Md. Safiqul Islam
Internet Security and Privacy
islam3@kth.se

1. Alice wants to use RSA to encrypt the message M=88 and send it to Bob. Bob has chosen two prime numbers (p=17 and q=11) to calculate the public number needed for the RSA keys.
Furthermore, Bob has selected the number e=7 to use in his public key.

a) What is the resulting public key published by Bob?

Answer:

The resulting public key is (e,n) = (7,187)

Where n=pq = 17 *11 = 187

b) What is the resulting ciphertext block C that Alice with send to Bob using RSA to encrypt
her message (M=88)?

Answer:

$C = M^e \bmod n = 88^7 \bmod 187 = 11$

c) What is Bob's private key?

Answer:

$de \bmod \phi(n) = 1$
so , $d * 7 \bmod 160 = 1$

So , d=23

Bob's Private key = (23,187)

d) Show how Bob can decrypt the ciphertext block C using his private key.

Answer:

$D = 11^{23} \bmod 187$

2. Password-guessing attacks, such as dictionary attacks, can be more or less effective, depending on how users pick their passwords and on how the attacks are launched.
a) Assume that an attacker is doing an on-line attack from a high-speed computer over a very
fast network, which means that the attacker has the capacity to generate guesses at high speed. What could be done in order to make such attacks more difficult to perform?

Answer:

By slowing down the attempt or adding delay between each attempt we can make such attempt difficult. We can allow the attackers for limited number of guesses per connection attempt. Also after a certain guesses we can lock down the account.

b) Adding "salt" to the UNIX password data base is an attempt to make password guessing
attacks less effective, in the cases where the attacker has access to the data base. Discuss what kind of attacks the salt can help against? Are there attacks where the salt does not help?

Answer:

The salt can help against Offline attacks. It makes impossible to perform single cryptographic hash operation and check the validity of password for group of users. But still it is not harder for guessing any one user's password.

c) The salt is only 12 bits. If we would make the salt larger (say 64 bits), would that make the
above attacks more difficult? Discuss what kinds of attacks this could help against, and what attack it might not help against?

Answer:

This will increase the complexity and difficulty for the attacker to calculate the hash operation. But still attacker can make guess to for any one user's password.

d) Organizations with strict security often enforce password policies in order to make password management more secure. What could such policies be? Give examples! Discuss in what ways strict password policies may actually make password management less secure.

Answers:

Organizations may enforce the user to use random string (like uppercase, lower case or number or special character) and also enforce the user to erase password after few days.
This may be insecure because user cannot be able to memorize this and he will store it somewhere or forget the password or write it to paper. This seems to be dangerous.

3. Nonces are frequently used in the Needham-Schroeder protocol.
a) What is the purpose of using nonces?

Answer:

The purpose of using nonce is to use it at once and make it unique. So, attacker will not be able to replay the same message.

b) Would it have been possible to use timestamps instead of nonces? In general, what is the drawback of timestamps, compared to nonces?

Answer:

Yes its possible to use timestamp. But the main drawback is here time synchronization. If it allows some time skew it would be possible for the attackers to impersonate using replay attacks. If time skew is in second granularity then the attacker will get time to do the replay attack. If time skew is in nanosecond granularity then it will be secure.

c) Explain the ticket invalidation problem in Needham-Schroeder.

Answer:

Attacker can steal Alice key and decrypt the message received from KDC. Then he will get the ticket by encrypting the message for Alice. If Alice manage to change his key but the ticket to Bob is still valid. So, the attacker can impersonate Alice to Bob with the key and communicate with Bob.

d) In the ticket invalidation problem, Trudy re-uses old information that she has picked up from previous communication between Alice and Bob. We have learned that timestamps and nonces are useful against replay attacks, so perhaps we could use

timestamps or nonces for the ticket invalidation problem as well. So assume that the ticket in Protocol 11-18 in Kaufman is defined as:
KBob{KAB, “Alice”, *R*},
where *R* is a nonce or a timestamp. Discuss if this could be useful for dealing with the ticket invalidation problem?

Answer:

If R is a timestamp then it will be more secure than nonce (random number or sequence number) because the attacker will be able to attack within a time skew. So, it is useful with dealing time invalidity problem.

But if R is nonce then it is not known to Bob. So, every time it will think that R is new and it will accept the ticket and ticket validity problem exists. But if this nonce is taken from Bob and use it then it would be secure (like Expanded Needham-Schroeder).

4. Kerberos is a protocol that is based around Needham-Schroeder.
a) Specify the Kerberos (V4) messages involved when a user wants to print a document on a “kerberised” print server.

Answer:

There are several steps:

1. **Users log into the workstation and request for service.**
2. **Authentication servers verify user access and generate session key and ticket granting ticket and results are encrypted using user’s master password password.**
3. **Workstation will send ticket and authenticator (that contains user’s name, network address and TGT).**
4. **TGS will decrypt that ticket and authenticator and verifies request and create ticket for requested service.**
5. **Workstation sends ticket and authenticator to server.**
6. **Server verifies and if match then it will grant access.**

b) The information in a TGT (Ticket Granting Ticket) is encrypted so the client cannot access the information in the TGT. However, all information in the ticket is already known to the client. Why is it still necessary to encrypt it?

Answer:
Yes it is necessary to prevent client adding any values to it or change the time period.