CS 166 ASSIGNMENT 4 SOLUTION

Overview

Marks on question:

- No mark: 1-part short answer question (the key point(s) for explanation is underlined)
- !: 1 part auto-graded question (answer in **bold**)
- +: n-parts question with autograding part and explanation (justify or show calculation) part. The answer for autograding part is in **bold**, the key point(s) for explanation is underlined.

Basic rules of grading: Full points if your explanation/equation was correct, even if the answer is wrong (because you already got points deducted in the fill-in-blank part). And you will get 0.5 (on the whole question, not each sub-question) as long as you wrote something. Those 0.5pt sub-questions and most of the 1pt sub-questions are all-or-nothing.

Week 8 - Authentication (Q1 - Q13)

!Q1: Biometrics vs. Password (L14 P4 & L15 P6)

I. Being authenticated: prefer **biometrics** since they don't need to memorize other things and passwords are not secure;

II. Authenticate others: prefer **passwords** since no extra hardware needed and it's cheap and convenient to set up passwords.

!Q2: 2 Phases of Using Biometrics (L14 P7)

I. **Recognition** phase needs process fast but can sacrifice some accuracy;

II. **Enrollment** phase sacrifices some speed to ensure precision.

III. Subjects will be more cooperative if the biometrics is used for **authentication** (as they want to be authenticated after).

+Q3: Fraud Rate vs. Insult Rate (L14 P8)

I. Fraud rate: Trudy mis-authenticated as Alice, that is, expect="negative" but actual="positive", so it's **false positive** rate;
Insult rate: Alice not authenticated as Alice, that is, expect="positive" but actual= "negative", so it's **false negative** rate.

II. 6 Alice & 4 Trudy so expects 6 positive and 4 negative.
But there's 1 false positive (Trudy mis-authenticated as Alice),
and 2 false negatives (Alice not authenticated as Alice)
So, fraud rate = false positive rate = FP / N = 1 / 4 = 25%;
Insult rate = false negative rate = FN / P = 2 / 6 = 1 / 3 = 0.33%

!Q4: Equal error rate (L14 P8)

Equal error rate (EER): rate where fraud == insult, and the lower the better.

In this question, you cannot get the EER, therefore, the answer is “**Can't tell**”

!Q5: Iris Code Hamming Distance (L14 P13)

Hamming distance = # of non-match bits / # of bits compared.
And if it's < 0.32, we consider it as a match.

In this question, Alice = 100110, X = 100010, and Y = 101100
Then d(Alice, X) = (100110, 100010) = 1/6 = **0.17** < 0.32,
d(Alice, Y) = (100110, 101100) = 2/6 = **0.33** > 0.32,
and therefore, the system would open the door for **X**.

!Q6: Key vs. Password (L15 P7)

I. 32-bit key: each bit has 2 possibilities (0 or 1), so there are $\mathbf{2^{32}}$ possible keys.

II. 8-char password and 16 possible chars: there are $16^8 = (2^4)^8 = \mathbf{2^{32}}$ possible passwords.

III. Since key is chosen at random will password is not, the **key** is more secure than a password with a same “keyspace”.

Q7: Best Way to Create a Password (L15 P8)

I. Using a passphrase to derive a password may be the best way.

II. Since it’s not too easy to guess, nor too hard to memorize (or easy to memorize but hard to guess)

+Q8: Hash vs. Encryption for Passwords (L15 P9)

8.1) False.

8.2) Encryption is not a good idea compared to hash when storing the passwords.
If using encryption, then we also need to store the key securely, which is an extra work. Moreover, if Trudy knows the key and gets the ciphertext, then she can decrypt it easily.
While for hash, even if Trudy knows the hash algorithm used and gets the hash values, she still can't easily recover the original passwords because of the one-way function property of hash functions.

+Q9: Case Studies of Cracking Passwords (L15 P11 & P12)

9.1) n-characters password and m choices per character, so there are m^n possible passwords in total.

9.2) Suppose Trudy doesn't have a dictionary.

I. Salted hash: need to brute-force half of all possibilities on average, so it’s $m^n / 2$.
II. Unsalted hash: also need to brute-force half of all possibilities on average, so it’s $m^n / 2$.
III. There's no difference, i.e., salt won't make Trudy's life harder. Trudy doesn’t have a dictionary so she can't pre-hash anything. Therefore, no matter if the hash is with salt or not, she needs to do an exhaustive key search (brute-force).

