Date:
Page:

1,

i, Perfect forward secrecy is , if old long term keys become compromised the old session keys will remain secured.

Yes the variant to the diffie Hellman key exchange does provide perfect forward secrecy as the inclusion of the random number selected by user A acts as a one time use method to skew all other session keys as it is impossible to reverse the equation without knowing the random number. This essentially changes this protocol to Ephemeral diffie-Hellman, as even if a key is compromised in the future it cannot be used to decrypt all of the past messages.

ii, Explict = key confirmation + implicit key authentication

It would be explicit key authentication as user 'A' knows user 'B' must have a secret private key in order to create the session key, without it it wouldn't be possible. (key confirmation)

And 'A' will be ensured that only 'B' will be able to derive the session key as 'A' included the random number within the initial exchange of information to 'B'. (implicit)

Date:
Page:

2.

1) The hash chain based authentication protocol is susceptible to a replay attack as the user, server interaction in step 5 does not provide any timestamping or any other means of protection against the replay attack. A solution or countermeasure for this could be to include a timestamp (TS) within the data sent to the database from the user;

Thus $E(k_i; H_{i-1}, TS, N_u, i)\ N_u, i$

this may cause time synchronosation issues though.

2) This protocol does not have forward secrecy as, for it not to we need to show that the old session keys can become known. It is noted that "k_i = the ith session key that $k_i = H_i$;" thus if we assume k_2 was comprimised to find H_1 and $H_1 = k_1$ so k_1 is comprimised also thus its not forward secrecy.

To prevent this dont store $k_i = H_i$.

3,

User -> Server: (ID_c, B, p, g), pwd

Server -> Rserver: (ID_c, N_c)

Rserver -> User: (N_s, T_s)

User -> Rserver: $(h(Z, N_s, N_c, K))$

After reviewing the 2 factor authentication protocol we can see that it is using three way authentication, as the two parties, user and server, are using a third "way" / partie to assist in carrying out the authentication process, thus preventing a replay attack more effectively.

For the 2 Factor Authentication to be secure, atleast one factor must remain secure even if the other is comprimised.

In this case it is not secure if the smart card is comprimised. In step one of the user login process, the adversary may insert the compromised card into the card reader and a 6-digit shared pwd will be needed which is not known yet. Considering there is no locking mechanism if the password is entered incorrectly the adversary may brute force the $10^6 \rightarrow 1{,}000{,}000$ combinations of pwd which with modern computers can take far less than 1 second

After the adversary attempts to brute force the pwd twice they will need to wait a bit before attempting again as there is a lock preventing an incorrect pwd guess a third time.

In the mean time, the adversary may assume the presence of more IDs and supplementary info to forge a new smart card of another existing client and since these clients share the same pwd for the login a slow but guaranteed process of bruteforcing the pwd is possible.

Thus it is not completely secure if the adversary knows the smart card.

Date:
Page:

4)

i) BLP

	Readable objects	writable objects
A	O1, O3, O4	O1
B	O3, O4	O1
C	O3	
D	O3	O2, O4

ii) Biba

	Readable objects	Writable objects
A	O1	O1, O2, O4
B		O4
C	O1	O3, O4
D	O3, O2	O4