Technische Universität München
Informatik VIII
Netzarchitekturen und Netzdienste
Prof. Dr.-Ing. Georg Carle
Dr. Heiko Niedermayer

Dipl.-Inform. Ralph Holz

Exercises for lecture „Netzsicherheit“
Assignment 4, WS10/11

Hand-out:	Thursday December 16th 2010
Deadline:	Thursday January 13th 2011
Exercise course:	Thursday January 16th 2011

We wish you a Merry Christmas and a Happy New Year! See you in 2011!

Task 1: Public Key Infrastructures 1: X.509

Recall that certificates are used to establish a cryptographically secure binding between an identity and the public key belonging to that entity: Cert(B) = Sig(B, $K_{pub,B}$).
In this task, we will have a closer look at a particular Public Key Infrastructure (PKI), namely the one defined in the X.509 standard. This PKI type is structured as a hierarchy. Today, the most important use case for X.509 certificates is to secure HTTP connections with SSL/TLS. Here, certificates are issued to “Web sites” and browsers rely on them to establish secure connections.

We will begin with a few theoretical questions and then turn to practically evaluating certificates in this task. The following references may be of help.

RFC for X.509:
http://www.ietf.org/rfc/rfc3280.txt
OpenSSL: you will need `openssl`.
Linux/Unix: use your distributions’ package manager to install `openssl`.
Windows: you can choose between the binaries
from http://www.openssl.org/related/binaries.html or download Cygwin, which gives you a UNIX-like environment under Windows: http://www.cygwin.com. The latter might work better as OpenSSL is not officially supported in any other way.

a) Figure 1 shows an “ideal” X.509 PKI that is structured as a tree, with one global CA at its root. Imaginatively, this CA is called “Root CA”. The Root CA does not issue certificates to end-entities itself. Rather, it delegates this task to “Intermediate CAs”. The idea is that the Root CA signs the certificate of the next intermediate CAs down in the tree, which in turn sign the next intermediate CAs further down in the tree etc. Finally, some intermediate CA will sign certificates of “end-entities”.
    a. Assume Alice is an end-entity and has been issued a certificate. Let Alice be the leftmost leaf in Figure 1. Bob wishes to verify that a given public key belongs to Alice. He has obtained her certificate. What does he have to do now in order to check the certificate is valid? Give the steps.
    b. What purpose could Intermediate CAs possibly serve? Give at least one example.
    c. There is an implicit assumption made in this form of PKI that is often called “transitive trust”. Explain: when Bob wishes to verify the certificate, which entities must he “trust” to issue correct certificates in the first step of the verification? What about the next steps?

b) Let's turn to practice. The main use of X.509 today is to allow SSL/TLS-secured connections to WWW servers. Let us have a look at the certificates of a few hosts. Download the following file first:
http://www.net.in.tum.de/pub/ca-bundle-2010-12-10.txt
Then use your OpenSSL installation and issue the following command:
`openssl s_client -connect www.google.com:443 -CAfile ca-bundle-2010-12-10.txt`

   a. What does this command do?
   (Hint: check OpenSSL's help function, e.g. "man" or `openssl help`)
   b. What does the option `-CAfile` do?
   c. Who has issued the certificate for `www.google.com`? Is there an intermediate CA?
   d. What length does Google's public key have?
   e. Have a look at the line that begins with `subject=`. You see a number of "fields", like "`C`", "`ST`" etc. In which field is the DNS name of the Web site encoded? What does the field's abbreviation stand for?
   f. What purpose do the other fields serve (i.e. `C`, `ST`, `O`, etc.)?

c) Copy the part beginning from `-----BEGIN CERTIFICATE-----` to `-----END CERTIFICATE-----` into a file `google.com.cert`. Then issue the following command:
`openssl x509 -in google.com.cert -text`
The result is a text representation of the certificate content. Explain what is stored in an X.509 certificate. You can skip the X.509 v3 Extensions.

