NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Telematics

Examination paper for TTM4135 Information security

Academic contact during examination: Colin Boyd

Phone: 73551758

Examination date: 2015-08-03

Examination time (from-to): 09:00 - 12:00

Permitted examination support material: (D) No printed or hand-written support material is allowed. A specific basic calculator is allowed.

Other information: –

Language: English

Number of pages: 2

Number of pages enclosed: 0

Checked by:

Date Signature

TTM4135 August exam 2015:
Outline answers

Exercise 1 Multiple choice questions

Candidate number: □□□□□

1.	(a) □	(b) ☑	(c) □	(d) □
2.	(a) □	(b) □	(c) ☑	(d) □
3.	(a) ☑	(b) □	(c) □	(d) □
4.	(a) □	(b) ☑	(c) □	(d) □
5.	(a) □	(b) ☑	(c) □	(d) □
6.	(a) □	(b) □	(c) □	(d) ☑
7.	(a) □	(b) □	(c) □	(d) ☑
8.	(a) □	(b) □	(c) □	(d) ☑
9.	(a) □	(b) □	(c) ☑	(d) □
10.	(a) ☑	(b) □	(c) □	(d) □
11.	(a) □	(b) □	(c) ☑	(d) □
12.	(a) ☑	(b) □	(c) □	(d) □
13.	(a) □	(b) ☑	(c) □	(d) □
14.	(a) □	(b) □	(c) □	(d) ☑
15.	(a) □	(b) □	(c) ☑	(d) □
16.	(a) □	(b) ☑	(c) □	(d) □
17.	(a) ☑	(b) □	(c) □	(d) □
18.	(a) □	(b) □	(c) □	(d) ☑
19.	(a) □	(b) ☑	(c) □	(d) □
20.	(a) □	(b) □	(c) ☑	(d) □

Exercise 2 Written answer questions

1. (a) Left and right halves of the current block state (in round i) and the round key.

   (b)

$$\begin{aligned} R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f(L_i, K_i) \end{aligned}$$

   (c) As can be seen in the decryption equation, f is only used in the forward direction.

2. (a) With a fixed IV there is no randomness so that each message is always encrypted the same way. In particular the first block is always encrypted the same way. An attacker can build up a dictionary using known or chosen plaintext attacks and decrypt blocks in the dictionary.

   (b) A valid MAC for message $M = P_1$ is $E(P_1 \oplus IV, K)$. Now to find a MAC for $M' = P_1'$ choose IV' so that $P_1' \oplus IV' = P_1 \oplus IV$ - that is $IV' = P_1 \oplus IV \oplus P_1'$. Then the same MAC is valid for M', replacing IV by IV'.

3. $M_p = C^{27 \bmod 4} \bmod 5 = C^3 \bmod 5 = 3$

   $M_q = C^{27 \bmod 10} \bmod 11 = C^7 \bmod 11 = 128 \bmod 11 = 7$

   $M = (3 \times 11 \times 11^{-1} \bmod 5) + (7 \times 5 \times 5^{-1} \bmod 11) = 33 + (35 \times 9) \bmod 55 = 18.$

4. (a) Compute $a^2 \bmod n$, $a^4 \bmod n$, $a^8 \bmod n$, $a^{16} \bmod n$, $a^{32} \bmod n$

   Now multiply $a \bmod n$, $a^4 \bmod n$ and $a^{32} \bmod n$.

   So there are 5 squarings and 2 multiplications required.

   (b) On average there are approximately 2000 squarings and 1000 multiplications required. The number of multiplications in each case depends on w, the number of 1 bits in the exponent b: there are $w - 1$ multiplications.

5. (a) By taking discrete logs the attacker can find k (or x). Then he can compute $y^k \bmod p$ from y (or $g^k \bmod p$) and hence divide this out from the second part of c to obtain m.

   (b) Reasonable choices today are at least 1024 and preferably 2048 bits. Too large a value decreases efficiency, lower than 1024 risks being broken by finding discrete logs. Realistic quantum computers will make taking discrete logs efficient (polynomial time) and there is no reasonable size which can work any longer.

6. (a) Verifier recomputes $h(m) = h'$ for received message. Checks that $s^e \bmod n = h'$.

   (b) Attacker finds m and m' with $h(m) = h(m')$. Then obtains a signature s on m, for example with a chosen message attack. Then s is a forged signature for m'.

7. Since B never uses a nonce or any other replay detection the replay must be against him. The attacker can eavesdrop an old token $T' = \{K', A, B, N_A\}_{K_{AS}}$ and obtain the old key K'_{AB}. Then the attacker C masquerades as B and starts a new session with the following messages.

$$\begin{aligned} A(C) \to B &: A, N_A \\ B \to S(C) &: A, B, N_A \\ C \to B &: X, T' \\ B \to A(C) &: X \end{aligned}$$

   C takes the roles of both A and S while X is just a random string which is not used by C. The exchange looks normal to B who accepts the old key K' .

8. (a) Both S and C compute the Diffie–Hellman value g^{xy}, S by $(g^y)^x$ and C by $(g^x)^y$. g^{xy} is the premaster secret which is used as input to a key derivation function from which the session keys are computed.

   (b) The only long-term key used is the server signing key. If this is revealed after the protocol is run, it does not help an attacker to recover the Diffie–Hellman value.