Feedback on Topic 5 Ex

Exercise Question – E5.1 - Q

For a hash value to be used as a cryptographic checksum, it must be protected with a secret, as it is clear, from the above, that a hash function does not have an embedded key.

Assuming that a sender, s, is to send a message, M, to a receiver, r. Propose as many different methods as you can to protect the hash value of M to assure the authenticity of M. Comment on the suitability/applicability of each of the methods you propose.

Exercise Question – E5.1 - A

- Assumptions: H is a hash function, M is a message to be protected, K a symmetric key shared between the sender and the receiver, {PUs, PRs} a pair of public/private key belonging to the sender of M, {PUr, PRr} a pair of public/private key belonging to the recipient of M, E is an encryption function, and Tag is the authentication tag.

Exercise Question – E5.1 - A

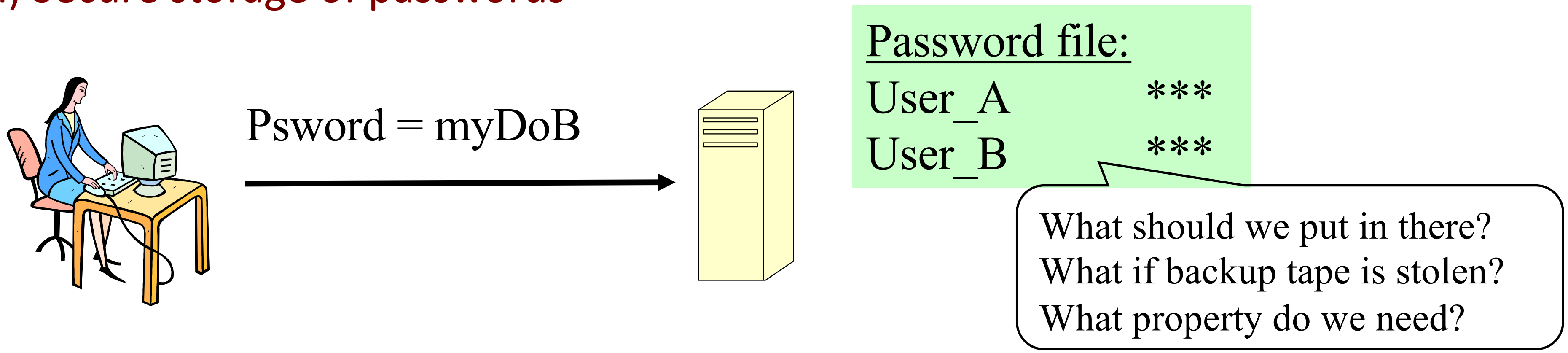
The four different methods of ensuring the authenticity of M:

1. M||Tag, where Tag=H(M||K)
2. M||Tag, where Tag=E(K, H(M))
3. M||Tag, where Tag=E(PRs, H(M))
4. E(K, M||H(M))

Exercise Question – E5.2 - Q

For each of the following applications, please identify what property(ies) the hash function needs to have to ensure the security of the application.

(i) Secure storage of passwords



Psword = myDoB

Password file:

User_A ***

User_B ***

What should we put in there?
What if backup tape is stolen?
What property do we need?

