**Authentication Protocol**

This protocol is designed to provide authentication between parties in which only authentication is needed, with no need for confidentiality or the exchange of secrets. Supposed that two parties, Alice and Bob wish to authenticate each other. The protocol is as follows:

1. Alice begins by choosing a fresh nonce and sends the nonce and her name to Bob.

A -> B: na, A

1. Bob choses a fresh nonce of his own and sends his nonce, Alice's nonce, and Alice's name, encrypted with Bob's private key, to Alice.

B -> A: {nb, na, A}PrivK(B)

1. Alice decrypts the message using Bob's public key and checks that her name and nonce are in the message, authenticating Bob. Alice then chooses a new fresh nonce and sends that nonce, Bob's nonce, and Bob's name, encrypted with Alice's private key, to Bob.

A -> B: {na', nb, B}PrivK(A)

1. Bob decrypts the message with Alice's public key and verifies that his name and nonce are in the message, authenticating Alice.

**Assumptions made for the protocol**:

It is assumed that

1. Alice and Bob's private keys are uncompromised
2. Public keys of Alice and Bob are available to anyone.
3. Nonce used by Alice and Bob are freshly generated for each communication.

CPSA Model for Authentication:

1. The CPSA model of the authentication protocol will have two roles, Alice and Bob.
2. Each role will have three messages.
3. Alice will send two messages to bob and receive one message from bob.
4. Bob will receive two messages from alice and send one message to alice.
5. We will analyze shapes from both Alice’s and Bob’s point of view. So, we will use two skeletons, one for alice and one for bob.
6. Each skeleton will assume that the private key of alice and bob are non-originating.
7. Each skeleton will assume that the nonce of alice and bob are uniquely originating.

CPSA shapes generated:

Alice Point of view:

1. In the first shape, CPSA tries to achieve the authentication without needing bob at all. It fails at the second message because it can be realized only when it is encrypted with Bob’s private key.

A black and red dot on a white background

Description automatically generated

1. In the second shape, we have two strands, alice and bob. We have solid lines going from alice to bob and bob to alice respectively which means messages were sent without any modifications made to them. There is no third lien from alice to bob because in the last message alice sends a message back to bob but doesn’t receive a confirmation. So alice doesn’t know if the message was ever received by bob.

A screenshot of a computer

Description automatically generated

Bob’s point of view:

1. In the first shape generated for the skeleton of bob, cpsa tries to complete the communication without needing alice, but it fails on the third node because the message received by bob on the third node can only be sent by alice using her private key.

A screen shot of a computer

Description automatically generated

1. In the second shape, we have solid lines going from bob to alice for the second message and solid line for message received by bob from alice. We don’t have a solid line from alice to bob for the message because the message has identities of alice and a nonce generated by her. This message was sent in the open without any encryption, so technically this message could have been sent by anyone who knows alice’s identity.

A screen shot of a computer

Description automatically generated

1. In this shape, CPSA has found a way to satisfy the protocol with two responder strands. An intruder initiates the communication with a responder strand (Bob). Bob responds back to the intruder with the a nonce nb, and the original nonce na and identity a encrypted under private key of responder bob. The intruder decrypts the message using bob’s public key and extracts the nonce nb out of it.

The intruder uses this nonce nb to initiate another responder strand with alice. The intruder sends the nonce nb with the identity of bob to alice. Alice responds with a nonce na, and original message nb and b excrypted under the private key of alice. Intruder has intercepted this message and now responds back to bob, with a message that he expected from alice. This is one of the flaw of the protocol, where an intruder was able to authenticate himself as one of the legitimate users using two responder strands.

A screenshot of a computer

Description automatically generated