OBLIG 1 INF140

Overview of CyberSecurity:

Question 1.

[5 pts] Use your own words to explain each of the following security attributes in computer systems: confidentiality, integrity, authenticity, accountability, availability.

Confidentiality:
is a way to make data unreachable or shield certain parts of data, text etc. from an unauthorized user. For instance using encryption as a way of “hiding” information.

Integrity:
Integrity is used to keep information from being altered by an unauthorized source. For instance using a hash function and a cryptographic key.

Authenticity:
requires a user to verify their identity to gain access to information etc. for instance using email, passwords…

accountability:
Making an entity accessing or altering data, accountable for the actions. For instance tracing the actions back to the entity’s information.

availability:
Making the information available for the authorized users. No point in having secure measures when it’s not available to the people who need it. Poor availability programs are prone to hardware failures, and also attackers can keep the information from the authorized users.

Question 2. [4 pts] Suppose you need to protect two of the above security attributes, e.g., integrity and authenticity, of a student grade system in a university, which maintains a database of grades of all students and allows a student to check his/her grade. List possible vulnerabilities in the system that may be exploited by attackers, and possible threats that may violate these two security attributes.

Using integrity to protect the information from being altered. It needs to protect the data in the system from being altered by an unauthorized user (like a student hehe), and also that the information can not be viewed by someone other than the authorized ones(student and

teacher?). Authenticity is also used so that the teacher or the student have to identify themselves before accessing the data. For instance using UiB login that is connected to your identity.

The integrity of the system needs to be equipped for intentional, as well as unintentional attacks like: user errors, data loss.

These two together need to form a system so that the user is verified and cannot bypass the integrity attribute or at least be strong enough together to stop the attacker.

Question 3. [6 pts]

For the following assets, assign a low, moderate, or high impact level for the loss of confidentiality, availability, and integrity, respectively. Justify your answers.

(i) A law enforcement organization managing extremely sensitive investigative information;
Confidentiality - High, because the information is so sensitive
Integrity - Moderate, would affect the organization, but not ruin it
Availability - Moderate, again not the worst
(ii) A hospital managing patients' health records, which contain privacy-sensitive diseases and allergy information for each patient;
Confidentiality - Low (?) I mean you can't access anything by knowing if person is allergic, so i would say low-moderate
Integrity - Moderate/High, if some of the data is lost it may be bad in some cases. lik if the data says you're not allergic to penicillin, but you actually are. this could be bad
Availability - Moderate, again. If you're unconscious and a doctor needs to know something then yes this would be bad if they can't reach it
(iii) An organization managing public information on its Web server
Confidentiality- Not used
Integrity - Moderate the information is public so exact data is not important
Availability - Moderate, again not very bad.

Cryptographic Tools (60 pts)

Task 4

(1) text

The code is located at the bottom of the pdf file, and also in the Zip-file handed in.
the GREEN - encryption round 1
BLUE - Final Cipher text,
PURPLE - decryption of text2 after being through the transposition cipher and the vigenere
ORANGE - final decrypted text.

Encryption round 1:
EPZTKGKTUGPRGAXNREYLDECMZCPSJGUTGCNZAPMSJZTXZYXJTWRUJLDMOVIO
SBAB

Ciphertext:
WLITNCQDHQCHMVXQBGRGNQPHGPWGEOATEEXJJJLTXQHONLYKGXVAXTRHZVUA
CSLJ

('output from vigenere cipher round ', 1, ' is: ',
'EPZTKGKTUGPRGAXNREYLDECMZCPSJGUTGCNZAPMSJZTXZYXJTWRUJLDMOVIO
SBAB')
('output from vigenere cipher round ', 2, ' is: ',
'BGWEGHXZGPLEXQQVRWIXVCHUGGTJAHOANENJXMNCQOCJTVLSPAQLRXYLHTD
THQKJ')
('decryption round', 1, ' transposition: ',
'BGWEGHXZGPLEXQQVRWIXVCHUGGTJAHOANENJXMNCQOCJTVLSPAQLRXYLHTD
THQKJ')
('vigenere decryption: ', 1,
'ZPYPNTRIGAEGPYLBEURZGJTOPGECCZWVTRLSZXUOKXCUMXDAKGDJAZJSTNMT
SJMB')
('decryption round', 2, ' transposition: ',
'EPZTKGKTUGPRGAXNREYLDECMZCPSJGUTGCNZAPMSJZTXZYXJTWRUJLDMOVIO
SBAB')
('vigenere decryption: ', 2,
'CYBERSECURITYISTECHNOLOGICALLYCOMPLICATEDITISAPROCESSNOTAPRODUC
T')
('decrypt',

