Ashni Shah CS568:
Applied Cryptography
April 22nd 2020

Needham-Schroeder Key Agreement Protocol

Secure computer communications depend on the encryption of messages transmitted in order to protect the confidentiality of a message. However, Professor Matthew Green, Johns Hopkins, states, “confidentiality xor authenticity is not possible. If you don't have both, often you don't have either.” If parties cannot guarantee the identity of the parties with whom they are communicating, they are susceptible to man-in-the-middle attacks. A man-in-the-middle attack is the most dangerous threat model in cryptography as the attacker can cause the most amount of damage by interfering with messages, yet still remain undetectable since both are under the impression they are communicating with each other. Man-in-the-middle attacks often rely on the malicious party impersonating a victim party in order to intercept and alter communications between two parties. Authentication is used to prove the identity of participants and ensure a client or server is who they claim to be. The use of authentication prevents misrepresentation of participants, removes the risk of transmitting confidential information to a fraudulent party and guarantees the integrity of a message.

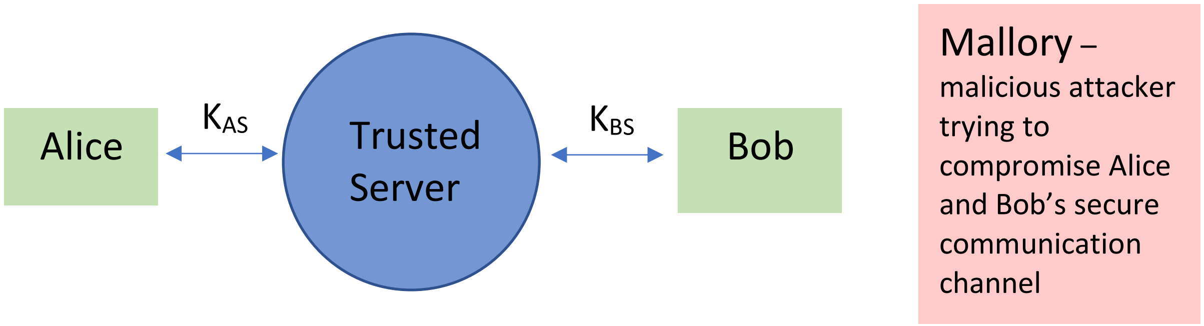
Problem Overview & Security Assumptions

The Needham-Schroeder key agreement protocol (1978) intends to address mutual authentication of large networks of computers, through the use of encryption[1]. Suppose Alice and Bob want to achieve mutual authentication so that they communicate securely. By reaching an agreement on a shared session key, which is only comprehensible to the parties, both parties can confirm and prove their identity, authenticating each other, and use the key for subsequent symmetrically encrypted communication.

The protocol requires the use of a trusted server, secure computing environment, and each party’s ability to encrypt and decrypt material. All communications are transmitted over unprotected networks and security assumptions include “an intruder can interpose a computer in all communication paths, and thus can alter or copy parts of messages, replay messages, or emit false material.”

The protocol supports (perfect) forward secrecy because the session keys are generated randomly and freshly by the trusted server and a comprised session key cannot help an attacker deduce previous session keys since they are independent[2]. On the other hand, the protocol does not support known-key security because a comprised old session key could allow an attacker to execute a replay attack. Subsequent fixes to the protocol have addressed this security flaw. In this report, following the Needham-Schroeder notation, encryption is indicated by braces subscripted with the key used.

Principals in the Protocol:


Alice
K_{AS}
Trusted Server
K_{BS}
Bob
Mallory – malicious attacker trying to compromise Alice and Bob's secure communication channel

Symmetric Protocol

The symmetric protocol establishes authentication using symmetric keys.

$A \rightarrow S: A, B, N_A$

Alice initiates the protocol by telling the trusted server S, her own identity, the identity of the desired correspondent (in this case Bob), and a nonce transaction identifier. The nonce identifier is randomly generated and never repeated to confirm that the message is fresh and not a replay from a previous exchange.

