Security Protocols and Application — Final Exam

Philippe Oechslin and Serge Vaudenay

16.6.2015

- duration: 3h00
- no document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

1 Relay Cost Bounding

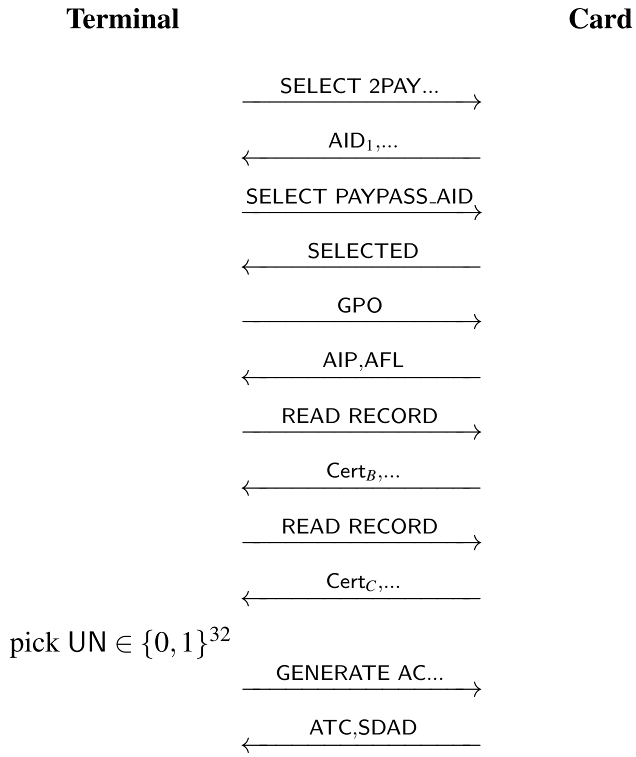
This exercise is inspired by the article *Relay Cost Bounding* by Chothia *et al.*, Financial Cryptography 2015. Questions Q.1, Q.2, and Q.3 are independent.

Q.1 Assume that a credit card holder is quietly taking the metro in Lausanne while someone is willing to order some drinks at Satellite — a popular bar — and make the payment through a relay attack.

Q.1a Describe the attack scenario and how many people and devices are involved.

Q.1b Assume that Satellite is located on the moon. Given that the Earth-Moon distance is of $384\,000\mathsf{km}$ and that the speed of light is of $300\,000\mathsf{km.s}^{-1}$, what communication delay would be introduced for each message from the card to the payment terminal and each message from the payment terminal to the card?

Q.2 We consider the PayPass payment protocol. (The meaning of the messages is not important for this exercise.)



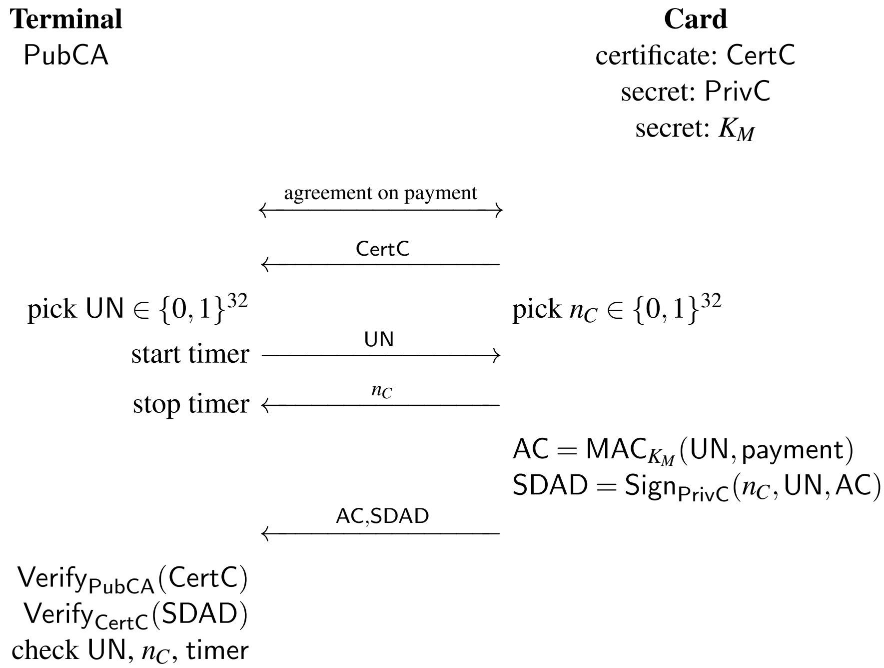
Given

- that standard equipments introduce a delay of 100ms on the top of the time of flight to relay a single message;
- that some cards take 637ms to complete all computations required on their side during the protocol;
- that payment terminals reject payments if the total protocol duration exceeds 1s;

Q.2a show that the standard relay attack does not work.

