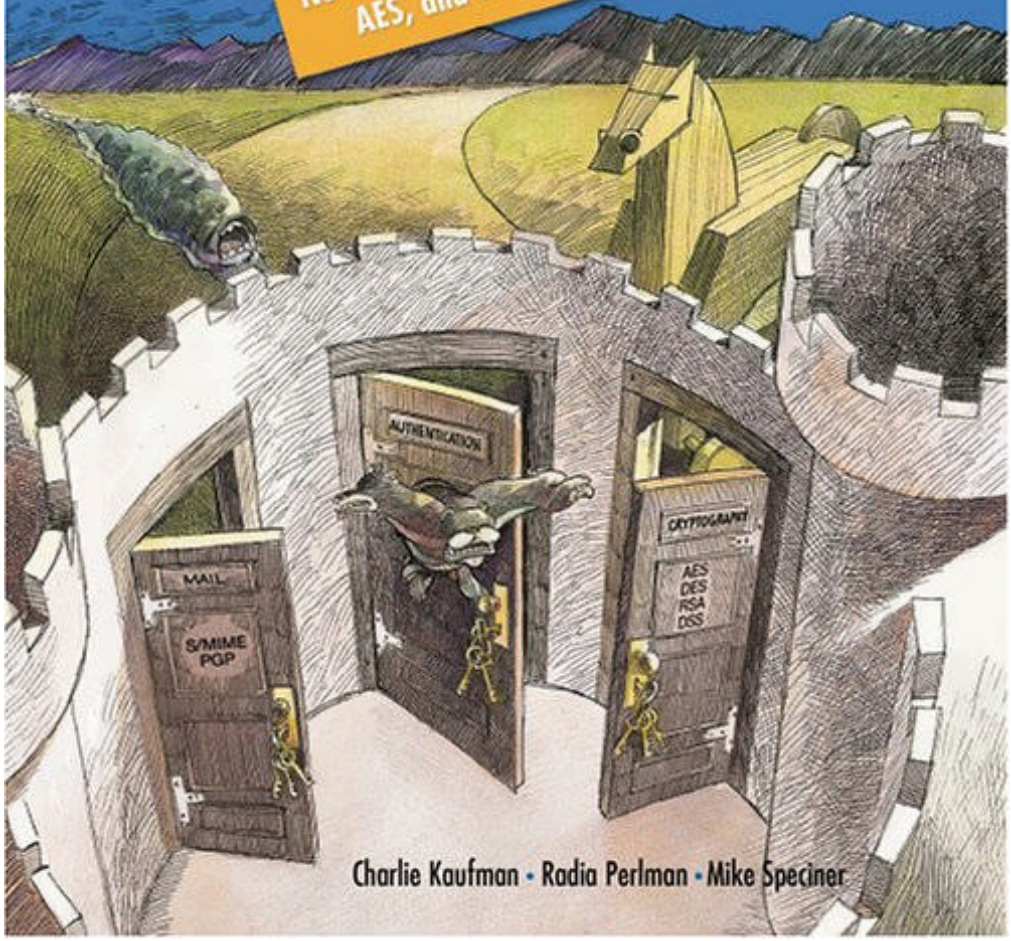
PRENTICE HALL SERIES IN COMPUTER NETWORKING AND DISTRIBUTED SYSTEMS

SECOND EDITION

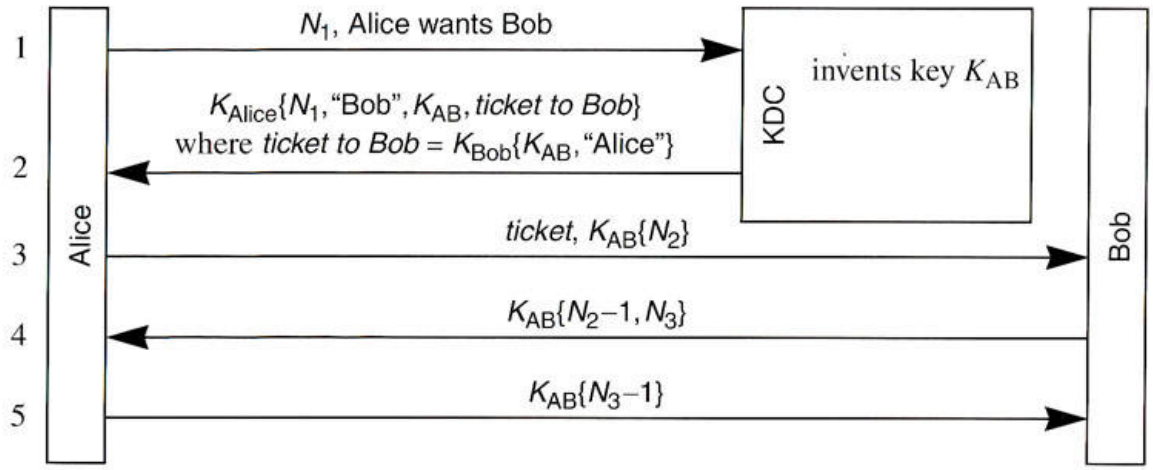
NETWORK SECURITY

PRIVATE *Communication in a* PUBLIC *World*

Now includes IPsec, SSL, PKI, AES, and Web security



Charlie Kaufman • Radia Perlman • Mike Speciner



Protocol 11-18. Needham-Schroeder

Message 1 tells the KDC that Alice wants to talk to Bob. The purpose of the nonce N_1 is to assure Alice that she is really talking to the KDC. The (admittedly far-fetched) threat the protocol is trying to avoid is where Trudy has stolen an old key of Bob's, and stolen the message where Alice had previously requested a key for Bob. Bob, realizing Trudy has stolen his key, has changed his key. Trudy waits around until Alice makes a request to the KDC to talk to Bob. Trudy then replays the stolen message, which looks like an ordinary reply from the KDC. Then Trudy successfully impersonates Bob to Alice, because Trudy knows the key that Alice thinks the KDC just created for Alice and Bob. As we've already said, it is hard to imagine a circumstance where this would be a practical threat, but adding the nonce costs virtually nothing, and it removes the need to think about whether the threat might be practical in any particular circumstance.

In message 2, the KDC securely (i.e. encrypted and integrity-protected) gives Alice the key K_{AB} it has generated for Alice and Bob to share. It puts in the string "Bob" to make it impossible for Trudy to tamper with Alice's request, substituting the string "Trudy" for "Bob" and then being able to trick Alice into talking to Trudy and thinking that Trudy is Bob. If Trudy were to tamper with message 1 by substituting her name for Bob's, then Alice will discover, in message 2, that the KDC has just given her a key to communicate with Trudy, not Bob. In message 2, along with the encrypted key K_{AB} and Bob's name, the KDC also gives Alice a ticket to Bob. The ticket to Bob consists of the key K_{AB} and Alice's name, encrypted with Bob's key. To Alice, the ticket will just be a pile of unintelligible bits. The Needham-Schroeder protocol has been criticized for doubly encrypting the ticket. The ticket is sent encrypted with Alice's key, though it would have been just as secure to send it in the clear.