$S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB,} A\}K_{BS}\}K_{AS}$

The trusted server responds with an encrypted message, only comprehensible to Alice, confirming the nonce and desired correspondent, the session key K_{AB} and a message intended for and comprehensible to Bob.

$A \rightarrow B: \{K_{AB,} A\}K_{BS}$

Alice forwards the intended message to Bob

$B \rightarrow A: \{N_B\}K_{AB}$

Bob decrypts the session key using the symmetric server key K_{BS} and uses K_{AB} to encrypt its own independent nonce transaction identifier $N_{B.}$

$A \rightarrow B: \{N_B – 1\}K_{AB}$

Finally, Alice confirms receipt of the nonce by responding with $\{N_B – 1\}$ encrypted again with the session key, K_{AB}.

However, this protocol is still vulnerable to a replay attack if the attacker, Mallory, knows an old session key, K_M = old K_{AB}. Then, Mallory can impersonate Alice to Bob by sending the old session key to Bob (Figure 1). For Alice and Bob to make sure their communication is not interrupted; the following fix is proposed:

$A \rightarrow S: A, B, N_A, N_B$
$S \rightarrow A: \{K_{AB}, B, N_A\}K_{AS}, \{K_{AB}, A, N_B\}K_{BS}$
$A \rightarrow B: \{K_{AB}, A, N_B\}K_{BS}$
$B \rightarrow A: B, N_B$

Now, Alice can confirm the nonce identifier N_A and her intended correspondent and Bob can also confirm the message origin, learn the session key and ask Alice to confirm the second nonce identifier N_B, which only she and the trusted server would know. Even if an attacker knew an old session key, the use of 2 unrepeated nonces prevent the attacker from impersonating any party without their knowledge.

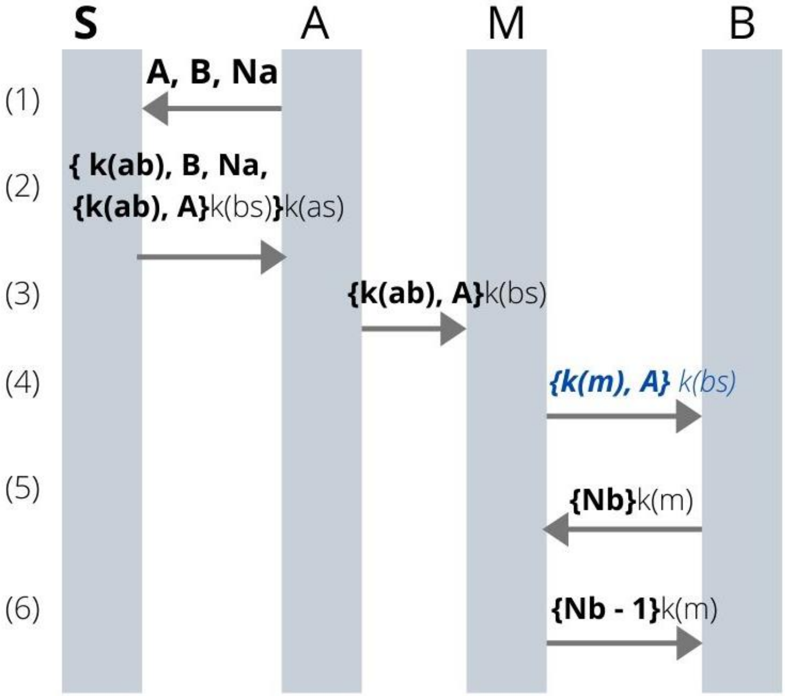


Figure 1
Symmetric Protocol Replay Attack

Public Key Protocol

The Needham-Schroeder public key protocol (Figure 2) is similar to the symmetric key protocol. In this case each party and trusted server has a public key (pk) and corresponding secret key (sk), which are the inverse of each other. It is important to note that double encryption of the message with the secret key of the sender then the public key of the recipient is necessary to prevent an attacker from injecting messages, since the public keys are not secret and accessible by any party. Double encryption provides a way of confirming the message's recipient as well as its origin.

