Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Dr. Heiko Niedermayer
Cornelius Diekmann, M.Sc.
Dr. Ralph Holz

Network Security Winter Term 2014/2015
Exercise 5 – Real Exam Problems

Protocol Breaking – Problem 1 Retake Exam 2013

Critical infrastructure, such as industrial control systems (think of nuclear power plants), is sometimes operated over the Internet. In the following, Alice A and Bob B are industrial control devices. We give a cryptographic protocol between them. Alice and Bob have securely exchanged a symmetric shared key $k_{A,B}$. Also, a Ticket Granting Server TGS is involved in the protocol. Alice shares a symmetric key $k_{A,TGS}$ with the Ticket Granting Server. Also, Bob shares a symmetric key $k_{B,TGS}$ with the TGS. All participants have synchronized clocks.
Alice and Bob only have limited resources for cryptographic operations. However, both devices have a dedicated hardware module for the AES algorithm with 128 bit. For encryption, AES-CTR is used. We use the following notation: $IV, Enc_k(m)$ is the message m encrypted with AES in CTR mode with key k and initialization vector IV. Assume that all IVs are fresh. For checksums, the cyclic redundancy check (CRC) is used. $CRC(m)$ denotes the CRC checksum of the message m.
Alice wants to send a command to Bob. We describe the cryptographic protocol.
First, Alice requests a ticket at the TGS for her command to Bob.

$$A \longrightarrow TGS : IV, Enc_{k_{A,TGS}}(\text{“}Alice\text{”}, \text{“}Bob\text{”}, timestamp, ticketLifetime) \tag{1}$$

The TGS verifies that *timestamp* is not older than five Minutes and that *ticketLifetime* is not more than 30 minutes. The TGS then replies with Alice's ticket.

$$TGS \longrightarrow A : IV, Enc_{k_{A,TGS}}(IV, Enc_{k_{B,TGS}}(\text{“}Alice\text{”}, \text{“}Bob\text{”}, timestamp, ticketLifetime)) \tag{2}$$

Alice extracts the ticket $Ticket_{A,B} = IV, Enc_{k_{B,TGS}}(\text{“}Alice\text{”}, \text{“}Bob\text{”}, timestamp, ticketLifetime)$. The ticket is 128 bit long.

The industrial control protocol is a binary protocol. We provide a short extract from the documentation. It consists of the following fields in the following order. The numbers indicate the position of the bits:

```
  0 -   3: Version, must be set to 2 (4 bits)
  4 -  15: Total length (12 bit)
 16 -  31: Id of source (16 bits)
 32 -  47: Id of destination (16 bits)
 48 - 175: Ticket (128 bits)
176 - 178: Flags (3 bits)
      176: Shutdown Flag, shutdown the device (1 bit)
      177: Reboot Flag, reboot the device (1 bit)
      178: Overload Flag, Warning: May physically damage the device! (1 bit)
179 - 340: Command (162 bit)
341 - 372: CRC checksum over all previous fields (32 bit)
373 - 501: Padding (128 bit)
```

Alice constructs the command m for Bob according to the documentation. In particular, she sets the version to two, calculates the lengths, sets "Alice" as source and "Bob" as destination, includes her ticket $Ticket_{A,B}$, and correctly calculates the CRC checksum.

$$A \longrightarrow B : IV, Enc_{k_{A,B}}(m) \tag{3}$$

Bob decrypts the message. He verifies the CRC and that the destination id is set to "Bob". Then, Bob extracts the values from $Ticket_{A,B}$ and verifies the source and destination again with this information. He also verifies that the the ticket is not expired. Finally, Bob executes the command according to the command field and the flags.

a) Show that an active attacker is able to cause severe damage to the control system. Give the **complete modified** message exchange in the same style as above and *clearly point out why your attack works*. If you make changes to binary data, explain exactly how it works.

b) Now fix the protocol so that no attack (not only last task's attack) is possible. Mind the resource limitations. You can make use of the AES hardware module but you cannot use other expensive cryptographic operations. You may use up to 512 bit of permanent storage in Alice and Bob. The TGS is not resource-constrained. Denote the *complete flow* of the changed protocol in the style as above and *and explain why your version is secure now*. Hint: 1credit for replay protection.

IPSec – Problem 2 **Retake Exam 2013**

In the following, you will have to apply your knowledge about IPSec.

a) Explain why IPSec associations must generally be set up with administrative rights.

b) In the context of IPSec, what does SPD and SAD stand for?

c) Figure 1 shows a setup that interconnects two networks. The goal is to protect HTTP traffic, and *only* HTTP traffic, between I8 NETWORK and ELITE NETWORKS with IPSec. However, only the gateways G1 and G2 are IPSec-enabled. The goals to be achieved are:

- HTTP Traffic between I8 NETWORK and ELITE NETWORKS must be confidential, i.e. noone on the Internet can read it.