In message 3, Alice sends a challenge (N_2) to Bob, encrypted with K_{AB}, along with the ticket. Bob decrypts the ticket to find the shared key K_{AB}. He uses K_{AB} to extract N_2. Alice knows that only someone who knows Bob's key can decrypt the ticket and thereby discover the shared key K_{AB}. In message 4, Bob proves he knows K_{AB}, since he was able to find N_2. Likewise, Bob

assumes that it must be Alice if she knows K_{AB} because the KDC puts Alice's name in the ticket with K_{AB}. In message 4, in addition to sending back N_2 (actually a decremented version of N_2), he includes a challenge N_3 encrypted with K_{AB}, and Alice sends back (in message 5) a message proving she knows K_{AB}.

There is an interesting reflection attack to which the protocol would be vulnerable if the encryption in message 4 were done using, say, DES in ECB mode. Suppose each of the nonces were 64 bits long. Because of the nature of ECB, N_2-1 and N_3 would each be separately encrypted. Suppose Trudy wants to impersonate Alice to Bob. First she eavesdrops on an authentication handshake where Alice talks to Bob. So Trudy sees messages 3 and 4. Later, she replays message 3 to Bob. Bob will respond with $K_{AB}\{N_2-1,N_4\}$ (where N_4 is a nonce different from the one Bob chose last time he talked to Alice). Trudy can't return $K_{AB}\{N_4-1\}$, but she can open a new connection to Bob, this time splicing in $K_{AB}\{N_4\}$ instead of $K_{AB}\{N_2\}$. Bob will return $K_{AB}\{N_4-1,N_5\}$. Then Trudy can take the first encrypted block, which will be $K_{AB}\{N_4-1\}$ and return that as message 5 of her first connection. In fact she will never have found out what N_4 was, but she didn't need to know N_4. She just needed to know $K_{AB}\{N_4-1\}$.