9.3) Suppose Trudy has a dictionary.

I. Unsalted hash:
If Alice's password is in the dictionary, it is pre-hashed, so no work needed, and the chance is p;
Otherwise, Trudy needs to compute $m^n / 2$ hashes on average, and the chance is $1 - p$.
Therefore, the work needed = $p * 0 + (1 - p) * m^n / 2 = (1 - p) * m^n / 2$.
Or only need to hash y passwords in Trudy's dictionary and the probability of success is p.

II. Salted hash: The difference happens when Alice's password is in the dictionary since Trudy can't use the pre-hashed values. That is, she needs to re-compute the hashes of the y passwords in her dictionary with the salt.
Therefore, the work needed = $p * y / 2 + (1 - p) * m^n / 2$.

III. The difference is the work needed to re-hash all passwords in Trudy's dictionary. Since m^n is way more than y, the work needed may not have make much difference for 1 attack.
However, Trudy won't just attack once. For unsalted hashes, pre-hash the password in the dictionary is a one-time job. For every attack, there's a p chance of success that she can find the hashed password in her pre-hashed dictionary.
But if the passwords are hashed with salt, Trudy needs to re-hash passwords in her dictionary.
Therefore, salt will make Trudy's life harder in this case, especially if she did more than 1 attack.
More attacks, more work on average for Trudy.

Q10: Other Topics for Authentication (L15 P14 & P15)

I. Single sign-on example: oneSJSU uses OKTA.

II. The timestamp is used as a salt, so the generated password can be different every time (like a one-time pad). Otherwise, there's no need for Alice to have the key generator if the generated password is always the same.
The problem is, if Alice and Bob's clocks are out of sync, then the authentication would fail.

Q11: Forget and reset password (L15 P9 & L8 P10)

I. The correct way is hash the password and store the hash value.

II. System 1 seems not using the correct way as the system send the actual password directly to the email. Then anyone knows the email address and have temporary access to the email account can get the password. Also, emails may be snooped.

III. Yes, system 2 can tell - if the hashes are the same, it means the passwords are the same, since the same password will be hashed to the same value based on the deterministic property of hash functions. Note that with/without salt doesn't matter, since salt is a public value and H(password1, salt) = H(password2, salt) iff password1 = password2.

Q12: Brute-force attack on password without dictionary nor salt (L15 P11 & P12)

I. For scheme 1 (not splitting): 14-char password with 32 possibilities per char,
so there are $32^{14} = 2^{70}$ possibilities in total;
For scheme 2 (splitting to 7-7): for each part, there are $32^7 = 2^{35}$ possibilities. Note that the work between each part is independent, so you should add both parts' work, not multiply.
Therefore, total number of possibilities is $2^{35} + 2^{35} = 2^{36}$.
$2^{70} > 2^{36}$, so scheme 2 is easier to crack.

II. If scheme 2 is used, and the length of password = 10, then the second half only contains 3 non-null characters. That is, Trudy can get the last 3 characters easily (only $32^3 = 2^{15}$ possibilities), then she may be able to use that information to narrow down the possibilities for the first part.
While for a 7-character password, there's no such shortcut and there are $32^7 = 2^{35}$ possibilities.
Therefore, using a 10-character password may be less secure than using a 7-character password.

Q13: Set password in different accounts (L15 P10 & P12)

I. If use the same password for all 5 accounts, then the possibility is the same as the probability that any given password is in Trudy's password dictionary, that is $1/5 = 0.2$.

II. If use 5 different passwords, we can use the "complement" technique to find the probability.
That is, P(A) = 1 - P(A'), where event A' is the "opposite" of event A.
The probability we want to know is "at least one password is in the dictionary"; the opposite is "none of them is in the dictionary" = "all 5 are not in the dictionary".
For each one password, 0.2 probability it's in dictionary, which means 0.8 probability it's not in it.
So, the probability of all 5 passwords are not in the dictionary is 0.8^5.
That is, the probability that at least 1 password is in dictionary is $1 - (1 - 0.2)^5 = 1 - 0.8^5 = 0.67$

III. The approach with lower probability is use the same password, but it's not better. Because, once Trudy knows one of your passwords (let's say, from a system stores original password), then Trudy also knows your passwords for other accounts.

