NSS Homework - 2

By Kaustav Vats (2016048)

How the Needham Schroeder protocol is vulnerable to a reflection attack. Can you describe a modification to the scheme which is not and DOES NOT rely on synchronize timestamps.

Needham-Schroeder protocol is used to protects against the reuse of the old ticket by using nonces. Nonce is a message that is used only once, it could be a random number, timestamp etc.

S.No	Alice	KDC	Bob
1.	To KDC: N1, Alice want to talk to Bob		
2.		To Alice: K_{Alice}(N1, $K_{Session}$, Bob, Ticket) & Ticket = K_{bob}($K_{Session}$, Alice)	
3.	To Bob: Ticket, $K_{session}$(N2)		
4.			To Alice: $K_{Session}$(N2-1, N3)
5.	To Bob: $K_{Session}$(N3-1)		

Note: Treat each row as an individual message

In the above table, we can see that N1, N2 and N3 are used as a nonce.

In message 1, Alice told KDC that she wants to talk to bob. Then KDC replies by encrypting with Alice public key, which only Alice can decrypt using her private key. This ensures that communication between Alice and KDC is encrypted and secure.

Then in message 3 Alice sends a challenge to Bob along with the ticket she received from KDC. Now Bob knows that Alice is talking to him. Bob knows because of presence of Alice string the ticket which is encrypted with Bob's public key, which only bob can decrypt using his own private key.

Now in message 4, Bob replies with the response of the challenge and a new Nonce N3 to challenge Alice.

In Message 5, Alice proves to Bob that she is the one who is talking to Bob.

Now the reflection attack is possible if all encryption are done using CBC. Since there is no chaining in CBC, Encrypted Nonce can be partitioned. Trudy(Attacker) can partition message 4. After extracting the partitioned encrypted nonce. Now Trudy can create a new session in which he can try to exploit the connection between Alice and Bob.

S.No	Alice	KDC	Bob
1.	To KDC: N1, Alice want to talk to Bob		
2.		To Alice: K_{Alice}(N1, $K_{Session}$, Bob, Ticket) & Ticket = K_{bob}($K_{Session}$, Alice)	
3.	To Bob: Ticket, $K_{session}$(N2)		
4.			To Alice: $K_{Session}$(N2-1, N3)
5.	To Bob: $K_{Session}$(N3-1)		
3'.	Trudy impersonating as Alice: Sending Msg to Bob: Ticket, $K_{Session}$(N2)		
4'.			To Alice which is intercepted by Trudy: $K_{Session}$(N2-1, N4)
3''.	Trudy impersonating as Alice: Sending Msg to Bob: Ticket, $K_{Session}$(N4)		
4''.			To Alice which is intercepted by Trudy: $K_{Session}$(N4-1, N5)
5'.	Trudy impersonating as Alice: Sending Msg to Bob: $K_{Session}$(N4-1)		

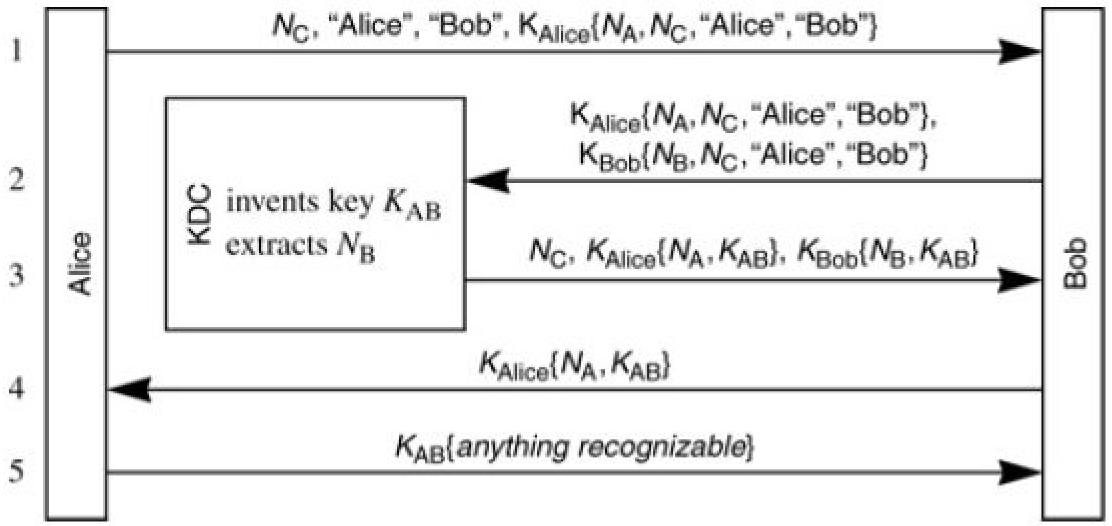
Reflection Attack
Trudy challenged Bob to with N4 then bob reply with answer and challenge for Trudy.
Now Trudy can use Encrypted answer for N4 nonce and verify itself as Alice.

Now to protect the protocol from above vulnerability.
We will proceed on an assumption that Alice Key is compromised.
One way is to use timestamp Nonce derived by Bob. So Alice will first request a Nonce from the Bob, which will be used by alice to get a ticket from KDC. KDC will add that Nonce of Bob. with the ticket. This will reassure bob that Alice is the one who is talking to him and has also talked to KDC before which generated above ticket. Once Alice changes her Key, Trudy won't be able to talk to kdc using Alice's Old key and any recorded message to KDC from Alice won't be useful.

S.No	Alice	KDC	Bob
1.	To Bob: I want to talk to you		
2.			To Alice: $K_{Bob}(N_{Bob})$
3.	To KDC: N1, Alice want to talk to Bob, $K_{Bob}(N_{Bob})$		
4.		To Alice: K_{Alice}(N1, $K_{Session}$, Bob, Ticket) & Ticket = $K_{bob}(K_{Session}$, Alice, $K_{Bob}(N_{Bob}))$	
5.	To Bob: Ticket, $K_{session}$(N2)		
6.			To Alice: $K_{Session}$(N2-1, N3)
7.	To Bob: $K_{Session}$(N3-1)		

Message 1, Alice requested a nonce from Bob. This ticket can be used only once.

Another way is Otways-Rees
Alice sends a message to Bob: N_C, “Alice”, “Bob”, $K_{Alice}(N_A, N_C$, “Alice”, “Bob”)
Then Bob forward this message to KDC and also encrypt N_B, N_C, “Alice”, “Bob”). Bob has attached his own Nonce and a common cleartext Nc.
Now KDC replies with Nc, K_{Alice}(Na, K_{AB}), K_{Bob}(Nb, K_{AB}).
Then Bob send part that is encrypted with Alice public key to Alice, which only she can decrypt.
Now they both can use common Key K_{AB}, Alice also proves her identity by sending an encrypted message which bob verifies and Bob doesn’t need to prove his identity.


1
N_C, "Alice", "Bob", $K_{Alice}\{N_A, N_C$, "Alice", "Bob"$\}$
2
$K_{Alice}\{N_A, N_C$, "Alice", "Bob"$\}$,
$K_{Bob}\{N_B, N_C$, "Alice", "Bob"$\}$
KDC
invents key K_{AB}
extracts N_B
3
N_C, $K_{Alice}\{N_A, K_{AB}\}$, $K_{Bob}\{N_B, K_{AB}\}$
Alice
Bob
4
$K_{Alice}\{N_A, K_{AB}\}$
5
$K_{AB}\{$*anything recognizable*$\}$

Above Image Ref: Network Security: Private Communication in a Public World, 2nd Ed Book