

+) 1 million users  
+) service provider generates a random pass for each user  
+) password contains any printable ASCII chars: 95 printable ASCII characters  
+) password length undecided  
+) password hash stored = (user + password) -> SHA-256

Plot as functions of password length (n)

**a) entropy of a password**

random n-character ascii printable passwords:

H = log2(95^n) = n⋅log2(95) = 6.56n bits

**b) cost of cracking at least one user’s password from the database**

From answer of c) We can see that the number of seconds to crack is 95^n/3∙10^9.  
In days: 95^n/(259200∙10^9).   
Assume that the hacker can test all the hashes for all million users at once, the chance of cracking one random user will increase by 1 million times  
=> days of cracking at least one random user password: 95^n/(2592∙10^17), which is also the costs as each day is 1 dollar rent.

**c) Time to crack a random 10-character (printable ASCII) password from its SHA-256 hash?**(Giga hashes per second) (GH/s) One billion hashes computed in one second. The term typically refers to cryptocurrency validation and mining, for example

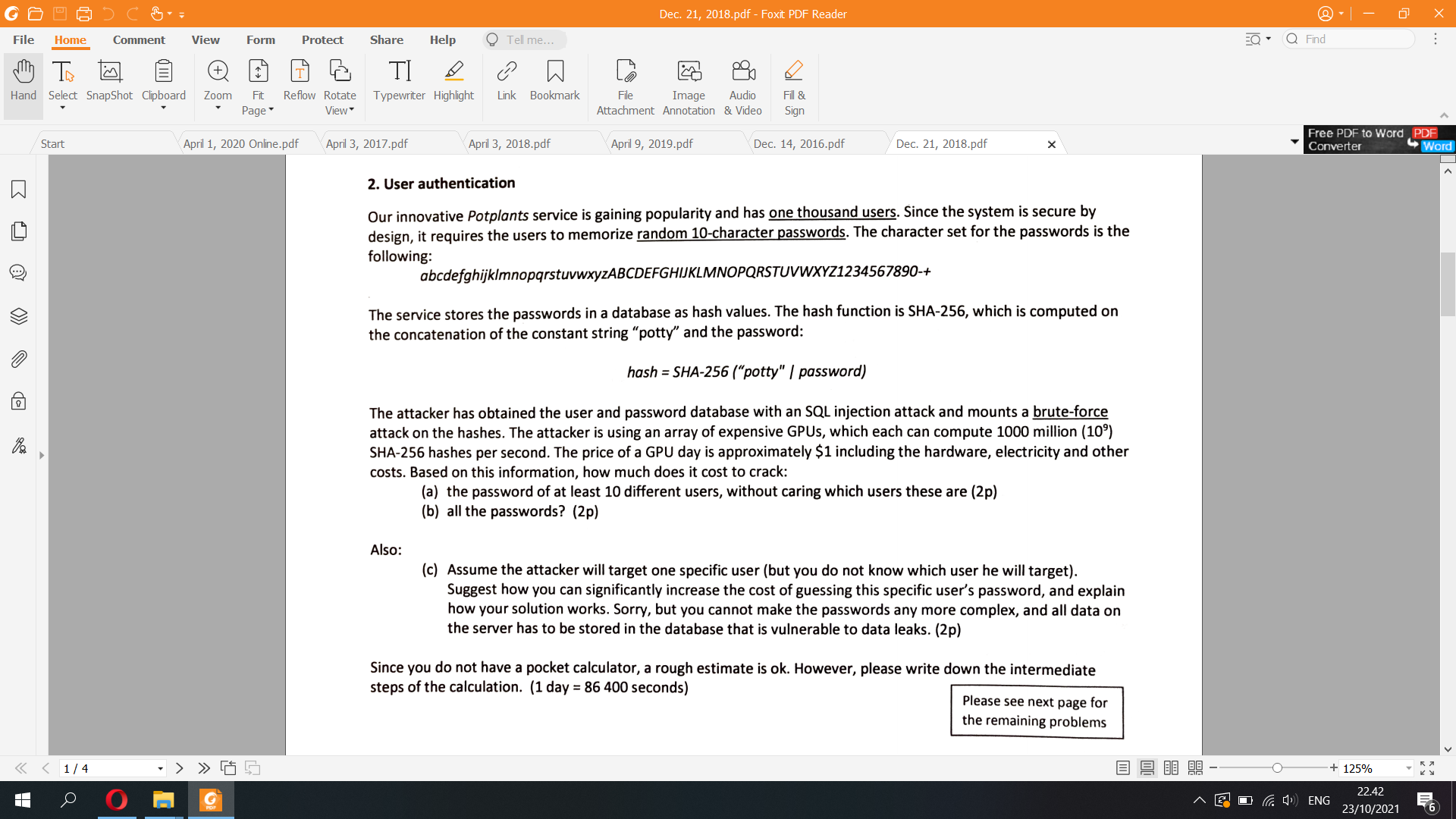
▪ How long does it take to crack a random 10-character password (printable 8-bit ASCII) from its SHA-256 hash?