If encryption were done in CBC mode instead of ECB mode, Trudy would not be able to accomplish this, because she couldn't splice pieces of messages. But if encryption were done in CBC mode, there would be no reason to have Bob and Alice decrement each other's nonces. Message 4 could just as securely be $K_{AB}\{N_2,N_3\}$, and message 5 could just as securely be $K_{AB}\{N_3\}$.

The lesson to be learned is that if pieces of a message must, for security reasons, be sent together, then the entire message must be integrity-protected so that parts of it cannot be modified by an attacker.

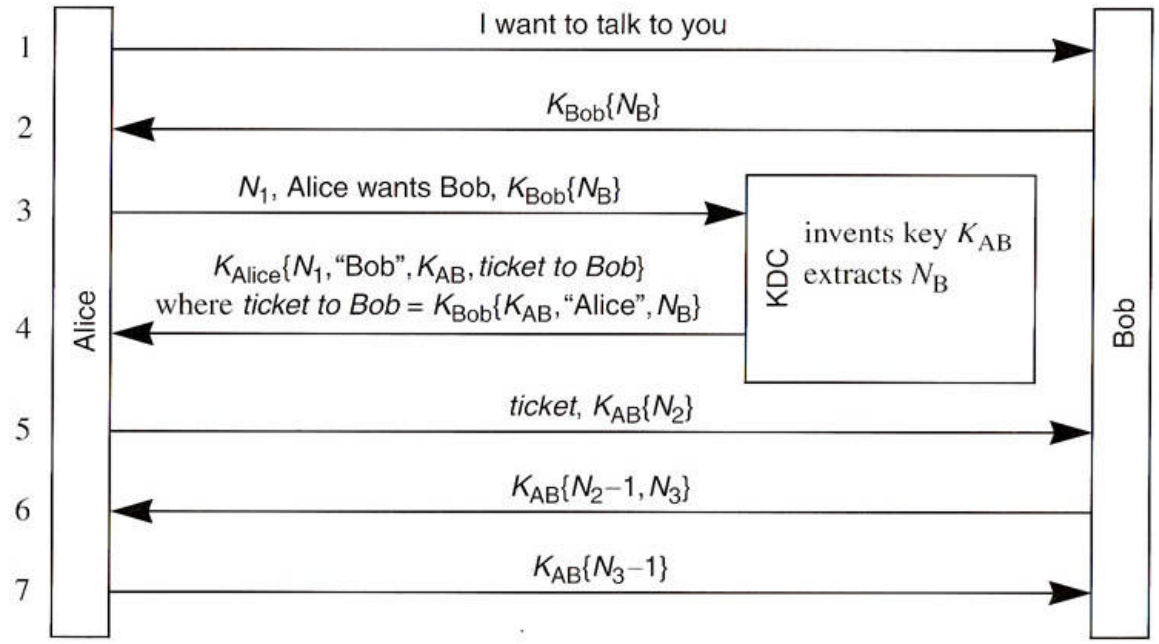
11.4.2 Expanded Needham-Schroeder

There is a remaining security vulnerability with Protocol 11-18. Suppose Trudy finds out Alice's key. Trudy can of course, at that point, claim to be Alice and get the KDC to give Trudy (which the KDC thinks is Alice), a shared key to Bob and a ticket to Bob. There's really nothing we can do about Trudy impersonating Alice if Trudy steals Alice's key. But we'd like it to be possible to prevent Trudy from doing any more damage once Alice changes her key. The problem is, with the protocol we've just described, the ticket to Bob stays valid even if Alice changes her key.

The vulnerability can occur even if Trudy only manages to capture a previous key used by Alice, say J_{Alice}, and not the key Alice is using now. Suppose that when Alice was using J_{Alice}, Trudy had overheard (and recorded) Alice asking the KDC for a ticket for Bob. At that time, the KDC would have generated a shared key J_{AB}, and Trudy would have seen $J_{Alice}\{N_1,\text{"Bob"},J_{AB},K_{Bob}\{J_{AB},\text{"Alice"}\}\}$. At the time, Trudy could not decrypt the message, but we'll assume she's stored it away, hoping to capture an old key of Alice's. Once Trudy does discover J_{Alice}, she can interpret the messages she's recorded, discover J_{AB}, and, using the ticket

$K_{Bob}\{J_{AB}, \text{"Alice"}\}$, she can convince Bob that Alice will be talking to him using J_{AB}. The fact that the KDC knew Alice's key changed is irrelevant; the KDC doesn't participate in this new authentication.

