NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Information Security and Communication Technology

Examination paper for TTM4135 Information security

Academic contact during examination: Colin Boyd

Phone: 73551758

Examination date: 2017-05-19

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 May exam 2017:
Outline answers

Exercise 1 Multiple choice questions

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) ☐
21.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
22.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
23.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
24.	(a) ☐	(b) ☑	(c) ☐	(d) ☐
25.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
26.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
27.	(a) ☐	(b) ☐	(c) ☑	(d) ☐
28.	(a) ☐	(b) ☐	(c) ☐	(d) ☑
29.	(a) ☑	(b) ☐	(c) ☐	(d) ☐
30.	(a) ☐	(b) ☐	(c) ☐	(d) ☑

Exercise 2 Written answer questions

1. (a) Add in the same keystream value so that: $P_t = O_t \oplus C_t$ where $O_t = E(T_t, K)$ and $T_t = N \| t$ is the concatenation of a nonce N and block number t.
   (b) Each bit in the ciphertext is just the addition of the corresponding plaintext and keystream bits. Therefore flipping a bit in the ciphertext only affects one plaintext bit in the same position.
   (c) The last block makes a very poor MAC because it depends only on the last block of the message. Therefore a valid MAC tag can be forged by an attacker for any message which ends with the same final block as an observed message.

2. (a) An element a is an inverse for x if $ax = 1 \bmod p$.
   (b) There are a few ways to find this, including trial and error. Systematically using Euclid, $13 = 4 \times 3 + 1$ so $1 = -4 \times 3 \bmod 3$. So $3^{-1} \bmod 13 = -4 \bmod 13 = 9$.
   (c) We need to find the elements which are co-prime with 15. These are $\{1, 2, 4, 7, 8, 11, 13, 14\}$.

3. (a) First compute $a^2 \bmod n$, $a^4 \bmod n$, $a^8 \bmod n$, $a^{16} \bmod n$, $a^{32} \bmod n$, $a^{64} \bmod n$. This requires 6 squarings. Then multiply $a^{64} \bmod n \times a^4 \bmod n \times a^2 \bmod n \times a^1 \bmod n$. This requires 3 multiplications.
   (b) When n is 2400 bits in length we need to use 2400 squarings to compute $a^{2^{2400}} \bmod n$. We then need to multiply as many elements from these squares as correspond to the 1 bits in the binary representation of b. On average half of these are 1 so we expect around 1200 multiplications (or 1199 to be more precise).

4. (a) Since $s = h(m)^d \bmod n$ for a valid signature, $s \times h(m)^e \bmod n = h(m)^d \times h(m)^e = h(m)^{d+e} \bmod n$. Also $d = -e \bmod \phi(n)$ so that $d + e = k \times \phi(n)$ for some integer k. Thus $s \times h(m)^e \bmod n = h(m)^{k \times \phi(n)} = (h(m)^{\phi(n)})^k = 1 \bmod n$ by Euler's theorem.
   (b) The attacker can compute directly a valid s since $s = (h(m)^e)^{-1} \bmod n$. This is because e and m are public and finding inverses modulo n can be done efficiently. (Note that it is not possible to find d directly without finding $\phi(n)$, which is as hard as factorising n.)

5. (a) N_1 is used so that C can check the freshness of the response from AS. C chooses N_1 at random and sends it in the message to AS. AS simply includes it in its response message. C checks that the value received back is the same as the one sent.
   (b) ID_C is included so that the user of the ticket (the TGS) knows which party shares the key $K_{C,tgs}$. If it was removed then a malicious C could masquerade as any other user when the ticket is used later.

6. (a) PGP provides end-to-end security, so it hides the message content from mail servers, but it does not hide the message headers such as source and destination. STARTTLS only provides link security so it does not hide anything from participating email servers which could be malicious. STARTTLS is opportunistic and servers can refuse to use it.
   (b) PGP is processed only by clients so does not affect server processing. Clients may need to encrypt/decrypt or sign/verify depending on services provided. STARTTLS is processed by servers for SMTP so they need to set up a TLS connection and use it. Servers may need to perform key exchange, encryption and authentication on the TLS connection. STARTTLS can also be used for IMAP and POP and can also affect client processing.