▪ 95^10 = 2^65.7 = 5.98736∙10^19 possible passwords. Thus, brute-force cracking takes at most this many trials (50% on average)

▪ Top-end GPU computes up to 3 GH/s (SHA-256)

– Thus, cracking the password takes 95^10/ 3∙10^9 = 1.99∙10^10 CPU seconds = 230324 CPU days (answer)

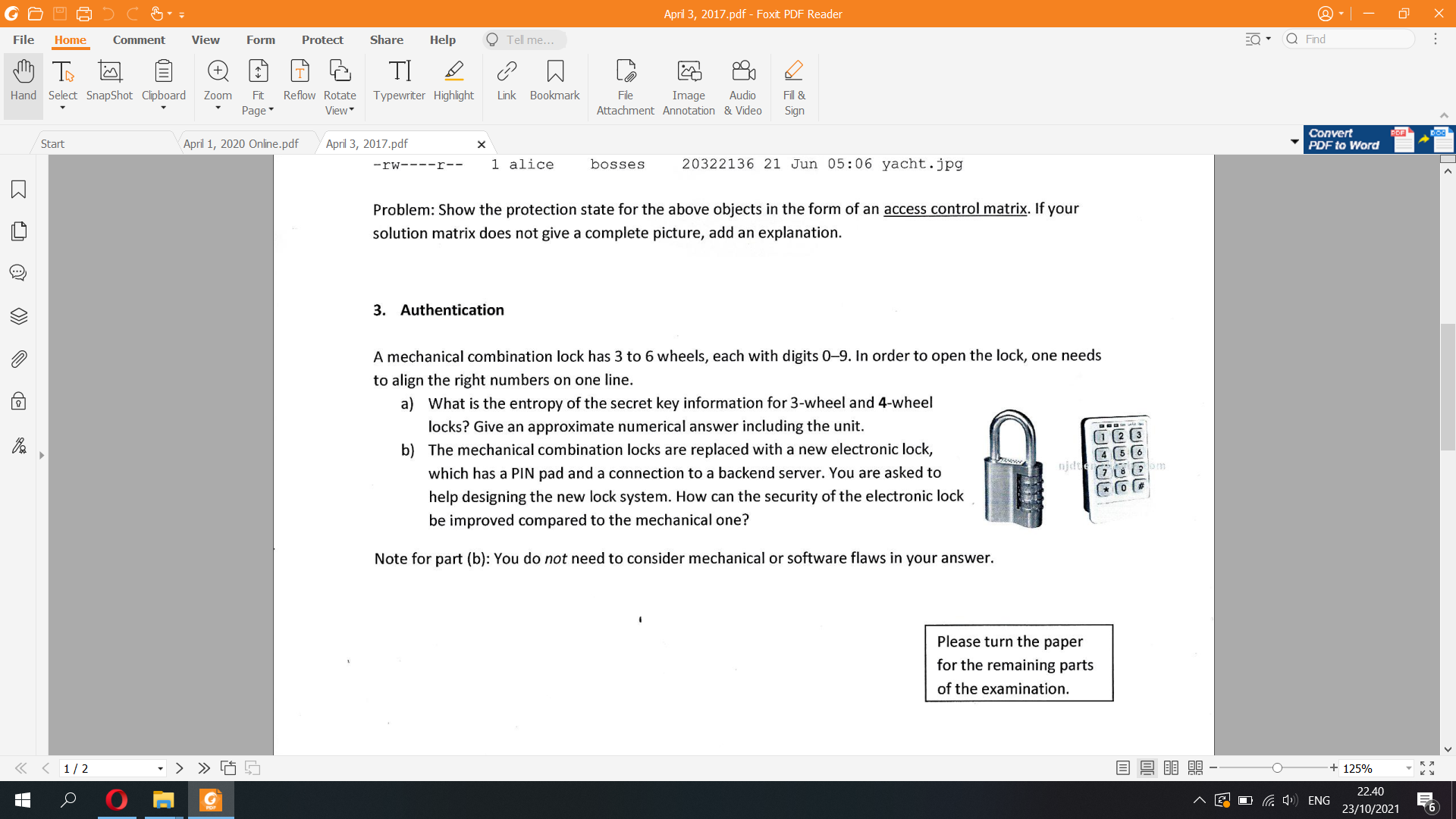
d) Given that the attacker have access to the database that has user name, cracking a hash means he can deduce the password from the username and the cracked hash, which is particularly dangerous. To improve this, websites can add a salt before the hash and remove username from it, such as SHA-256 (salt | password) so it will be very difficult for the attackers to reverse engineer for a password of a particular user. If possible, store the salt from another table or server from the passwords.