CYBERSECURITYISTECHNOLOGICALLYCOMPLICATEDITISAPROCESSNOTAPRODUC
T')

code:

located at: /Users/nooralindeflaten/Downloads/INF140Oblig1.py

my comments may be confusing so:
i use the function cleanString to fix the string into an all letter uppercase string.

KeyGen is used to generate the key to match the vigenere input.

vigenere is used to encrypt a text with the key.

columnGen
generates the columns that will be used for the transposition cipher.

transposition
this was supposed to be the final function of this encryption but the final output is located in: fixCipher, after i cleaned it up to match the rest of my code.

The general overview is located in the toyCipher function (line 155)

after this i have the functions for decryption.
Vigenere decryption, decrypts a vigenere text with a key.
i also used this in task10.

the functions:
transDecryptio, and vigenere decryption are used together in the
Decrypt function.
The two functions: decryptionkey, and stringfix are never used,but i'm scared of deleting code :)

Question 5. [3 pts]

Explain the six design requirements of cryptographically strong hash functions.

a hash function is a function f(x) which accepts a message x as input of some arbitrarylength and outputs a fingerprint of fixed length.

a good hash function has to satisfy three conditions:

1. Preimage Resistance (one-wayness)
2. Collision resistance (second pre-image resistance)
3. Strong collision resistance (collision resistance)

Design

the 6 requirements lead to trap-door one-way function

1. The same message always result in the same hash
2. Every function has unique inverse
3. calculation of function is easy
4. calculation of inverse is infeasible
5. it's infeasible to find two different messages with the same hash values
   from lecture notes:
   Y = fk (X) easy, if k and Y are known
   X = f −1k (Y) easy, if k and Y are known
   X = f−1k (Y) infeasible, if Y is known but k is not
6. PreImage Attack resistente.

Task 6:

This CF function is not considered a strong hash function.
It’s not weak collision resistant, I can easily find messages that have the same hash Value.
I could then modify the original message to be the equal message and send it back without being detected.

TASK 7:

anwsers are yellow.
The code is located in the pdf and also in the zip.
i did this in python, but the deriving i did by hand so:

['HELEFTTWENTYMILL', 'IONUSDOLLARSTOHI', 'SBELOVEDCHILDREN']
('First Block: ',

[[7, 4, 11, 4],
[5, 19, 19, 22],
[4, 13, 19, 24],
[12, 8, 11, 11]])
('RunningTotal: ', [2, 18, 8, 9])

('new Block: ',
[[4, 11, 4, 7],
[19, 22, 5, 19],
[24, 4, 13, 19],
[11, 11, 8, 12]])
[6, 22, 4, 5]

ROUND 1: [8, 14, 12, 14]

[[8, 14, 13, 20],
[18, 3, 14, 11],
[11, 0, 17, 18],
[19, 14, 7, 8]])

[[14, 13, 20, 8],
[14, 11, 18, 3],

[18, 11, 0, 17],
[8, 7, 14, 19]]

('RunningTotal: ', [4, 5, 25, 5])
('new Block: ', [[14, 13, 20, 8], [14, 11, 18, 3], [18, 11, 0, 17], [8, 7, 14, 19]])
('First Block: ', [[18, 1, 4, 11], [14, 21, 4, 3], [2, 7, 8, 11], [3, 17, 4, 13]])
('RunningTotal: ', [11, 20, 20, 12])
('new Block: ', [[1, 4, 11, 18], [4, 3, 14, 21], [11, 2, 7, 8], [13, 4, 17, 3]])

(1)

Final output:
(['C', 'Q', 'C', 'Y'], [2, 16, 2, 24])

Trying to implement a way of finding a message by finding the pattern.