IV. (Open answer) No specific answer, full points as long as your answer was reasonable. For example...

- If use one password for all accounts, people may choose a complex password as there's only one to remember. But once one of the accounts has been compromised, all accounts are in danger.
- If use different passwords, even if one of the accounts was cracked, the other ones will still be fine. But people may be lazy and come up with simpler passwords as need to remember more passwords.
- I personally prefer having a 2 distinct passwords approach, since I only have 2 passwords to remember and as long as the security-important one is not comprised, it's ok. But this approach requires extra effort to differentiate the security important site from non-important site. Moreover, I may accidentally set the complex password on a site that doesn't store the password securely.

Week 9 - Authorization & Other Access Control Topics (Q14 - Q24)

!Q14: Orange Book vs. Common Criteria & Covert channel of MLS (L16 P8 & L17 P9)

- Orange Book: also gives some guidance about how to design more secure products
  Common Criteria: read ONLY if you want to sell your product to the government.
- Covert channel(s) exist as long as we want to share resources (which is what an information system is for). So, there's no way to prevent them. We can only reduce the covert channel capacity.

!Q15: Basic Terminologies (L14 P2 & P6 & L17 P10)

- For an unknown fingerprint, find the owner by searching in a fingerprint database: Identification;
- For a fingerprint, decide whether give the owner access to a system or not: Authentication;
- After giving access to a person, check if the behavior seems normal or not: Intrusion detection;
- After giving access to a person, check if he/she can read a file or not: Authorization.

+Q16: Lampson’s Access Control Matrix (L16 P9 - P14)

16.1) The system requires frequent updates for permissions associated with a file:
use ACL - it's based on the objects (column).
The system needs to add/delete users often: use C-list since it's based on the subjects (row).

16.2) r - read; w - write; x - execute. Just follow the description: Alice can read the compiler but doesn't have any permissions on BILL. The compiler can read and execute itself. File BILL can be read and written by the compiler. (grey shading means given)

	BILL	Compiler
Alice	-	r
Compiler	rw	rx

16.3) From the above matrix...

I. Examples of ACL (column): BILL: (Alice, -), (Compiler, rw) OR Compiler: (Alice, r), (Compiler, rx).
II. Examples of C-list (row): Alice: (BILL, -), (Compiler, r) OR Compiler: (BILL, rw), (Compiler, rx).
III. Given the matrix, it is not possible for confused deputy to incur. Because Alice only has read permission on Compiler, so she can't execute the compiler to write data on BILL/
IV. Change the “Alice x Compiler” cell to something includes x (execute). If Alice can execute the compiler, she can “ask” the compiler to write data on BILL by invoking compiler to debug BILL.

+Q17: Multilevel security models (BLP vs. Biba) with compartments (L12 P15 & P17 & P19 - P21)

17.1) If follow BLP:
If you have SECRET{Dog} clearance, you can write TOP SECRET{Cat, Dog} classification objects.

17.2) If follow Biba:
If you have TOP SECRET clearance, you can write SECRET classification objects.
If you have SECRET clearance, you can read TOP SECRET classification objects.

17.3) These answers should also explain the choices in 17.1 & 17.2

I. Objects (file/data) have “classifications”.
II. Subjects (user, “you” in this case) have “clearance”.
III. For BLP, it’s “no read up, no write down”.
IV. For Biba, it’s “no write up, no read down”.
V. They follow different rules because they focus on different aspects of security. BLP focuses on confidentiality so it's restricting read-up; while Biba focuses on integrity so it's restricting write-up.
VI. For both models, based on the rules, you can write something you can’t read, which leads to a kind of paradox.

+Q18: Anomaly-based IDS (L17 P14 & P15)

18.1) Update “normal” statistics:

Pros: reduce false alarms (normal mis-identified as abnormal). Since the normal behavior may change over time, we need to update the statistics regularly so that the IDS adapts over time.
Cons: can’t prevent a patient attacker. Trudy can pretend to be Alice and remain undetected if she changes the behavior slowly. If the “normal” updates regularly, as long as Trudy doesn't stray too far from Alice's usual behavior every time, she can eventually convince the IDS that her evil behavior is normal for Alice with enough patience.

18.2) Combining several statistics:

Pros: provide a more comprehensive view of normal behavior
(or make it harder for Trudy to pose as Alice);
Cons: More complex to implement and less effective since it will require more computations.