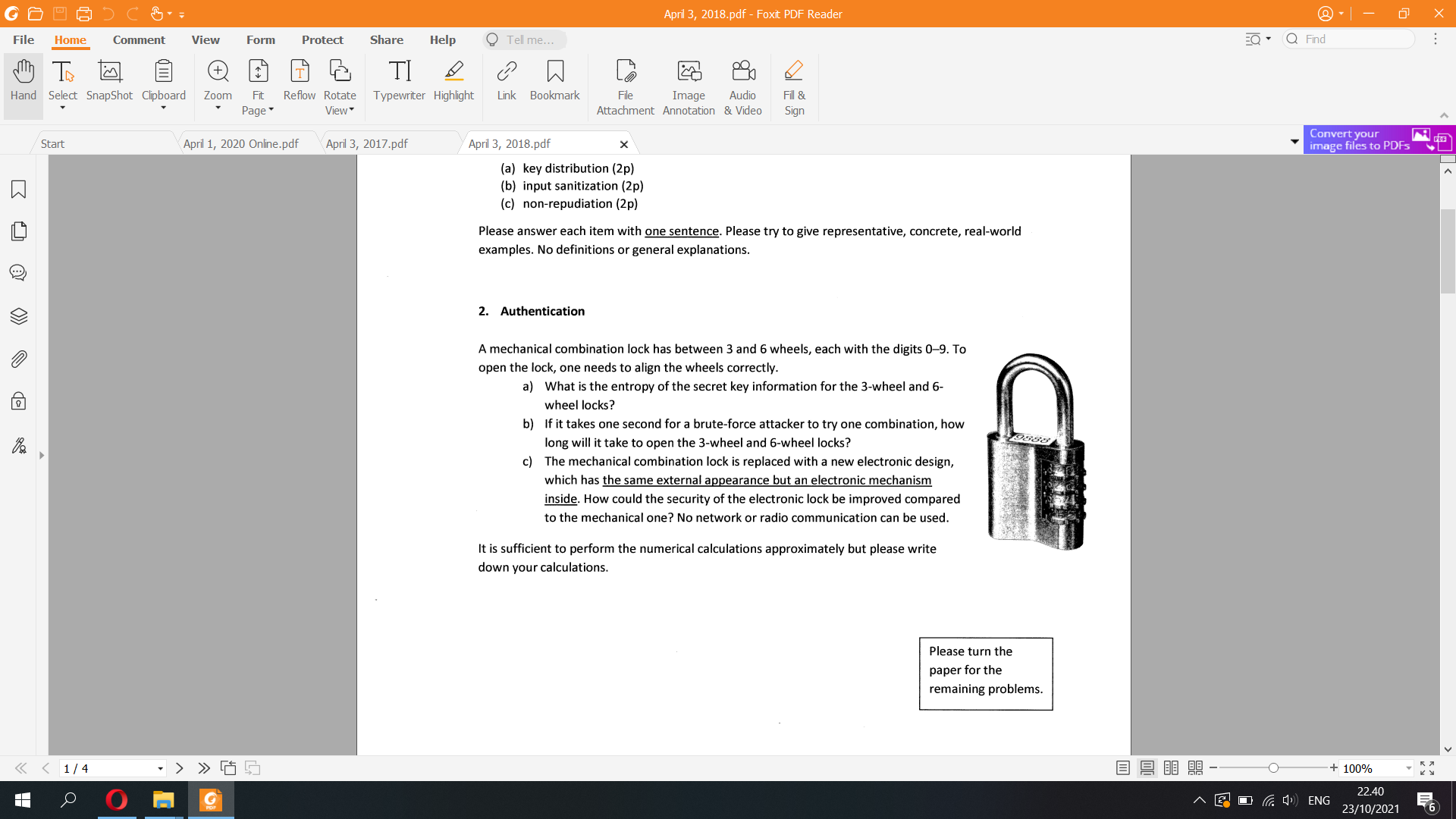


a) There are 64 different chars in the password. Password is 10 chars long => 64^10 different passwords for each user => seconds of hacking: 64^10/10^9  
Assume that the hash computed by the GPU is compared to all 1000 users at the same time => Chances of cracking on random user’s password on potplant increase by 1000 times => time of guessing one random user: 64^10/10^12 seconds  
=> Hack at least 10 users’ password means the time above will be multiplied by 10, assuming the chance of hacking each different users password are independent of each other.  
=> 64^10/10^11 seconds or 133.4 days => 133.4$ to hack

b) Same probability of hacking one user. => 64^10/10^9s => 13343 days => 13343$ to hack all

c) Since potplant has only 1000 users and the speed of hashing doesnt compromise user’s experience, a fast hashing function isnt necessary => potplant should use a slow hash function. Also, instead of just using the simple “potty” string, the website should use a salt that is computed in the hash together with the password  
For example: hash = SHA-256 (password | SHA-256 (password | salt))  
=> Number of possible hashes increase exponentially bigger





a) H = log2(10^3) = 9.96 bits for 3 wheel  
H = log2(10^6) = 19.93 bits for 6 wheel  
b) 10^3 seconds and 10^6 seconds

c) Mechanical locks

Cons:

Reset and maintenance requires the expertise of a locksmith

Requires physical keys which may be lost

Takes more time when unlocking, not ideal for emergencies

Locks can be picked or unhinged

Digital Locks

Includes wireless locks, the NS1000, biometrics, RFID, keypad, etc.

Pros:

Unknown length of password  
Password can be reset

Quick access, ideal for emergencies

LED indicators

No need for physical keys (except for RFIDs)

Includes tamper alarms

Has warranty for parts (depending on the supplier or manufacturer)

User-generated PIN codes

Can track data whenever used

Can be easily managed without the help of a locksmith or 3rd party

Easy to reset in the event PIN codes are forgotten

