be guessed by brute force
  - The secret key was static and thus by recording the emitted signal and reverse engineering, VeriChip could be cloned leading to Identity Theft
  - Health Concerns: correlation between implanted chips and cancer
- Privacy
  - Mandated by governments as compulsory could lead to erosion of civil liberties.
  - Since VeriChip data contains health history data and the data is stored in cloud stealing the key could lead to disclosure of patient data

Problem 2.1

Suppose that for some customer X (with a random card different from the example above), you've captured the passcode "y9ahe." Briefly describe an attack in which you use this passcode to try to impersonate X. Give an example of a passcode you might use in this attack. [5 points]

- Since the "Free navigation" constraints are symmetric (up/down, left/right).
- By reversing the characters of passcode you get another valid passcode.
- Attach for Passcode "y9ahe" : "eha9y"

Suppose the character 'e' is not incident on an edge, i.e., isn't in row 0 or 8 or column A or K, and that it is adjacent to a square containing the 'h' of the captured passcode, but not to any of its other characters. Give an estimate of the probability that your attack succeeds and show how you arrived at it. (Assume for simplicity that this is user X's first authentication attempt, and make the simplfying assumption that no character is adjacent to two identical ones.) [10 points]

- Since the reversal trick in previous answer generates a valid passcode everytime. Probability of success is 1.
- Assuming that reverse passcodes are not allowed.
  - Under the constraints given since e is not adjacent to any other characters. The combination "he" will stick together
  - An attack could be to use "9yahe" that is traversing from 9 to y to a instead of y to 9 to a
  - The rough probability estimate of this attacks success is 2/7.
    - It is based on the idea that since "9" has 8 neighborhood squares.
    - One of them is occupied by "a".

- Therefore "y" needs to be in any of the two "9"'s neighborhood squares adjacent to "a". Such that the transition "y" to "a" is valid in the attack
- The probability of "y" being in 2 out of 7 elements is 2/7

Problem 2.2

Does the password-protected variant help protect against your attack in the previous question? Why or why not? [5 points]

- Yes it does protect against the previous attack.
- Since the attack's passcode generated by reversal of intercepted passcode is no longer valid.

Suppose that X's password is a diagonal line (e.g., "move lower right" four times). With what probability can you impersonate X after eavesdropping just once on X's submitted passcode? How did you arrive at this result? (Again, assume that X chooses a sequence that does not terminate in a character incident on the edge of the card. You may assume for simplicity that this is X's first authentication attempt and that you know X's "password" in addition to the captured passcode.) [5 points]

- Since the X's password is a diagonal line. First 4 digits of another valid passcode are same as last four digits of intercepted passcode
- The attacker has to guess the next character in 3 attempts.
- The probability of success is
  - Total_characters = 36
  - Prob(Success in First Attempt) + Prob(Success in Second Attempt) + Prob(Success in Third Attempt)
  - 1/36 + (35/36) X (1/35) + (35/36) X (34/35)X(1/34) =1/12.

Problem 2.3

After eavesdropping once a day for a month (30 days) on challenges and the passcodes with which X responds, what is the probability that you can learn the characters in at least 75% of the squares in X's card? You may code up a simula- tion and use it to provide an (accurate) approximation of the answer. [10 points]

Analytical Solution:
The challenges can be modeled by nested binomial distributions

- Bernoulli : For every grid-index in a single challenge, Probability of visibility = 5/90
- Binomial: for any grid-index summing up the random variable representing visibility we get a binomial distribution with 30 trials. The probability of a grid-index never being made visible in 30 challenges = 0.18 (k = 0, n = 30, p = 5/90)
- Bernoulli: The probability of grid-index never seen in 30 challenges is now the Bernoulli probability

- Binomial: The probability of atleast 68 indices appearing atleast is the expression listed below whose value is 95%.

$$\sum_{n=0}^{22} \frac{90! \times 0.18^{n}\, 0.82^{90-n}}{n!\,(90-n)!} =$$

Numerical/Simulation Solution

Code:

```
import numpy as np

trials = 10000

nonZeroIndicesCount = np.zeros(trials)
for k in xrange(trials):
    sum = np.zeros(90)
    for i in xrange(1,30):
        indexesSet = np.random.randint(90, size=5)
        random_array = np.zeros(90)
        for j in xrange(5):
            random_array[indexesSet[j]] = 1
        sum = np.add(sum, random_array)
    nonZeroIndicesCount[k] = np.shape(np.nonzero(sum))[1]

print np.mean(nonZeroIndicesCount >= 68)
```

The numerical simulation gives probability around 94.5%

Assuming that you've learned exactly 75% of the squares, what is your success probability in attempting to impersonate X? [5 points]

The probability of 5 random indexes to be one of 68 out of 90 = ${}^{68}C_5$ / ${}^{90}C_5$ = 0.23

Problem 2.4

Devise and simulate a full attack in which you eavesdrop once a day for a month (30 days), as above. Give the probability, via simulation, that you can successfully impersonate X. [5 points]

Analytical Solution

Given that the average number of visible characters = 90 *(1-0.18) = 73. The probability of 5 random indexes to be one of 73 out of 90 = ${}^{73}C_5$ / ${}^{90}C_5$ = 0.33

Numerical Solution

Code:

```
import numpy as np
trials = 100000
impersonateCount = np.zeros(6)
for k in xrange(trials):
    sum = np.zeros(90)
    for i in xrange(1,30):
        indexesSet = np.random.randint(90, size=5)
        random_array = np.zeros(90)
        for j in xrange(5):
            random_array[indexesSet[j]] = 1
        sum = np.add(sum, random_array)

    indexesSet = np.random.randint(90, size=5)
    random_array = np.zeros(90)
    for j in xrange(5):
        random_array[indexesSet[j]] = 1
    impersonateCount[np.shape(np.nonzero(np.multiply(random_array, sum)))[1]] += 1

print impersonateCount/trials
```

The probability, via simulation, that you can successfully impersonate X is approximately 33%

- The probability of 5 character match is 0.3, 4 character match is 0.4 and 3 character match is 0.2
- Multiplied by probability of successful guessing, total probability = 0.32

Then devise a different "challenge-response" scheme that yields a lower probability of impersonation of X in the 30-day eavesdropping scenario. Describe your scheme and show using your attack simulation that your scheme is stronger than Otis's in this particular setting. [10 points]

The scheme is at some level of abstraction just adding a salt to the response to make it resistant to eavesdropping.

Scheme

- The users grid has a few set of rules to conveniently transform the challenge indices and respond using the transformed values
- The server sends user a salt-like rule id, that the user has to apply before sending the response

Using just 5 rules. The space of possible combinations has combinatorial explosion from mappings.

Problem 2.5

Without modifying bank account numbers, compromising customers' bank accounts, or otherwise wiring money directly from a user's account, how might you steal money from hacked accounts? [5 points]

- The attacker can login as the user and make the user deliberately lose money playing poker against the attacker's account.
- The attacker can spy on the user's cards and thus cheat to increase the opponent's chance of winning.