18.3) Y = 1 - 0.15 - 0.45 - X = 0.4 - X since the sum of the frequency should be 1.

So, $\sum_{n=0}^{3}(O_n - R_n)^2 = (O_0 - R_0)^2 + (O_1 - X)^2 + (O_2 - R_2)^2 + (O_3 - 0.4 + X)^2 < 0.1$
Other than X, all other numbers are given in the question, putting the numbers, we got...

$$(0.2 - 0.15)^2 + (0.1 - X)^2 + (0.4 - 0.45)^2 + (0.3 - 0.4 + X)^2$$

$= 2(X - 0.1)^2 + 0.005 < 0.1$
Solving the above equation, we know any number with 0 < X < 0.31, and Y = 0.4 - X is correct.
For example, X = 0.25 and Y = 0.15.

18.4) Let d be the maximum difference on average,

then $\sum_{n=0}^{3}(O_n - R_n)^2 = 4d^2 < 0.1$
And we got $d < \sqrt{0.025} = 0.158$

Q19: CAPTCHA's paradox (L17 P16)

The computer generates and scores the tests that itself cannot pass.

Q20: Worse Than Nothing (L17 P19)

20.1) Applies to Encryption. For other two, a weak one at least can reduce some amount of information that leaks.

20.2) A weak encryption may be worse than no encryption at all. Since encryption implies that the data is important, Trudy can filter the encrypted data and easily break it if crypto used is weak. Better to hide the important information in a large pool of information, like hiding woods/trees in a forest.

Q21: 3 ways of inference control (L13 P19)

I. Advantage of randomization: will always show an answer so that the trend of data can still be analyzed without leaking the actual data, while the other 2 may prevent people from getting important data, such as medical research on a rare disease.

II. Disadvantage of randomization: if a precise answer is needed, then adding a random noise will cause a problem, while query set size control and N-respondent, k% dominance rule will provide accurate answers (if they can).

III. Attack on query set size control: suppose set A has a small size, set B has a large size, and set C = A ∪ B also has a large size. We can't get the data from A directly as A's size is too small, but we may calculate it using B and C. For example, we want the average of elements of A. We can calculate $avg_A = (sum_C - sum_B) / (size_C - size_B)$.

IV. Attack on N-respondent, k% dominance rule: Suppose set C = A ∪ B where A contributes the majority of the data (> k%) but has a small size (< N). We can't get the data from C directly based on the rule, but we may calculate it using A and B. For example, we want the average of elements of C. We can calculate $avg_C = (sum_A + sum_B) / (size_A + size_B)$.

V. Attack on randomization: we can match the property of data with another source to get a reasonable guess.

Q22: Open answer question for real-life MLS (L16 P15)

No specific answer, full points as long as answer is reasonable.

For example, any video streaming website. Levels may be "guest", "free", "paid". Everyone can get the basic information of the video (like the title) but people with "guest" clearance can only watch the videos that are classified as "guest"; users with "free" clearance (that is, registered but without a paid subscription) can watch both "guest" and "free" videos; and users with "paid" clearance can watch any video on that site. This encourages user to pay a subscription so they can watch as many videos as they want, and the website can make money.

Q23: Open answer question for real-world visual CAPTCHA (L17 P16)

No specific answer, full points as long as answer is reasonable.

For example, a slider CAPTCHA. The CAPTCHA is a slider on screen. The test is to slide the slider from beginning (left) to end (right) in a short amount of time (like 1 second). It is easy for human to do, especially on touch screen smartphone and laptops. But it's hard for a computer to do it in a short amount of time.

Q24: Open answer question for CAPTCHA (L17 P16)

No specific answer, full points as long as answer is reasonable.

I. They may want to register for web services such as emails so they can send spams, fake news, advertisement, even links to malwares to others.

II. The price varies based on the type of the CAPTCHA (and different websites). For example, from 2Captha (https://2captcha.com/), now the price is $0.26 (human solved) or $1.00 (auto solved) per 1,000 traditional CAPTCHAs.
By the way, in last Spring, it was $0.50 - $1.00 per 1,000 traditional CAPTCHAs; and when I researched it in Oct. 2019, it was $0.50 per 1,000 traditional CAPTCHAs.

III. Post a fake contest/test with CAPTCHAs without saying they are CAPTCHAs. For example, "80% of people can't solve this, but can you?", "test if you are sensitive to graphic", etc.