Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Computer Networking and IT Security (INHN0012)

Tutorial 13

Problem 1 DNS Zonefile

You are a student assistant at the chair and have been given the task of creating a zone file for `grnvs.net.`.
Relevant servers and their domain names and IP addresses are shown in Figure 1.1.
Fill in the following template so that the requirements of the individual subtasks are fulfilled. Enter the letter of the corresponding subtask in the box at the end of each line.

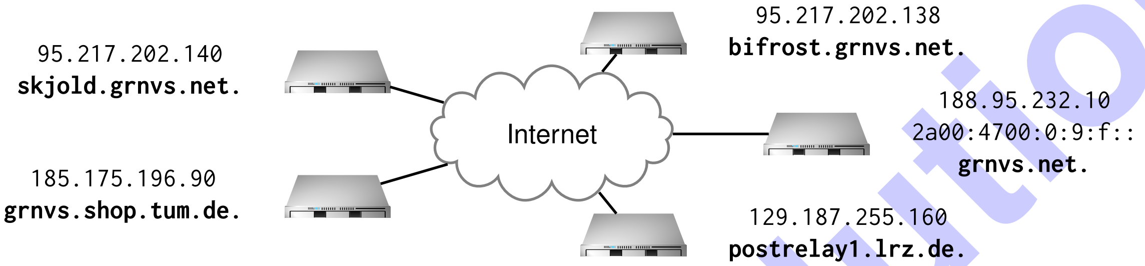


Figure 1.1: Servers relevant for the zone `grnvs.net.` with domain names and IP addresses

```
grnvs.net.          600   IN     SOA      bifrost.grnvs.net.      moepi.grnvs.net.
                                          164214 3600 900 604800 900

grnvs.net.                       NS       bifrost.grnvs.net.                        a)

grnvs.net.                       NS       skjold.grnvs.net.                         a)

bifrost.grnvs.net.               A        95.217.202.138                            b)

skjold.grnvs.net.                A        95.217.202.140                            b)


grnvs.net.                       A        188.95.232.10                             c)

grnvs.net.                       AAAA     2a00:4700:0:9:f::                         c)

grnvs.net.                       MX       10 postrelay1.lrz.de.                     d)

shop.grnvs.net.                  CNAME    grnvs.shop.tum.de.                        e)
```

a)* The primary name server of the zone is `bifrost.grnvs.net` and is already entered. For fault tolerance, the server reachable under `skjold.grnvs.net` shall act as the secondary name server.

b)* Ensure that the name servers are reachable under their respective IP addresses.

c)* When accessing `grnvs.net` in the browser, the GRNVS website should be displayed. This currently only works for IPv4; it should also be reachable via IPv6.

d)* In order for students to send an email to `info@grnvs.net`, a mail server must be configured. To avoid operating one yourself, the LRZ email service Postrelay shall be used with priority 10.

e)* In the next semester, merchandise and bonus points will be sold in a webshop. To avoid operating an online shop, a cooperation with the TUM Shop has been arranged. For access, `shop.grnvs.net` shall serve as an alias for `grnvs.shop.tum.de`.

f)* Why is it not possible to configure `grnvs.net.in.tum.de.` as an alias for `grnvs.net.` in this zone file?
The domain `grnvs.net.in.tum.de.` is located in the zone `net.in.tum.de.` and not in `grnvs.net.`.

Problem 2 Alice in Cryptoland

Alice and Bob want to securely communicate over an insecure channel. Therefore, they want to define a means of message transportation that provides authenticity, integrity, and confidentiality.
Alice has come up with a scheme for a protocol. It is a hybrid scheme: both parties posess a keypair which is used to exchange a shared key. This key is then used for symmetric encryption after the initial exchange is completed. Since Alice and Bob don't know each others public keys yet, the protocol includes them in the initial handshake.
To initiate the communication, the requesting party sends a message request which contains their public key. The requestee responds with the freshly generated secret k as well as their public key. The message containing k is encrypted with the requesting parties public key that was received in the first message. The exchange is completed by an acknowledge by the requesting party, encrypted with the requestee's public key.