Q.2b In their paper, how did Chothia *et al.* adapt the relay attack to make it work?

Q.3 We now consider a man-in-the-middle attack against the following payment protocol (which is a simplification of PaySafe). The terminal holds the root certificate PubCA of the Card's PKI. The card holds a certificate CertC (to be verified with PubCA) and a secret key PrivC which allows it to sign. Signatures are verified with CertC. The card also holds a secret key K_M which is shared with the bank. This key is used to authenticate a message by means of a MAC algorithm.



At the end, the terminal accepts the payment in SDAD if the certificate is valid, if the signature is valid, if the nonces in SDAD are correct, and if the elapsed time between sending UN and receiving n_C is lower than a given bound B.

Q.3a We consider any man-in-the-middle attack in which the man-in-the-middle sends some random UN to the card before he receives UN from the payment terminal. Show that the attack fails, except with small probability. (Make a detailed proof.)

Q.3b We consider any man-in-the-middle attack in which the man-in-the-middle sends some random n_C to the payment terminal before he receives n_C from the card. Show that the attack fails, except with small probability. (Make a detailed proof.)

Q.3c Assuming that relaying a message with standard equipments introduces a delay of 100ms, adjust B and prove that no man-in-the-middle attack can break the protocol. (Make a detailed proof.)

2 Heartbleed

Q.1 Below is the source code of the OpenSSL library that includes the Heartbleed bug.

```
dtls1_process_heartbeat(SSL *s)
  {
  unsigned char *p = &s->s3->rrec.data[0], *pl;
  unsigned short hbtype;
  unsigned int payload;
  unsigned int padding = 16; /* Use minimum padding */

  /* Read type and payload length first */
  hbtype = *p++;
  n2s(p, payload);
  pl = p;

  if (s->msg_callback)
        s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
              &s->s3->rrec.data[0], s->s3->rrec.length,
              s, s->msg_callback_arg);

  if (hbtype == TLS1_HB_REQUEST)
        {
        unsigned char *buffer, *bp;
        int r;

        /* Allocate memory for the response, size is 1 byte
         * message type, plus 2 bytes payload length, plus
         * payload, plus padding
         */
        buffer = OPENSSL_malloc(1 + 2 + payload + padding);
        bp = buffer;

        /* Enter response type, length and copy payload */
        *bp++ = TLS1_HB_RESPONSE;
        s2n(payload, bp);
        memcpy(bp, pl, payload);
        bp += payload;
```

Q.1a Mark the lines were the error occurs and explain what is wrong with it.

Q.1b Heartbleed leaks memory. Which part of memory is being leaked ?

Q.2 Exploiting Heartbleed:

Q.2a Describe a scenario in which an attacker makes use of Heartbleed to access a victim's e-banking application even if that application uses two-factor authentication.

Q.2b Describe a scenario which explains why the private key of a web server can be retrieved using Heartbleed even if a Diffie-Hellman key exchange is used and thus the private key is not used for the key exchange.

Q.2c Why is it not sufficient to overwrite the private key after it has been used, to remove it from memory ?

Q.3 Preventing Heartbleed:

Q.3a Administrators are told that they need to update their software regularly to avoid security issues. Explain why in the case of Heartbleed, people who update regularly were more impacted than others.

Q.3b Several groups of developers have decided to fork their own version of OpenSSL. Cite the name of two such projects or groups:

Q.3c Do you believe that having different versions of OpenSSL that are developed separately will increase its security ? Give a justification for your answer.

Q.4 What good is it for anyways ?

Q.4a Give one in example in which using a heartbeat request is more useful than doing a simple ping at the network level.

Q.4b Give one example in which it is useful to have a mechanism for sending heartbeat packets of *variable* length.