Exam Security

20 January 2015, 9:30–11:30

You can score a maximum of 100 points. Each question indicates how many points it is worth. You are NOT allowed to books or notes, or a (smart) phone. You may answer in Dutch or in English. Please write clearly, and don't forget to put your name and student number on each page that you hand in. You can keep this exam yourself.

1. (**12 points**) The creator of PGP is Phil Zimmerman. Of course, he has his own RSA key pair for use in PGP. Online you can find:

Phil's key fingerprint is:	9E 94 45 13 39 83 5F 70 7B E7 D8 ED C4 BE 5A A6
Phil's Key Id is:	C7A966DD

   The fingerprint is the hash $h(n, e)$ of the public key (n, e). The example contains 32 hexadecimals; it is thus a 128 bit hash value. The Key Id consists of 8 hexadecimals, corresponding to 32 bits. It is a part of the public key itself.

   (a) (**6 points**) Some people put their Key ID on their personal webpage, or on their business cards. Other, more security aware people, insist on putting their key fingerprint instead. Why is this more secure? Explain briefly what could go wrong if you publish only your Key ID.

   (b) (**6 points**) On an implementation level an encrypted (PGP) message $\{m\}_{(n,e)}$ is of the form

   $$keyid \mid \cdots$$

   where $\mid$ is concatenation. Thus it has the Key ID as a prefix. Is it more secure in this case to use the key fingerprint instead of the Key ID as prefix? Answer with Yes/No, and explain briefly.

2. (**21 points**) Consider smart cards using symmetric encryption. Each card C has its own "diversified key". We write id_C for some unique identifier of card C, known by the card itself. Card readers do not have the keys of all cards. (This takes too much memory, and is too risky and inflexible). Instead, card readers should be able to somehow compute the (symmetric) key K_C of card C from the card's identity id_C. Let's write this computation as $K_C = f(\mathrm{id}_C)$ for some function f. What f precisely is, does not matter for the moment.

   (a) (**7 points**) Write a short protocol with which the card can authenticate itself towards the reader, using this function f. Write this protocol in standard protocol notation, starting with the message $C \to R : id_C$ where C = card and R = reader.

   Below we list three possible versions of the function f, where h is some hash function, and M is a secret master key that is known to the "scheme manager" that operates these cards. Each card reader knows M.

   (i) $K_C = M\{id_C\}$
   (ii) $K_C = h(id_C)$
   (iii) $K_C = h(id_C \mid M)$.

   (b) (**7 points**) One of these choices of f is insecure, in the sense that card-authentication is not achieved. Which choice? Briefly explain an exploit.

   (c) (**7 points**) Concentrate on the above third option $K_C = h(id_C \mid M)$. Adapt, if needed, your protocol in (a), without changing the number of lines/messages, in such a way that the card knows that the reader knows M. Does this authenticate the particular reader that the card is communicating with? Answer Yes/No, and briefly explain your answer.

3. (**24 points**) The counter (or CTR for short) mode of operation is increasingly popular. It can be used with many encryption functions; here we make use of AES.

   Alice and Bob share a 128-bit AES key K. When Alice wants to encrypt a message to Bob, she generates a fresh 80-bit-long nonce N, and sets the counter CTR to 0 (represented with appropriate bit length). She also divides the message in blocks of length 128 bits: $m_0, m_1, \ldots, m_k$. Finally, the ciphertext is computed block by block as described below and sent to Bob along with the plaintext nonce N. As usual, concatenation is denoted by $|$ and bitwise XOR by $\oplus$.

   $$c_0 = K\{N \mid CTR\} \oplus m_0,\ c_1 = K\{N \mid CTR + 1\} \oplus m_1, \ldots,\ c_k = K\{N \mid CTR + k\} \oplus m_k.$$

   In this notation the nonce N concatenated with the values $CTR + i$ fill one block. The resulting ciphertext is:

   $$K\{m\} = (N \mid c_0 \mid c_1 \mid \ldots \mid c_k).$$

   (a) (**6 points**) Using one nonce N, what is the maximum number of message blocks m_i for Alice's message that she can securely encrypt? Explain your answer briefly.

   (b) (**6 points**) Describe how Bob can decrypt the received ciphertext $K\{m\}$. Which property of $\oplus$ is used?

   (c) (**6 points**) Write down at least two favorable properties of the counter mode.

   (d) (**6 points**) Assume that the bit length of the message is not a multiple of 128, that is, the last message block m_k is less than 128 bits. How can Alice still encrypt this message to make sure that Bob receives it correctly? Explain briefly why your construction is secure.

4. (**28 points**) Consider the RSA cryptosystem. Assume that Alice's public key is ($n_A = 55, e_A = 3$) and Bob's public-private key pair is ($n_B = 55, e_B = 17, d_B = 33$).

   (a) (**5 points**) Assume that you want to send an encrypted message $m_1 = 3$ to Bob. What is the ciphertext?

   (b) (**5 points**) Verify that $d_B = 33$ is indeed the private key of Bob.

   (c) (**5 points**) What is Alice's secret key? Make all computations explicit.

   (d) (**8 points**) We will use the blind RSA signature. Have Bob sign the message $m_2 = 13$ for you using $r = 3$ as the random value.

   (e) (**5 points**) Verify that the signature in the previous point is valid.

5. (**15 points**) In the previous exercise Alice and Bob use the same modulus, namely $n = 55$. In general, this is bad. In practice users don't have access to the factorisation $n = p \cdot q$ of their own modulus, but they do know $\varphi(n)$. We will use this below.

   Assume You and Bob both use $n = 1591$ as modulus, and thus both know $\varphi(n) = 1512$.

   (a) (**5 points**) Let the public key of Bob be $(1591, 5)$. Now you can compute Bob's private key. Do it!

   (b) (**10 points**) You can also factorise n. Make this explicit, step-by-step, and compute the factors.