d) Have a look inside `ca-bundle-2010-12-10.txt`. The file is a copy of all Root CA certificates that are shipped with the Mozilla Firefox browser (as of 2010-12-10). How many Root CAs are included?
(Note: use the search function of your editor or use "grep" under Linux/Unix. No manual counting, please...)

e) Browse through the file a bit – give the names of 2 CAs that you can identify and state in which country they reside. Optional: If you feel like it, browse a bit more and try to find "interesting" Root CAs. There are some. ☺

f) When Firefox connects to https://www.google.com and receives a certificate – how does it essentially proceed to verify the validity of the certificate? Which entries in the certificate does it need to consider at the very least?

g) You have seen many Root CAs in the root store. Compare the ideal X.509 PKI from Figure 1 with the reality of a root store in a browser. Who gets to choose which CAs are trustworthy – the user or the browser vendor?

h) A little-known fact is that every Root CA in the Mozilla root store is "equal" to every other Root CA in there. This means that every Root CA is allowed to issue certificates for every DNS domain name. What would be the impact for the Mozilla Firefox browser if just one Root CA were actually not trustworthy but malicious?

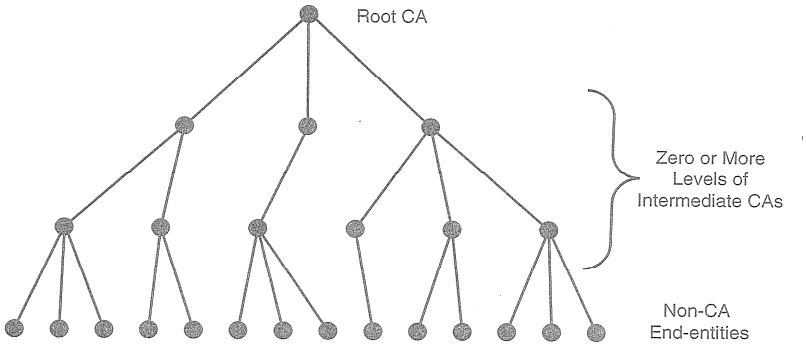

Figure 1: PKI with strict hierarchy.

Task 2: Public Key Infrastructures 2: Webs of Trust

Figure 2 shows a PKI that is based on the "user-centric" model of a Web of Trust. This form of PKI is used in *Pretty Good Privacy (PGP)* and its GNU pendant, *GnuPG* (GPG). Since PGP and GnuPG are interoperable, one often speaks of *PGP/GPG*.
A good introduction to PGP/GPG can be found here:
http://www.wim.uni-koeln.de/uploads/media/The_PGP_Trust_Model.pdf

Download the paper and read Sections 1-3.2. Then answer the following questions.

a) What is the difference in the certification process between the Web of Trust of PGP/GPG and X.509 from Task 1?
b) The graph in Figure 2 shows an example of who has signed whose key. Assume all edges of the graph are bi-directional, i.e. if "A has signed B's key, then B has also signed A's key". According to the trust architecture described in Section 2 of the paper from above, what does Alice have to do to verify Bob's certificate?
c) What is an "introducer" in PGP/GPG?
d) PGP/GPG knows two notions of "trustworthiness": trustworthiness of a public-key certificate and trustworthiness of an introducer. What is the difference? What are the trust levels, and what are their meanings?

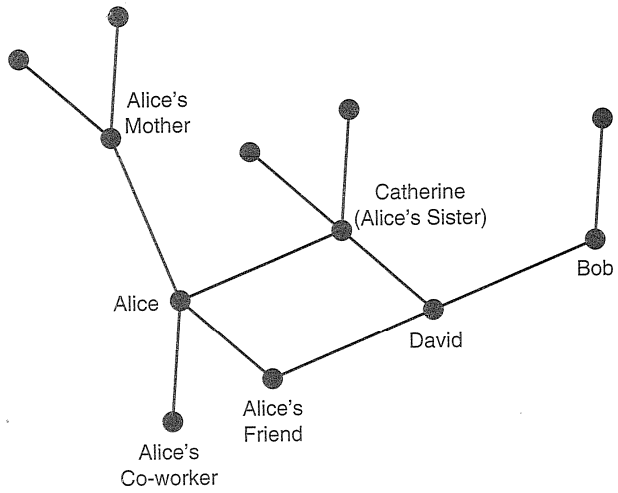

Figure 2: Web of Trust-PKI like PGP/GPG.

