UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

# COMPUTER SECURITY

**Monday 11th August 2014**

**14:30 to 16:30**

## INSTRUCTIONS TO CANDIDATES

**Answer any TWO questions.**

**All questions carry equal weight.**

**CALCULATORS MAY NOT BE USED IN THIS EXAMINATION**

Year 3 Courses

Convener: S. Viglas
External Examiners: A. Cohn, T. Field

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. A Message Authentication Code (MAC) consists of two algorithms $(S, V)$. Algorithm $S(k, m)$ uses a secret key $k$ to generate an integrity tag for a message $m$. Algorithm $V(k, m, t)$ uses a secret key $k$ to validate a given integrity tag $t$ for $m$. Recall that a cryptographic hash function $h$ is a non-keyed function that outputs a short hash $h(m)$ for an input message $m$. The function is said to be collision resistant if it is difficult to find a collision: two distinct messages $m_0$, $m_1$ such that $h(m_0) = h(m_1)$.

   Let us consider four mechanisms for providing file integrity for a single file $F$ on disk. The file system must be able to detect any unauthorized modification to this file. We say that the system is secure if an attacker cannot modify $F$ without being detected. You may assume that the owner of file $F$ has a password known to the system, but not to the attacker.

   Method 1: Compute an integrity tag for file $F$ and store the integrity tag in the header of $F$ . Upon file open the file system checks that the integrity tag is valid.

   (a) Suppose the integrity tag is computed using a collision resistant hash function applied to $F$. Validating the integrity tag upon file open is done by rehashing the file and comparing the result to the value in the file header. Is the resulting system secure? (Justify your answer) [*5 marks*]

   (b) Suppose the integrity tag is computed as the MAC of $F$ using the user's password as the MAC secret key. Is the resulting system secure? (Justify your answer) [*5 marks*]

   Method 2: Compute an integrity tag for file $F$ and store the integrity tag in read only memory (say, a disk partition that the attacker can read but not modify).

   (c) Suppose the integrity tag is computed using a collision resistant hash function. Is the resulting system secure? (Justify your answer) [*5 marks*]

   (d) Suppose the integrity tag is computed using a MAC with the user's password as the secret key. Is the resulting system secure? (Justify your answer) [*5 marks*]

   (e) Propose a mechanism, that doesn't rely on read only memory, and that provides file integrity. (Justify your answer) [*5 marks*]

2. The (fictional) security comany *STB Devices* produces a *secure token block* which is a proprietary software code module that can be embedded into mobile devices used for second factor authentication. The STB module implements a *Time-based One-time Password Algorithm*. It works as follows:

   - $X$ is a system-specific parameter, the *time-step size*
   - $T_0$ is the *starting time*, also system-specific.
   - For each use, where *UnixTime* is the number of seconds since the epoch, the module calculates:

$$T = (UnixTime - T_0)/X$$
$$OTP(k, T) = Truncate(hmac(k, T))$$

     where *hmac* is the standard HMAC algorithm based on SHA-1, and the *Truncate* function chops the 160-bit resulting hash into an 8 decimal digit code that can easily be entered by the user onto a web page.

   The one-time password $OTP(k, T)$ is based on a shared key stored in the STB module and in the authentication server. The authentication server is also sold by STB Devices as a self-contained web application running on a Unix machine.

   (a) Briefly, describe in general what is meant by *second factor authentication*; explain the principal benefit of using a second factor mechanism and explain the main underlying assumption concerning the second factor. [*3 marks*]

   (b) Explain three additional assumptions in this specific case, necessary for the STB module to provide trustworthy additional authentication of Alice, and preventing an imposter Mallory from successfully authenticating. [*3 marks*]

   (c) To allow for some network delay, the authentication server will accept OTPs for its current time step $T_i$ but also for one previous time step $T_{i-1}$.
   Give an explanation of the steps the authentication server takes starting from a login attempt by Alice, to decide whether to authenticate her. [*5 marks*]

   (d) Following on from above, suppose that $X$ is 30 seconds. Explain what may happen if Alice attempts to log in twice within one minute. [*2 marks*]

   (e) A classic study recommended a number of general security design principles including these two:

      i. *separation of privilege*
      ii. *least privilege*
      iii. *open design*

   Explain briefly what these principles means and, in more detail, how each one might be interpreted to apply to the design of the STB code module *and* its deployment. [*12 marks*]

3. Alice always signs her email with a GPG (GNU Privacy Guard) signature, using the Thunderbird mail client to automatically attach signatures on outgoing messages. She uses a single-user desktop machine, but her files are kept on a networked fileserver which serves around 20 machines on an internal firewalled network. Her private keys are protected using a passphrase, which she inputs when she starts her mail client program.

   One day, Alice is summoned to her manager's office and is asked to explain an offensive email which appears to have come from her. It contains a signature that verifies in the company's webmail application. Alice did not send the email, and wants to find out how it could appear to have come from her.

   (a) Using an *attack tree* to structure your answer, provide an analysis of the possible ways that this threat may have been realised.

   Recall that a simple attack tree is an AND-OR tree with an attacker's goal at the root. The AND nodes in the tree indicate sub-goals that must all be achieved, while OR nodes indicate alternative possibilities. The leaf nodes are labelled with *Possible* or *Impossible* according to the judgement of the security analyst.

   You should include at least 4 different possibilities in the first (OR) level of decomposition and at least 10 leaf nodes. Paths on the tree should cover (and highlight) at least one instance of each kind of the following generic problems: *access control mistake*, *confidentiality leak*, *authentication error*, and *input validation failure*.

   Be sure to label the nodes with *Possible* or *Impossible*, using your background knowledge and assumptions about the specific scenario.

   You do not have to draw the tree graphically, but be careful to indicate the tree structure in your answer and label nodes as AND or OR. To explain the tree, give **brief** descriptions for the nodes. [*16 marks*]

   (b) Identify two likely attack routes among paths in your tree which end with possible nodes. Explain why you believe these are likely. [*4 marks*]

   (c) Describe appropriate responses for Alice and her manager to take as the next steps. And, in the longer term, explain how to reduce the vulnerabilities which lead to the two attack routes you identified. [*5 marks*]