(['C', 'Q', 'C', 'Y'], [2, 16, 2, 24])

0, x, y, 24
z, 0, 0, 0

x, y, 24, 0
0, 0, z, 0

21, 7, 9, 24
7, 9, 24+21, 0

0 + x=7 +21 mod 26 = 2
7 + y(9) mod 26 = 16
(9 y) + 24+21 mod 26 = 2

0, 7, 9, 24
21, 0, 0, 0

21, 7, 9, 24

7, 9, 24, 0
0, 0, 21, 0

7, 9, 45, 0

2, 16,2,24
C,Q,C,

(2)

final anwser:

m1 = "AHJY VAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA"

Code:

is located in file:
/Users/nooralindeflaten/Downloads/INF140Task7.py
The comments in the code may be confusing but:

function:
cleanString -> fixes the string and ignores punctuations, whitespaces etc.
Blocks -> divides the message into blocks of 16.
numberBlocks -> takes the message and divides it into four by four number tables
running total -> calculate the running total of a given block.
rotation -> rotates a given block and returns the running total of the rotated block.
sumRunningTotal -> takes a runningtotal and updates the sum with a new one
hash -> perform all the operations and produce the final runningtotal and word.

Task 8:

(1)

p = 3; q = 17, e = 5; M = 9 [3 pts]
cipher = 42, decryption = 9
p = 5; q = 17, e = 7; M = 5 [3 pts
cipher = 10, decryption = 5.

I used python code to find this.

(2)

CRYPTOGRAPHY:

```
['0xa7', '0xc7', '0x43', '0x3f', '0x2e', '0x28', '0x62', '0xc7', '0x41', '0x3f',
'0x20', '0x43']
```

Code

this task is located at:
/Users/nooralindeflaten/Downloads/INF140Task8.py

running the “RSA” function with the values gives me the anwser for (1)
and running the “Encryption” function on the values from (2) i get the encrypted version of the word, and by using convert i got the hexadecimals. unless you wanted me to sum them together and get a single one. I wasn’t sure.

TASK 9:

solved in python.
Use the diefell function in “/Users/nooralindeflaten/Downloads/INF140Task8.py” and the values given to run.

(1)

("Bob's private: ", 34, " Bob's Public: ", 8)
("Alice's private: ", 28, " Alice's public ", 10)

(2)

('shared secret key (g**ab mod p) :', 197)

Code:

to get these i just ran the dieffel function in the same python file as the code for task 8.

Task 10:

(1)

The vigenere cipher key used by Tobias is:
HEI

(2)

The original message was:
MOSTPEOPLEARESTARTINGTOREALIZETHATTHEREAREONLYTWODIFFERENTTYPESOFCOMPANIESINTHEWORLDTHOSETHATHAVEBEENBREACHEDANDKNOWITANDTHOSETHATHAVEBEENBREACHEDANDDONOTKNOWIT

(3)

usually the basic calculation of this is: 2**keylength, but in this case i just think it’s all the possible 4 letter combinations you can have from the given table. which i did and i got:
In the worst-case scenario the attacker will have to try all keys:

10 000 keys.

Code:

the code is located at:
/Users/nooralindeflaten/Downloads/INF140Task10.py

functions
cleanString, keygen and vigenere decryption are all basic functions for vigenere decryption like I used earlier.

First I had the key table, and made a list of all the possible 4-letter combos of these letters. The “keyToString” method takes a key and returns the integer of it. like in CCG -> 225. Yes I know it returns an int despite its name.
the convert Key just takes a number and returns the string for it.

so then the FunCipher method checks the key list until it finds the right one. There are probably more mathematical correct ways of solving this task, but testing all values is very fun :)

i mean for instance by modular inverse, prime factorization methods etc.

User Authentication (25 pts)

Task 11:

qwerty:

Very weak.
It’s in wikipedia's top 1000 most common passwords list :/
it only checks of the box for the “one lower-case letter” condition

Einstein:
Weak,
Also in the Wikipedia list.
at least one lower-case
at least one upper-case (combo of these is good)
length is at least 8 characters
*laptop_admin#
Medium
Good password, has almost everything.
length is at least 8 characters, there is special characters
and also a combination of more than one special char, and letters.

wysiwyg:
again. Very weak, easy to check the dictionary if a pattern is detected.
only lower-case chars. length is 7

wy$1wYg:
medium, only thing missing is that there should be a longer password.