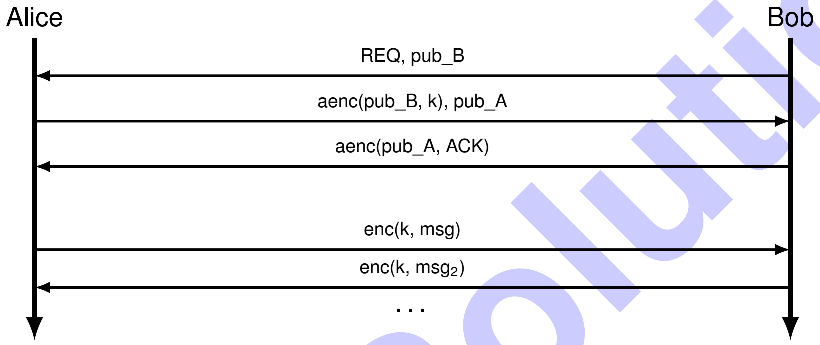
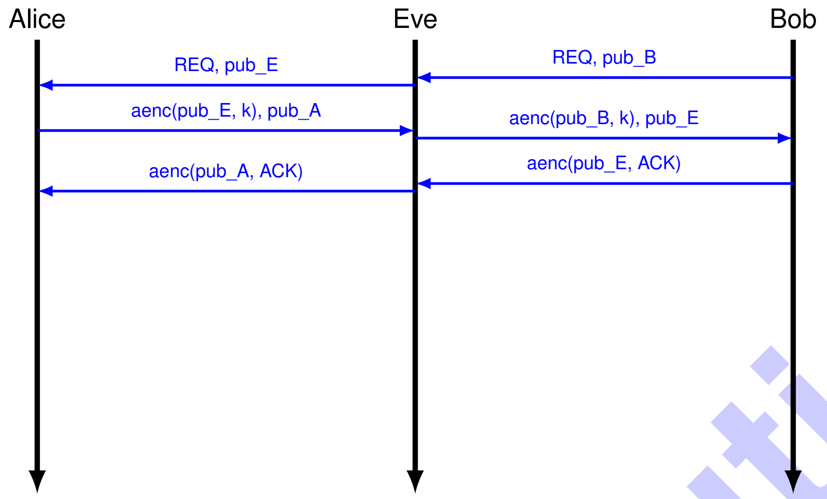


Figure 2.1: The communication protocol proposed by Alice.

For this task, assume an active man-in-the-middle attacker Eve, who can see and tamper with all traffic between Alice and Bob. You can use the following primitives: `hash(msg)`, `sign(asymm_key, msg)`, `MAC(key, msg)`, `aenc(asymm_key, msg)` and `adec(asymm_key, msg)` for asymmetric cryptography as well as `enc(key, msg)` and `dec(key, msg)` for symmetric cryptography primitives. `x | y` denotes the concatenation of `x` and `y`.

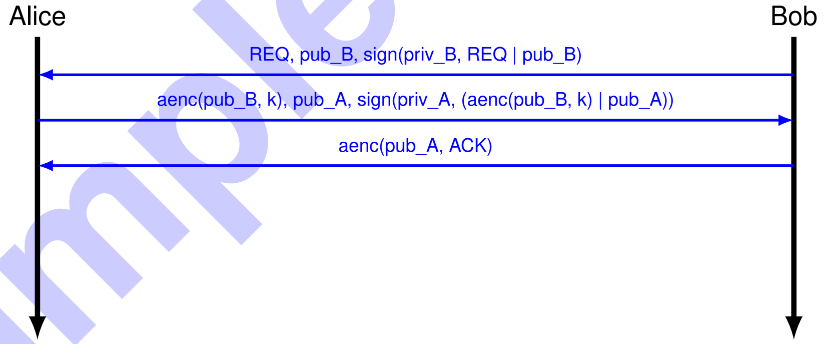
a)* Is this protocol secure? If not, list what it is vulnerable to.

No. It is vulnerable to Man-in-the-Middle attacks, replay attacks, and message tampering since there is no integrity protection b)* Eve wants to eavesdrop on the encrypted messages exchanged. Sketch how Eve can manipulate the key exchange in order to retrieve k. Assume that Bob initiates the communication. Use the key exchange from Alice's protocol in Figure 2.1.

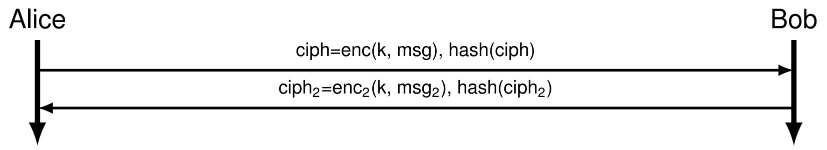


c) Extend Alice's protocol to enable authenticity and integrity protection for the key exchange, thereby preventing Eve from getting the key. How can you solve the problem that Bob initially does not know and trust Alice's key?

First of all, we have to solve the issue that Alice cannot trust Bob's key and vice versa. We do this by either exchanging the public keys in advance through a secure channel (e.g. out-of-band) or by employing a trusted third party that verifies the identity and then issues a certificate for the keypair. For the following, assume that the keys have been exchanged and verified in advance. If the TTP solution is chosen, the protocol has to be modified to include signatures in the first messages.