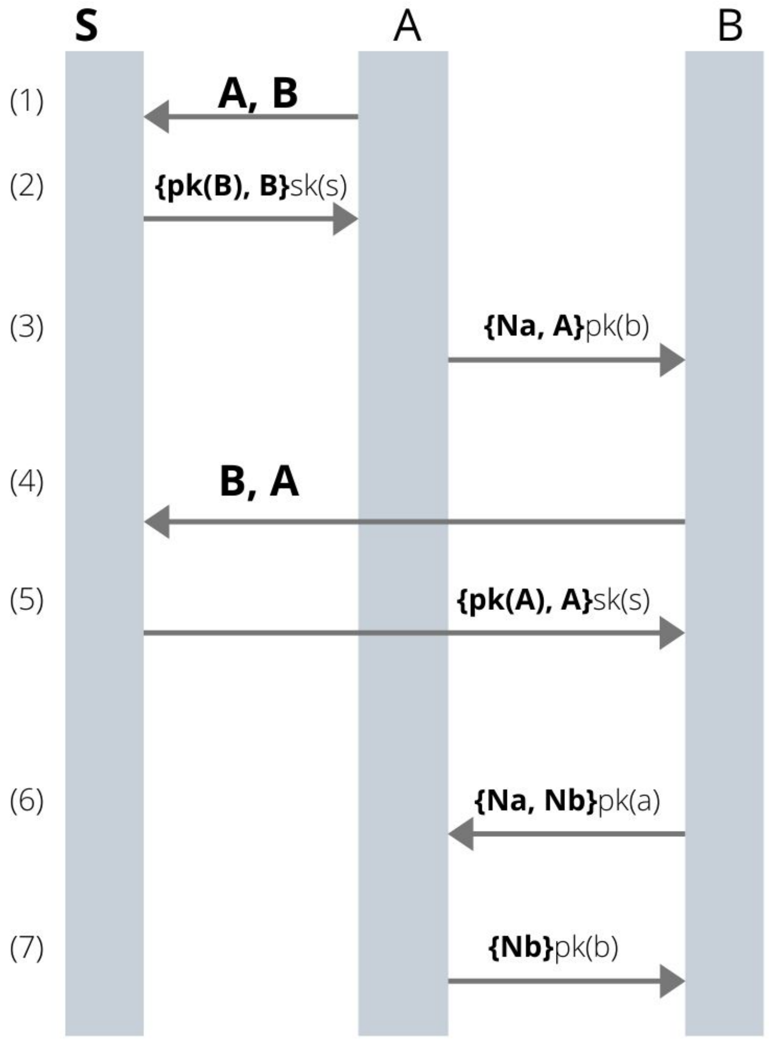


Figure 2
Needham-Schroeder Public Key Protocol

(1) Alice initiates the protocol by telling the trusted server S, her own identity, the identity of the desired correspondent, Bob.

(2) Server responds with the public key of Bob, encrypted with the server's secret key. Here, encryption is required to ensure integrity not confidentiality of the message.

(3) Alice sends a nonce transaction identifier and her own identity to Bob, encrypted with his public key. This ensures the message can only be decrypted by Bob.

(4) Bob decrypts the message and requests the public key of Alice from the server.

(5) Server responds with the public key of Alice, again encrypted with the server's secret key for integrity purposes.

(6) Bob sends Alice the nonce identifier to confirm receipt of her message, and his own identifier N_B, encrypted with Alice's public key.

(7) Alice responds with confirmation of Bob nonce identifier in a message only comprehensible to him by decryption using his secret key sk(b).

