Solutions to Selected Problems

Guide to Internet Cryptography

Companion Material

February 17, 2026

Preface

This document provides solutions to selected problems from the book *Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications*. The material is intended for educational use in courses and self-study.

Book website: https://link.springer.com/book/10.1007/978-3-031-19439-9

1 Chapter 14: Kerberos

Problem 14.1 Symmetric key management

How many symmetric keys do you need to secure communication between 1,000 IoT devices in case there is no TTP and no public key crypto available?

Solution

Each pair of devices needs a distinct, secret, symmetric key. There are $1,000 \cdot 999 = 999,000$ such pairs. On each device, a different set of 999 keys must be installed.

Problem 14.2 Kerberos communication pattern

Which of the three communication patterns does Kerberos use? Why is this pattern best suited in a client-server scenario?

Solution

As mentioned in Figure 14.2, Kerberos uses communication patter (a). This pattern is best suited for client-server scenarios because C may always act as a client, and TTP and S may always act as servers.

In pattern (b), S must act as both client and server. In pattern (c), TTP must act as both client and server.

Problem 14.3 Needham-Schroeder protocol

(a) The nonce n_C chosen by the client guarantees the “freshness” of the session key k_{CS}. Why is this mechanism not used by the server, i.e., why is no server nonce n_S used in the protocol?

(b) Describe a replay attack on the server if the optional challenge-and-response protocol is omitted.

Solution

(a) Because the server would have to create *and communicate* this nonce to the client. In a client-server scenario, this would require the client to send a request to the server, and the server to answer this request, resulting in 1 RTT added in protocol delay.

(b) Simply sending c_2 to the server is enough to convinve the server S that client C wants to contact it. Whether this results in a 'real' attack depends on the use of k_{cs}: If this key or a derivate of it is used to encrypt or MAC the communication between client and server, this communication will eventually aborted. If the key is only used to authenticate C, but later another means of protection is used (e.g., TLS), this replay attack could be successful.

Addition: Any adversary who manages to decrypt c_1 may use c_2 as a means to authenticate to the server S as C, as long as the key k_S is valid. This attack does not require any interaction with the TTP.

Problem 14.4 3-Party Kerberos

What is the purpose of the timestamps t_{KAS} and t_C? Why is there no optional challenge-and-response protocol anymore?

Solution

The timestamp t_{KAS} is used to limit the lifetime of a Kerberos 'ticket', i.e., a cryptographic 'Bearer token' with which the client (or any adversary in possession of this token) may authenticate to the server. The timestamp t_C acts as a (predictable) challenge in a challenge-and-response protocol.

Problem 14.5 4-Party Kerberos

The two instances of $3Kerberos$ used in the 4-party case differ slightly in the number of ciphertexts exchanged. The first message in the first instance does not contain a ciphertext, whereas the first message in the second instance contains two ciphertexts. Why?

Solution

The first message in the first instance contains

$$id_C, id_{TGS}, n_1$$

The first message in the second instance contains

$$TGT, Enc(id_C), id_S, n_3$$

So the structure is very similar, except that the identity of the client is now encrypted. The second ciphertext TGT results from the double role of this message: It is simultaneously the 3rd message in the first exchange, and the 1st message in the second exchange.

Problem 14.6 Kerberos Security

(a) Can you give an example of a malleable encryption scheme?

(b) How can an attacker generate "Golden Tickets" if she knows the key $k_{KAS,TGS}$?

Solution

(a) All (unauthenticated) stream ciphers, and AES-CBC for the first block.

(b) For any triple $(k_{C,TGS}, ts_{KAS}, id_C)$ chosen by the adversary, a valid ticket granting ticket can be created by an adversary who knws $k_{KAS,TGS}$. All other values sent to TGS can be created knowing these values, and any message received from the TGS can be decrypted with them. Thus such an adversary can impersonate any client id_C, and can access any server on which this client has access rights.

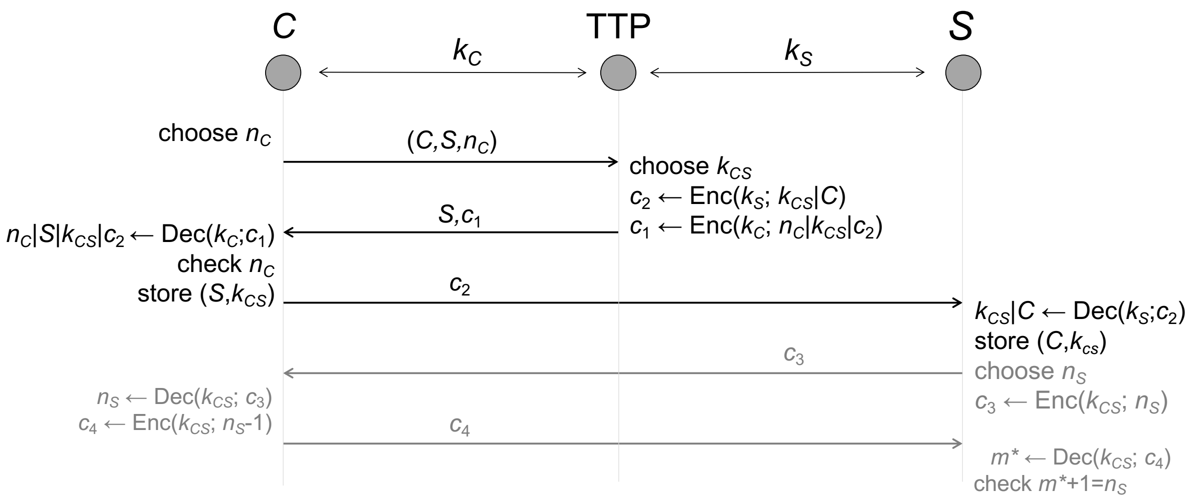

Figure 1: Needham-Schroeder Protocol

Problem 14.7 Insecurity of a Needham-Schroeder variant

Figure 1 shows a slight variant of the Needham-Schroeder protocol. The only difference to the original protocol is the omission of the server identity from c_1. How can a MitM attacker, who runs another server S' attached to the TTP, impersonate a server S?

Solution

The MITM adervsary proceeds as follows:

1. In the first message, he exchanges S with S'.
2. In the second message, he cahnges S' back to S.
3. He redirects the third message to S'. S' can decrypt message c_2, since the TTP used $k_{S'}$ to encrypt it. The adversary now knows $k_{CS'}$ and uses it to succeed in the optional challenge-and-response protocol.

This attack works because the MITM adversary has full control over the identity of the server in this modified protocol.