**2022/2023 3010 Computer Security Semester 2**

**Week 11 Tutorial**

**Authentication/Password/Crytpo Tutorial**

1. Is it always a bad idea to write down your password? Is there a way to do it more safely?

Ans:

Not always a bad idea.

Write down all your difficult pswds on a paper.

Then lock it in a safe or locked office drawer.

Hackers will not be able to access your office, and open your safe or locked drawer.

But if he can install keyloggers into your pc thru your carelessness, then your secure passwords will be captured by it and render your long secure pswds USELESS.

1. (a) Verify the computational complexity of these 3 entries: 10 character lower case alphanumeric, 9 character alphanumeric, 8 character printable keyboard characters in slide 16 of the Password Slides. What do you notice?

Ans:



I will just work on 9 char alphanumeric. U can try the rest, imitating me.

# of possible 9 char alphanumeric

= (62)^9 = 2^k, solve k.

Take log on both sides,

9 lg62 = k lg2,

So k = 9lg62/lg2 = 53.6, as expected (see table above)

(b) The fastest Software Password crackers has cracking speed of up to 16 million passwords per sec (take it as 2^24) on a 3GHz PC. Estimate the time taken to compute cracking time of

(i) 10 numeric characters passwords

(ii) 8 character lower case alphanumeric passwords

(iii) 8 character alphanumeric passwords

(iv) 8 character all 95 printable characters passwords

Ans:

doing computations for 8 char alphanumeric = 2^47.6

U can try the rest, imitating me.

Since cracking speed of pswd crackers is 2^24 pswd per sec,

Time needed to crack 8 char alphanumeric

= 2^47.6/2^24 sec

= 2^23.6 sec

= (2^23.6)/60 mins

= (2^23.6)/3600 hours

= 3532 hours!

= 147 days, almost 5 months of cracking time on 3 GHz PC!

If attacker uses 50 3GHz pc, it will ONLY take 3 days!

For this reason, we use at least 9 char printable PSWDS.

Please dont forget that attackers can use assembly of PCs to crack your password hash.

Please also remember that attackers can use hardware password crackers (abt 100-1000 times faster than 1 PC) to crack your password hash.

Lessons learnt: Always use a much longer pswd: at least 9 chars (95) printable chars

1. Suppose that a certain email system uses hash of time when document is encrypted & emailed together. Example: time 20220203073000 means(3 February 2022 7:30:00)

* Alice encryption software works as follow:
* It uses 256-bit AES-encryption.
* Key used is 256-bit key, namely SHA256(Date & Time of email).
* When Alice use this encrypted email document and want to send to Bob, system looks at the time, then perform SHA256(Date & Time of email) and use this as 256-bit key for AES.
* Bob downloaded document into his pc. One day his pc is hacked and this document is in hands of hacker. Assume that hacker knows encryption algorithm and hash function used but not the date and time when document is sent.
* (a) Explore if hacker has a chance to read this document.
* Ans: Lets make some assumptions, say his document was written some time in 2022. Since the format is measured up to seconds, we only need to compute how many seconds there are in 2022. That will be the number of possible AES keys. Check this number is very small compared to 2^256.
* (b) What are the key lessons we can learn from this implementation of strong algorithm AES? Ans: Strong algorithm alone is not enough. Keys generated must be of as high entropy as possible and cryptographically random.

1. Why do we want to use slow hash for password hashing?

Ans: Slow hash is to slow down attacker’s bruteforce or dictionary attack



Ans:

Let x = (x1,x2,…,x7). Solve xA = (0,1,0,1).

Write the system down as a system of 4 equations with 7 unknowns.System is already in echelon form.

Use guaussian elimination, can express x4 in terms of 3 free variables x5,x6,x7 – 8 possible choices.

Then similary express, x3,x2,x1 in terms of x5,x6,x7. Draw a table.



1. Why do we need 256-bit hash length instead of 128-bit length to pair with AES-128?

Ans:

Hash space = 2^256

By birthday paradox, hash will repeat approx. after sqrt(2^256) = 2^128.

So cracking 256-bit secure hash takes 128-bit complexity, same as AES128.

1. Explain why a long codeword such as hippopotamus is not secure for use in Vignere cipher. Ans: Dictionary word.

Attacker just need to bruteforce each dictionary word as a possible keyword and do decryption. Since all operations are simple additions, even if dictionary has 10 million words, the bruteforced plaintext can be bruteforced out within minutes since PCs can crack millions of such cases within seconds, at most minutes.

To flush out the correct plaintext, use some frequently occurred words such as “THE” and search for this keyword. It shud appear often in the correct plaintext.

1. Keystream of {0,1} generated by pseudo-random number generators will be periodic. Secure keystream must necessary have long period. Show that even with extremely long period, some keystreams may not be suitable for use to generate encryption keys.

Ans:

Let N be as large as you want.

Consider keystream 11…100...0 (N 1s followed by N 0s, and repeat.

Clearly long period, but totally predictable, so bad keystream.

So long period alone NOT ENOUGH to guarantee secure keystream.

1. Explain why for CTR mode in block cipher encryption, your counter values must all be distinct to be secure.

Ans: If CTRi & CTRj is the same, then the XORing of the corresponding plaintext is EQUAL to the XORing of the associated ciphertext! Keys have disappeared because of XORing! Analogous to one time pad re-use.

1. If the hash reflected on the web page of download coincides with your computed hash, does it always mean the files have not been tampered with, assuming the hash used is a real secured hash?

Ans:

The only thing that can go wrong in such an instance:

If webpage has been hacked, the hacker can simply replace the software by malware, and then replace the original hash checksum with the new malware hash checksum.

Thus very impt for administrators to regularly check software and checksum have not been tampered with.