Task 3: IPSec: AH and ESP

a) Why does the Authentication Header not protect all header fields of the outer IP header?
b) Consider IP fragmentation: argue why the processing of incoming packets requires all fragments of an IP packet to be reassembled before the processing of IPSec can proceed.
c) What measures are taken in the AH or ESP protocol to prevent a replay attack?
d) Why is it reasonable to check the sequence number is "expected" *before* the cryptographic checks are performed?

Task 4: IPSec – Incompatibility with Network Adress Translation (NAT)

An employee („Road Warrior") is visiting a costumer or partner company (cf. Figure 3). The foreign network only assigns private IP addresses[1] to the network devices, including the computer of our road warrior. The connection to the Internet is established via a so-called Network Address Translator (NAT) that is located in Router A in the given figure. NAT changes the IP addresses of incoming and outgoing IP packets. This is also true for the IP packets from the computer of the road warrior and the

[1] Private IP adress ranges are in the ranges 10.x.x.x, 172.16.x.x oder 192.168.x.x. For more details, see RFC1918.

IPSec gateway B. IPSec is used in Tunnel Mode.

a) What is the resulting conflict when the packets between the computer of the Road Warrior und B are protected with Authentication Header (AH)?
b) Now assume that the NAT additionally changes port numbers of the transport layer protocol (Layer 4)[2]. Why does this also conflict with the use of ESP (unless the encryption algorithm is „NULL“)?
c) RFC3948 describes a solution for this problem of NAT + ESP (→ Section 3.4 there). Why does this solve the problem from b)? Explain by example, using IP adresses and port numbers in the inner and outer IP headers / payload.

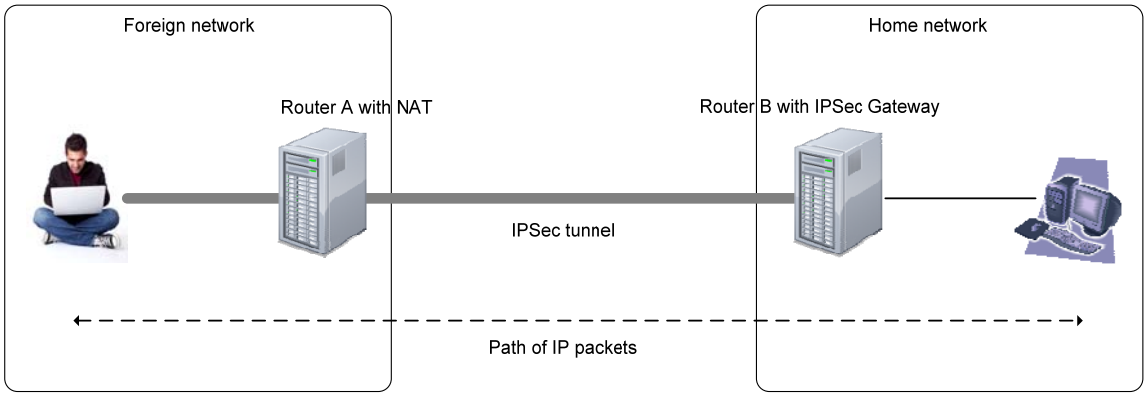

Figure 3: Road Warrior behind NAT.

[2] This kind of NAT, also called NAPT (Network Address and Port Translator), is very common and can be found in DSL routers as well as at the border of company networks with private addresses.