Assignment 6

1.

Hash function	C	P	P2	F
a	No, the output always is the same bit string.	Yes	No, the output always is the same bit string.	Yes
b	Yes, the whole output relies on the whole input.	No, it is very easy because the message is repeated.	Yes	No, the output size relies on the input size.
c	Yes	Yes, you can find H(x) but not x itself.	Yes, because H(x) has this property, h(x) also has this property	Yes, because the length of H(x) is fixed, the length of h(x) is also fixed.
d	No, when two strings x and y have the same length, the hash function returns the same value.	No, because the length of the input is used, it is possible to brute force (try all lengths).	No, because the length of the input is used, it is possible to brute force (try all lengths).	Yes
e	No, when two strings, for example 1 and 01 are hashed, the output is the same.	No, when a string, for example 1 is hashed, x can have value 0….01.	No, when two strings, for example 1 and 01 are hashed, they have the same output.	Yes

2.

a.

No, a_i is an element of {1, 2, 3, 4}. When Bob hashes all these values, he can get the output of the hash function and compare it with the email he receives from Alice. He can then see which answer Alice chose.

b.

Yes, now it’s no longer possible to try out all the anwers.

c.

Yes, they both have the hashed output. It does not matter which of the two verify the other’s answer first.

d.

$A \rightarrow B : h(a_i \,||\, salt_{ai}) = x_i$

$B \rightarrow A: h(b_i \,||\, salt_{ai}) = y_i$

Now, each question they pick their random salt and send it back with their answers in the verification phase.

e.

(C): It is important because otherwise Alice can find H(x’) = H(x) where x’ is another answer to the question.

(P): It is important because otherwise Bob can find x (the answer of Alice).

(P2): It is important because otherwise Bob can accuse Alice of lying (by finding another x’ for which H(x) = H(x’).

3.

a.

Each digit can contain 62 possible characters (10 + 26 + 26).

A 10 digit passcode can contain 62^{10} different combinations.

62^{10} = 839299365868340224 (839 299 365 868 340 224).

1 billion = 1 000 000 000

839299365868340224 / 3000000000 = 279766455 seconds

279766455 seconds = 77712 hours = 3238 days to try all combinations.

It would take 3238 / 2 = 1619 days to break a fully random 10 digit passcode on avarage.

b.

Assume there are no identical passwords.

There are 43 million entries.

62^{10} / 43 000 000 = 19518589903

Per hash, we have a 1 on 19518589903 chance to find a matching hash.

19518589903 / 3000000000 = 6,5 seconds to find a matching hash.

On average, it takes 6,5 / 2 = 3,25 seconds to find a matching hash.

c.

We have 62^6 possible combinations.

62^6 = 56 800 235 584

56 800 235 584 / 10 / 3600 / 24 = 65 741 days to try all combinations

On average, 65 741 / 2 = 32 870 days to break a password.

d.

(P), they first have to dehash a stored hashed password.

e.

Dropbox needs to hash the SHA1 hashes with their new hash function. When a user then logs in, they need to first hash the password with SHA1 and then hash the SHA1 hash with the new hash function.