Homework 02

Due: 11:59 pm, Monday February 19

Please handin your homework assignment on gradescope as a PDF file with each problem on a separate page.

Public Key Encryption

Problem 1

Alice and Bob, both MI6 Agents undercover as Brown CS students, are secretly dating. In order to set up a meeting, they exchange encrypted messages using a deterministic public key encryption scheme (deterministic in the sense that encrypting a given plaintext multiple times will always produce the same ciphertext). Alice has a public key/private key pair, (PK_A, SK_A), and Bob has a public key/private key pair, (PK_B, SK_B). Whenever Alice wants to send a message, m, to Bob, she encrypts it using his public key and sends the ciphertext, $c = Enc_{PK_B}(m)$, to him. Similarly, Bob's messages to Alice are of the form $Enc_{PK_A}(m)$. Assume that they always meet at a Brown building and that the messages they exchange are always of the following form:

Bob: `CIT, 7:00pm`
Alice: `NO`
Bob: `GCB, 8:30pm`
Alice: `YES`

Assume that when referring to a specific Brown building, the name they use is consistent (i.e. they won't alternate between "SciLi" and "Sciences Library").

a) Trudy is Alice's curious roommate (who also secretly works for SPECTRE) who wants to find out about the secret dates between Alice and Bob. She knows both of their public keys, the form of their messages, and she can eavesdrop on the ciphertexts being exchanged. Describe how Trudy can find out when and where the next meeting is going to be, even though she is unable to learn the secret keys. Assume Trudy knows all possible Brown buildings.

b) Alice and Bob found out that Trudy can learn about their meetings. Describe a simple modification of the protocol in order to avoid the attack that you described.

Passwords

Problem 2

Work Factor is an asymptotic measure of the number of operations required to perform some calculation. In particular, we reference work factor when dealing with the computational feasibility of 'cracking' large quantities of passwords. Keep in mind that work factor here is a measure of the *work expended per successfully-cracked password*. Hint: Think about the total amount of work that needs to be done overall, and then how you can distribute this among passwords that are cracked successfully.

a) Blofeld has ordered his henchmen to find the passwords for all secret agents' personal email accounts, and they manage to gain access to an email account database that stores usernames and hashes. After analyzing the source code, they find that the passwords for these accounts are hashed without a salt.

Analyze the work factor associated with breaking this scheme. Assume that the attack is carried out by precomputing a table of hashes and associated passwords. Let:

- The size of the table be m
- The time required to compute the hash function be h
- The number of attempts to look up passwords in the table be t
- The proportion of these attempts which succeed be s $(0 \leq s \leq 1)$

Assume that:

- Lookups in this table take constant time
- h is constant, and does not vary with the size of the password hashed

Remember that work factor takes into account all work done, including the work expended precomputing the table.

i. Write a formula for the work factor for this approach.

ii. What asymptotic value does the work factor approach as the number of trials, t, increases?

b) Thrilled with gaining access to agents' emails, Blofeld and his henchmen now want to gain access to their MI6 email accounts, and they manage to gain access to the MI6 account database. However, after analyzing the source code, they find that these passwords are hashed with a per-user salt. Analyze the work factor associated with breaking this scheme. Assume that the attack is carried out by testing every password from a bank of m passwords - each password is tested by hashing it with the user's salt, and comparing it to the stored hash. Let:

- The time required to compute the hash function be h
- The number of password hashes which the attackers try cracking be t
- The proportion of these attempts which succeed be s $(0 \leq s \leq 1)$

You may assume that, on average, a hash is successfully cracked after half of the passwords in the list have been tried. However, an unsuccessfully cracked password will need to have tried all passwords in the list. Also assume that h is constant, and does not vary with the size of the password hashed.

i. Write a formula for the work factor for this approach.

ii. What asymptotic value does the work factor approach as the number of password hashes the attacker is trying to crack, t, increases?

Problem 3

Q needs your help in designing the criminal database system in an effort to identify and target criminals who belong to SPECTRE. To gain access to the database, an MI6 Agent would need to provide some sort of authentication. You have come up with two possible authentication systems:

a) Agents provide one password of length 9 (salted and hashed)

b) Agents provide two different passwords, each of length 8 (salted and hashed using two different salts)

Assuming passwords are alphanumeric (upper case and lower case) and chosen uniformly at random, which system is more secure against an attacker who is capable of hashing 100 million passwords per second? Justify your answer analytically.

Problem 4

In practice, when people choose 10-character alphanumeric passwords, they tend to provide much weaker security than when computers generate 60-bit cryptographic keys. There are 62 possible characters (26 lower-case letters, 26 upper-case letters, and 10 digits) in an alphanumeric password, so there are 62^{10} possible 10-character alphanumeric passwords. There are 2^{60} possible 60-bit cryptographic keys. 62^{10} is approximately equal to 2^{60}, so you might expect them to be equally secure. Why, in practice, is this not the case?

Pre-work

These are questions to which we do not expect you to know the answer. We don't expect you to know the details underlying the questions. As long as you state any (reasonable) assumptions you are making, and be explicit about what your understanding of the background material is, we will give you most of the credit so long as the reasoning based on that understanding is sound. Think of it like a math problem — show your work, and we won't take off too much for minor calculation mistakes.

Problem 5

A common way that web application developers check that user inputs are valid (for example, that they adhere to password length requirements, that emails are properly formatted, etc.) is with javascript executing in the browser. When a form is submitted, the javascript first checks to make sure that the inputs are valid, and refuses to submit the form if they are not. Explain why a web site cannot assume that the input it receives from users will be valid as a result of these checks.