Alice
Bob
REQ, pub_B, sign(priv_B, REQ | pub_B)
aenc(pub_B, k), pub_A, sign(priv_A, (aenc(pub_B, k) | pub_A))
aenc(pub_A, ACK)

d) Alice proposes the following as integrity protection. Evaluate Alice's proposal.

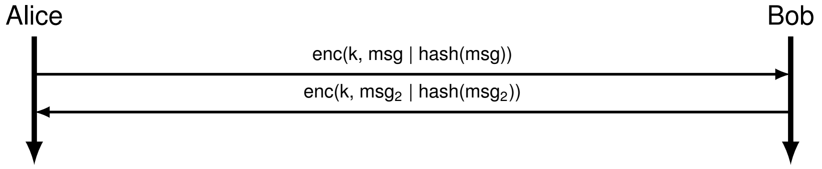

Alice
Bob
ciph=enc(k, msg), hash(ciph)
ciph$_2$=enc$_2$(k, msg$_2$), hash(ciph$_2$)

An attacker can change the encrypted message and afterwards just recompute the hash. The message will thereby seem unmodified. Still, since the message is encrypted, the cleartext cannot be forged by an attacker.

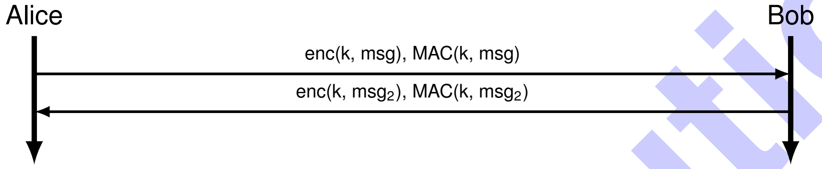
In the following tasks, assume that Bob and Alice have exchanged their public keys through a secure channel in advance.

e) The protocol provides neither integrity nor authenticity for data messages exchanged after k has been established. Extend the protocol with a MAC scheme to protect the messages.

We have to ensure that either, the hash is encrypted and therefore cannot just be changed by an attacker, or alternatively, that the secret influences the hash. In both cases the attacker cannot change the message and then just replace the hash by the correct recomputed value.



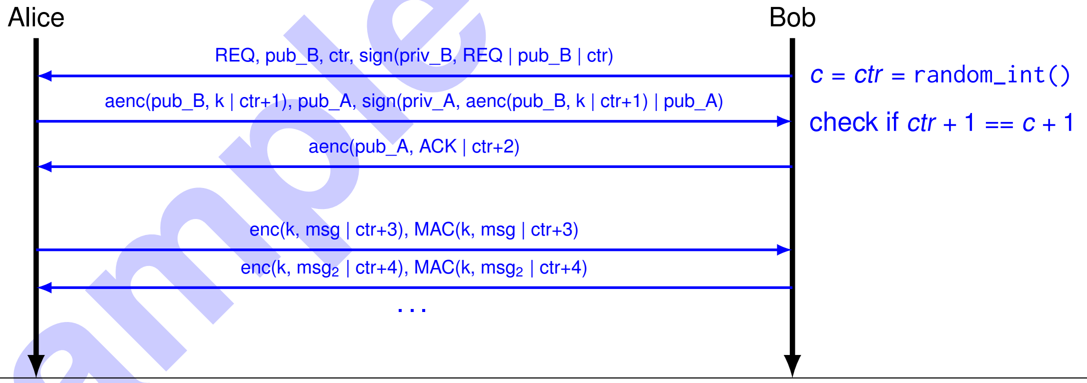
Alternatively:



Alice and Bob have defined a key exchange that provides authenticity and integrity protection as shown below. Thereby, Eve cannot eavesdrop on the key and therefore cannot decrypt the following symmetrically encrypted data messages.

f) Modify your so-far proposed protocol such that replays are detected.

Including a counter in the messages that starts at a random value and is unique per message prevents replays, as the counter will not match the expected value. The expected value can be derived from the last message the regarding side has sent.


Alice
Bob
REQ, pub_B, ctr, sign(priv_B, REQ | pub_B | ctr)
aenc(pub_B, k | ctr+1), pub_A, sign(priv_A, aenc(pub_B, k | ctr+1) | pub_A)
aenc(pub_A, ACK | ctr+2)
enc(k, msg | ctr+3), MAC(k, msg | ctr+3)
enc(k, msg$_2$ | ctr+4), MAC(k, msg$_2$ | ctr+4)
...
c = ctr = random_int()
check if $ctr + 1 == c + 1$