- HTTP Traffic between I8 NETWORK and ELITE NETWORKS must be integrity-protected. The protection must include IP header fields such as IP source address, i.e. noone on the Internet can spoof source addresses (neither of a sending host nor of a gateway). Noone on the Internet can modify any payload.

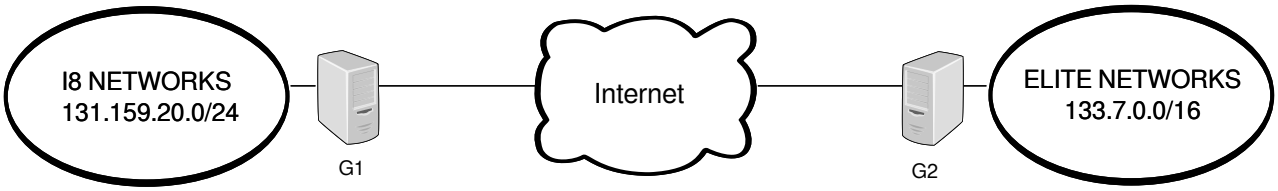
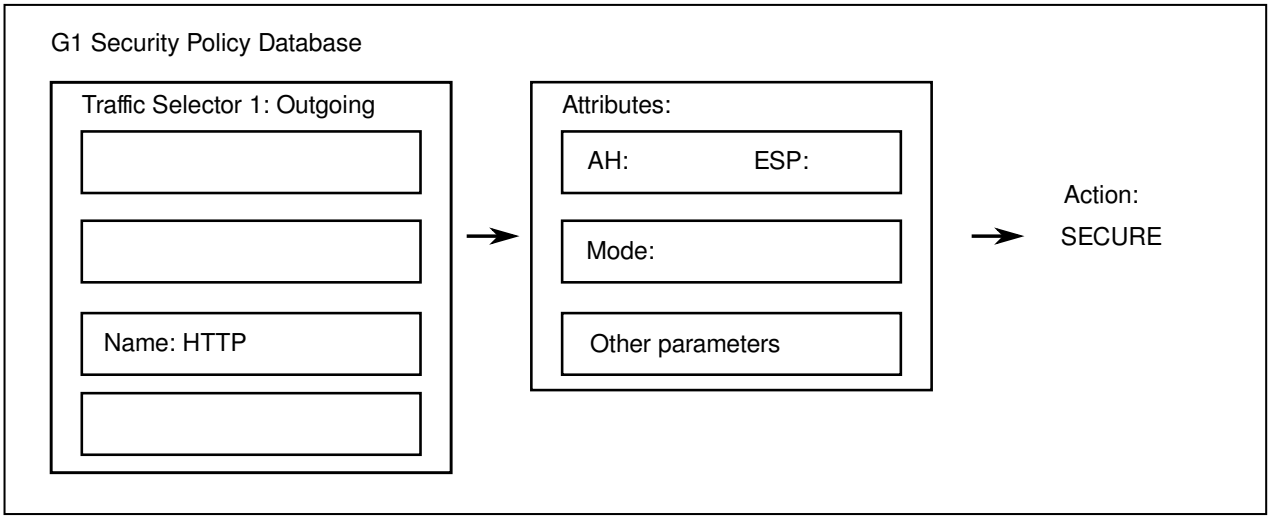
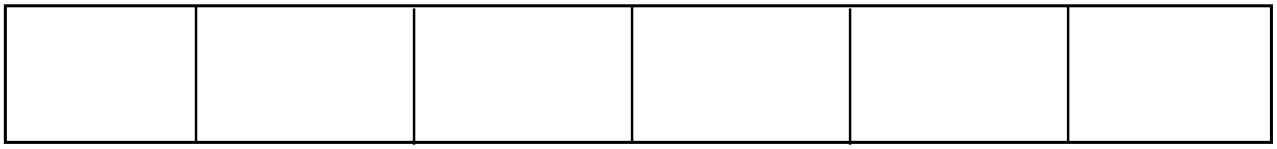


Figure 1: The two networks to be connected with IPSec.

Fill out the Security Policy Database for the *outgoing* direction at gateway G1 with the necessary values! Note: For AH and ESP, simply indicate 'Yes' or 'No'. You can assume that I8 NETWORK only contains clients and the servers are in ELITE NETWORKS.



d) For the solution that you found for c), fill in the layout of a typical IPSec-protected IP packet that has just left G1. You may choose only from the following field names: IP, AH header, ESP header, ESP trailer, payload.



e) Explain in how many security associations the setup from a) results in, and why!

f) Describe what is stored in *one* security association of IPsec-secured traffic from I8 NETWORK to ELITE NETWORKS.