(ii) Protection against viruses

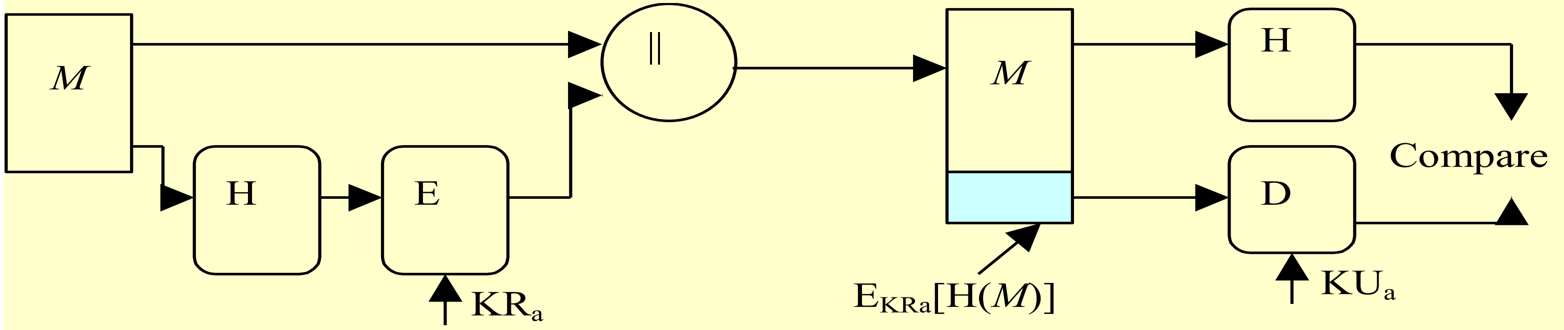
- A software manufacturer wants to ensure that the executable files are received by users without modification.
- They send out each file to users and publish its hash value on an authentic website.

Exercise Question – E5.2 - Q

(iii) Digital signatures

One party can sign a message, M, and many parties can verify.
Such applications include contract signing, code signing, etc.
A raw signature scheme only signs a few hundred (e.g. 160) bits.
What properties do we need?

Integrity, authentication, and *non-repudiation* are provided.
*This is the essence of the digital signature technique.



Exercise Question – E5.2 - A

1. UNIX passwords stored as hash(password)

   One-wayness: hard to recover password, although weak collision resistance can make the system stronger.

2. Integrity of software distribution

   Weak collision resistance

   But software images are not really random… maybe need full collision resistance

Idea: given goodFile and hash(goodFile), it is very hard to find badFile such that hash(goodFile)=hash(badFile)

3. Digital signature requires both one-way and strong collision resistance properties to counter signature forgery and repudiation.

Exercise Question – E5.3 - Q

- This is a Coin Flipping Over the Telephone problem.

(i) Assuming there is only one car, and Alice and Bob have to decide who can have this car (only one of them can have it, i.e. they cannot share it). Alice and Bob cannot see each other, and they do not trust each other. So they have decided to make a decision by flipping a coin over the telephone. Design a protocol to support this using a hash function.

(ii) Identify any factors that you should consider to ensure the security of this protocol.

Exercise Question – E5.3 (hint)

Assumption: Alice and Bob agree that if the outcome is 1 then Bob takes the car, if it is 0 then the car goes to Alice.

Solution 1 (an insecure solution):

- Alice generates a random bit *b*: 0=heads, 1=tails.
- Alice asks Bob: heads or tails?
- Bob sends Alice his choice_B: ‘heads’ (or ‘tails’).
- Alice compares *b* with choice_B: if b=choice_B, then outcome=1; if not, outcome=0.
- Alice sends the comparison outcome to Bob.

What is wrong with this solution?
Please fix the flaw.

Exercise Question – E5.3 - A

Here are the protocol steps:

1. Alice generates a random number, *r* **(the first bit, *b*:** 0=heads, 1=tails**).**
2. Alice computes commitment, x=hash(r).
3. Alice sends *x* to Bob and also asks Bob: heads or tails?
4. Bob sends Alice his choice_B: ‘heads’ (or ‘tails’).
5. Alice compares *b* with choice_B: if b=choice_B, then outcome=1; if not, outcome=0.
6. Alice sends *r* and the comparison outcome to Bob.

Exercise Question – E5.3 - A

Answers to (ii):

The random number generator used by Alice should be truly random. Otherwise, Bob could guess r, putting Alice in a disadvantaged position.

The hash function used should be secure, i.e. one-way and collision resistant. Otherwise, if it is not one-way, Bob could benefit; if it is not collision resistant, Alice could benefit.