A paper [DENN81] pointed out this weakness, and in [NEED87], a suggested fix was proposed by adding two additional messages. Alice requests a nonce N_B from Bob, which Alice will send to the KDC, and the KDC will package N_B in the ticket to Bob. This will reassure Bob that whoever is talking to him has talked to the KDC since Bob generated N_B. Once Alice changes her key, Trudy will not be able to talk to the KDC using Alice's old key, and any recorded messages from the KDC using an old key of Alice's will also not be useful. The proposed protocol added 2 messages, and therefore uses 7 messages. The protocol can, however, be reduced to 6 messages (see Homework Problem 9).

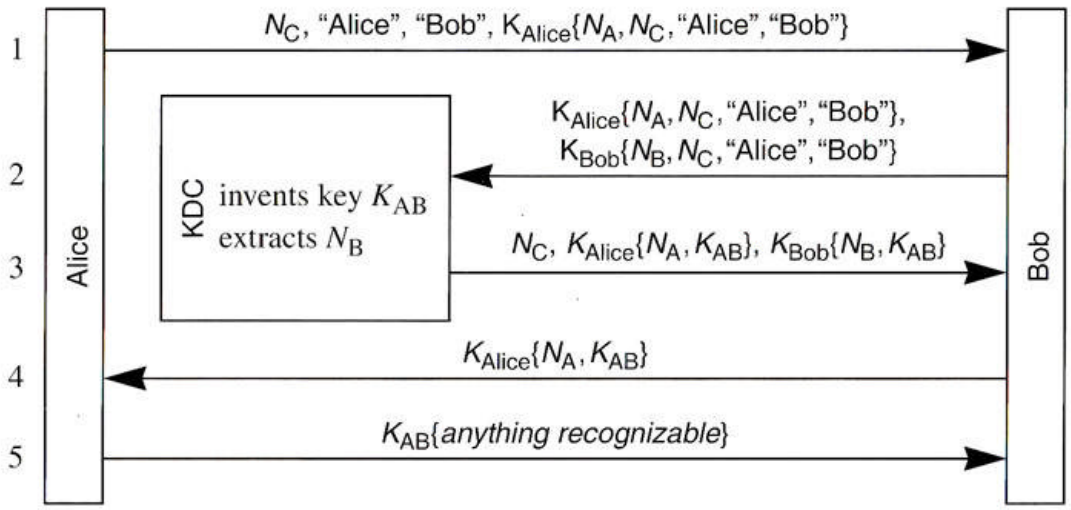


Protocol 11-19. Expanded Needham-Schroeder

11.4.3 Otway-Rees

In [OTWA87], an improved authentication protocol is given. It also solves the ticket invalidation problem, and it does mutual authentication in 5 messages.

Otway and Rees mention that they were inspired by a remark from Needham that *the suspicious party should always generate a challenge*. Let's look at their protocol (Protocol 11-20). Alice generates two nonces. One, N_C, is sent in the clear to Bob. In addition N_C is included (along with



Protocol 11-20. Otway-Rees

the other nonce, N_A) in a message encrypted by Alice that Bob cannot interpret. It just looks like a pile of bits, and he will simply forward $K_{Alice}\{N_A, N_C, \text{"Alice"}, \text{"Bob"}\}$ to the KDC.

In message 2, Bob forwards the encrypted message Alice sent, along with a message Bob encrypts which includes his own nonce N_B, the nonce N_C that Alice sent in the clear, "Alice", and "Bob", to the KDC. The KDC checks to make sure that the common nonce N_C is the same in both encrypted messages. If not, the KDC will reject the message. The fact that they are the same proves that Bob is really Bob, since only someone knowing K_{Bob} can encrypt N_C inside a message.

In message 3, the KDC gives Bob a message to forward to Alice. That message, when forwarded to Alice, reassures Alice that both the KDC and Bob are legitimate. (Alice knows the KDC is legitimate because it encrypts N_A with key K_{Alice}. Alice knows Bob is legitimate because the KDC would not have continued the protocol if it hadn't already verified that Bob was legit.) The KDC also tells Bob the common key, and reassures Bob that it is really the KDC by putting Bob's nonce N_B inside the message to Bob.