In 1995, Lowes came up with an attack for the public-key protocol[3]. If the attacker, Mallory, can get Alice to initiate a second session communicating with the attacker, the attacker can impersonate Alice to Bob (Figure 3).

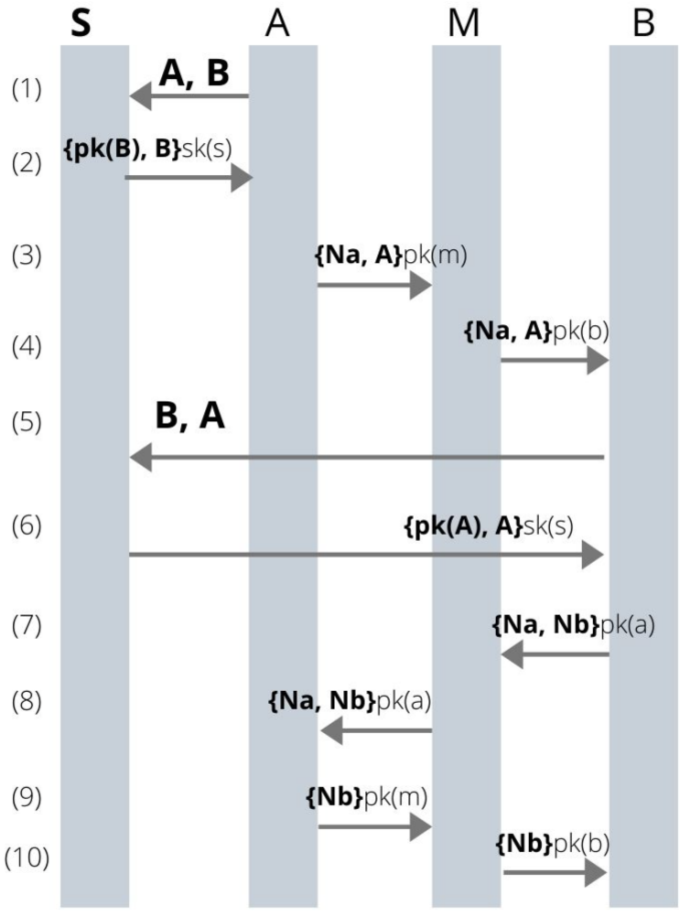


Figure 3
Lowes' Public Key Protocol Attack

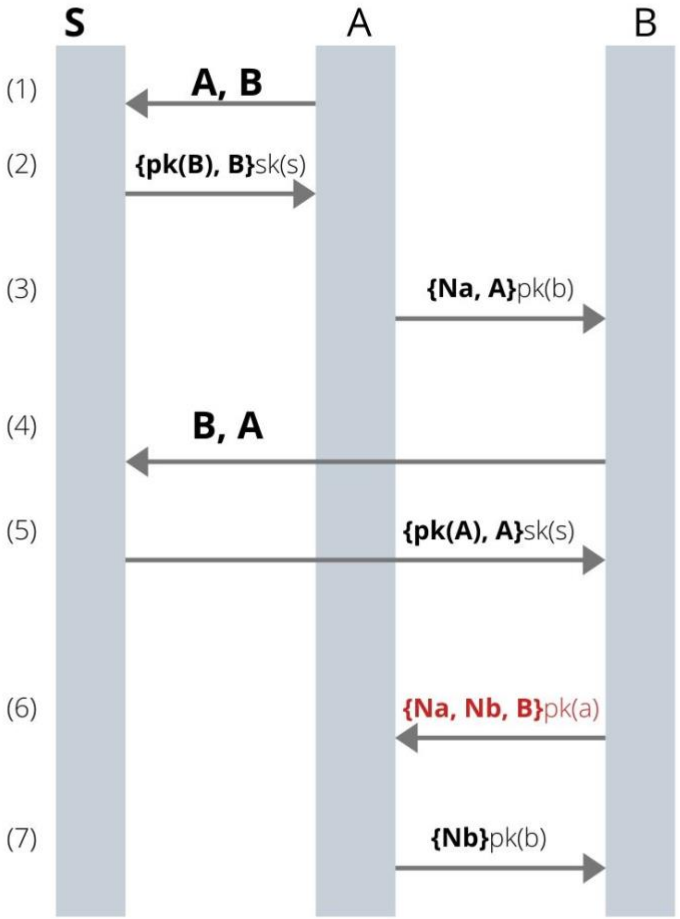


Figure 4
Lowes' proposed Public Key Protocol Fix

Lowes proposed a fix for this insecurity (Figure 4), in which message (6) is altered to include the sender's identity - Bob. This prevents an attacker from impersonating Alice to Bob because Alice will know the message came from Bob, not Mallory with whom she is communicating. Alice then send the confirmation nonce identifier from Bob to Bob or alert Bob that their communication is being interfered with.

To conclude, authentication is a key property necessary for secure communication. the Needham-Schroeder protocol is an effective method for parties to mutually authenticate each other, with the help of a trusted server. Although the protocol has its flaws and security vulnerabilities, subsequent fixes have addressed these issues allowing for the development of influenced protocols. It is recognized as one of the first proposed key exchange protocols and laid groundwork for what assumptions and goals were necessary from a key exchange protocol, even influencing the construction of the Kerberos protocol[4].

Code Description

Implementation of Code can be found at:

https://github.com/sashni/Needham-Schroeder-Protocol

I have implemented a secure communication network in Java using the Needham-Schroeder symmetric key agreement protocol[5]. The Needham-Schroeder protocol is used to establish a shared session key, used in encrypting following messages. The shared session key allows for mutual authentication between two parties: Alice and Bob, with the help of a trusted server, S. By running either the server.java or client.java file, a user is able to connect and transmit messages over an unprotected network, implemented using Sockets in Java. The client then declares whether it is Alice or Bob and the protocol is initiated, running through a simulation of the protocol with each party resulting in the shared