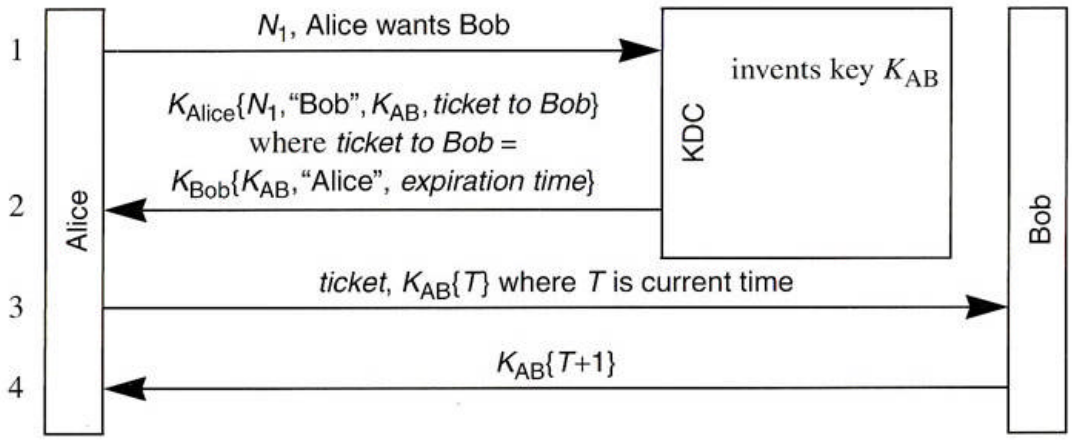
Message 4 is merely having Bob forward $K_{Alice}\{N_A, K_{AB}\}$ to Alice. In message 5, Alice proves her identity to Bob by showing that she knows K_{AB}. Bob does not need to prove his identity explicitly to Alice because Alice will assume that the KDC authenticated Bob.

In fact, this protocol could be simplified by getting rid of one of Alice's nonces (see Homework Problem 4).

If we want to be really paranoid, then for subtle reasons it is necessary in the Otway-Rees protocol that N_C be not just a nonce but also unpredictable. Otherwise Trudy, in a rather involved scenario, could impersonate Bob to Alice. This is the way it would work: Suppose Alice is using a sequence number for N_C. Trudy watches and sees that Alice is currently using, say, 007 for N_C. So Trudy sends a message to Bob claiming to be Alice. She sends 008, "Alice", "Bob", *garbage*. Bob can't tell that the fourth element in the message is garbage. He forwards *garbage* on, along with his

own message $K_{Bob}\{N_B, 008, \text{"Alice"}, \text{"Bob"}\}$. Trudy records that message. The KDC will reject Bob's message, since the garbage won't decrypt properly. Then Trudy waits until Alice generates her next request to talk to Bob. That will be 008, "Alice", "Bob", $K_{Alice}\{N_A, 008, \text{"Alice"}, \text{"Bob"}\}$. Trudy forwards that, along with the message $K_{Bob}\{N_B, 008, \text{"Alice"}, \text{"Bob"}\}$ she recorded from Bob, to the KDC. The KDC will accept the messages now since both will contain 008. The only authentication of Bob in the Otway-Rees protocol is done by the KDC verifying that the nonce N_C is the same in the message encrypted with K_{Bob} and the message encrypted with K_{Alice}. Trudy can complete the exchange by forwarding to Alice the message the KDC sends her for Alice. Alice will wrongly assume the party to whom she is talking is Bob. If Alice and Bob are planning on using the shared key, say for encrypting the messages to one another, then Trudy won't have succeeded. But if all Alice and Bob want to do is authenticate, then Trudy will have successfully impersonated Bob to Alice.

The Kerberos authentication service (see Chapter 13 *Kerberos V4*) is roughly based on the Needham-Schroeder protocol. It looks a lot simpler than these protocols because it assumes a universal idea of time, and includes expiration dates in messages. The basic Kerberos protocol is:



Protocol 11-21. Kerberos

11.5 NONCE TYPES

A **nonce** is a quantity which any given user of a protocol uses only once. Many protocols use nonces, and there are different types of nonces with different sorts of properties. It is possible to introduce security weaknesses by using a nonce with the wrong properties. Various forms of nonce are a timestamp, a large random number, or a sequence number. What's different about these quantities? A large random number tends to make the best nonce, because it cannot be guessed or predicted (as