session key. This session key is then used to encrypt further following messages until at least one of the parties disconnects from the network. The next time the parties reconnect to communicate securely, fresh and random nonce identifiers are used and new encryption keys are generated in order to prevent against replay attacks. Instead of using a DES symmetric key to encrypt transmitted messages during the protocol (e.g. K_{AS}, K_{BS}), I chose to use a key generated using AES mode with CBC (Cipher Block Chaining) and a randomly generated initialization vector, 16 bytes, because of its better security guarantees in encrypting messages with a block size of 128 bits[6]. The shared session key is also generated using the same AES mode, allowing clients to easily encrypt the subsequent messages.

Instead of following the protocol precisely, I chose to implement the proposed fix to prevent replay attacks, see top of page 3. The fixed protocol incorporates sending both nonce identifiers encrypted to their corresponding parties (Alice or Bob) in order to ensure the messages are new and aren't being reused by an attacker in a replay attack.

Further work that could be added to the implementation is to implement a hybrid protocol combining the symmetric key and public key protocol for client-server and client-client communications respectively.

Citations

[1] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), December 1978.

[2] A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. *CRC Press*, 1996. http://cacr.uwaterloo.ca/hac/about/chap12.pdf

[3] Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131--136, November 1995.

[4] Jason Garman. Kerberos: The Definitive Guide, The Needham-Schroeder Protocol (Chapter 3) https://www.oreilly.com/library/view/kerberos-the-definitive/0596004036/ch03s01.html

[5] Mandeep Kumar, Alok Tuli, Ruby Tuli. Secure Communication using Needham-Schroeder protocol. *CPMR-IJT, Volume 1, No. 1,* December 2011. https://pdfs.semanticscholar.org/ad0c/75d01aadd8137e45b6ac9fcb6f7fb4a596d7.pdf

[6] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani. New Comparative Study Between DES, 3DES and AES within Nine Factors. *Journal of Computing, Volume 2, Issue 3.* March 2010.