Re-Exam Security

7 May 2015, 8:30–10:30

You can score a maximum of 100 points. Each question indicates how many points it is worth. You are NOT allowed to use books or notes, or a (smart) phone. You may answer in Dutch or in English. Read each exercise carefully (RTFE). 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. (**22 points**) (authentication and signatures)
   Consider the next two protocols between Alice and Bob. They each have their own RSA modulus, public, and private exponents (n_A, e_A, d_A) and (n_B, e_B, d_B); h is a (secure) hash function.

   (1)
   $$\begin{cases} A \longrightarrow B : \text{contract} \\ B \longrightarrow A : \text{ok} \\ A \longrightarrow B : h(\text{constract}) \\ B \longrightarrow A : [h(\text{constract})]_{d_B} \end{cases}$$

   (2)
   $$\begin{cases} A \longrightarrow B : \{N_A\}_{e_B} \\ B \longrightarrow A : N_A, \{N_B\}_{e_A} \\ A \longrightarrow B : N_B \end{cases}$$

   The hash values $h(x)$ and nonces N_A and N_B all have a length of 128 bits. Alice and Bob use the same keys for both protocols.

   (a) (**4 points**) What is the goal of protocol (1)? Give a brief and concise description.

   Begin Secret Info: .....
   Bob puts a digital signature on a contract that is proposed by Alice

Grading (total 4):	
aspect:	*points*
"contract signing"	*4*
integrity (partial point)	*1*
authenticity (partial point)	*1*
non-repudiation (partial point)	*1*

   End Secret Info .....

   (b) (**6 points**) What does Bob need to check at which stage in protocol (1)? What should Alice check?

   Begin Secret Info: .....
   Bob needs to check two things:

   i. the contents of the contract, sent in message 1

   ii. that the hash sent in message 3 is the hash of the contract in message 1

   Alice needs to check one thing:

   i. the signature of Bob in message 4

Grading (total 6):	
aspect:	*points*
each of three	*2*

End Secret Info .....

(c) (**4 points**) What is the goal of protocol (2)?

Begin Secret Info: .....
Mutual authentication, first of Bob to Alice via N_A, then of Alice to Bob, via N_B.

Grading (total 4):	
aspect:	*points*
authentication of Bob	*2*
authentication of Alice	*2*

End Secret Info .....

(d) (**8 points**) Check if there is a protocol run that is part of (1), (2) or both, in which an evil party Eve (E) makes Bob sign a bad contract (bad for Bob). If so, use protocol notation to describe this protocol run, as above, and briefly explain why your answer works.

Begin Secret Info: .....

$$E \longrightarrow B : h(\textit{bad contract})$$
$$B \longrightarrow E : N, \{N_B\}_{e_E}$$
$$E \longrightarrow B : N_B$$

Then number N is obtained by decrypting the first message $x = h(\textit{bad contract})$. This is done by $x^{d_B} \mod n_B$, since $(x^{e_B})^{d_B} \mod n_B = x$. The latter equation is typical for RSA. Hence the reaction to x gives a signature of B. Thus $N = [h(\textit{bad contract})]_{d_B}$.

Grading (total 8):	
aspect:	*points*
correct protocol	*4*
correct RSA-property	*4*

End Secret Info .....

2. (**20 points**) (symmetric crypto, hashing, security concepts)
Alice and Bob have a shared symmetric key K_{AB} which they can use for exchanging encrypted messages. They are worried that their messages may become eventually known to someone who intercepts all ciphertexts, for instance because the key K_{AB} is stolen or lost at some stage. They are interested in obtaining higher levels of confidentiality.

They agree to do the following, using a hash function h. For their first message they use K_{AB} as encryption key, for the second message they use $h(K_{AB})$ instead, for the third $h^2(K_{AB}) = h(h(K_{AB}))$, ..., for the $(n+1)$-th message they use $h^n(K_{AB})$ as encryption key.

Briefly explain your answers to the following questions.

(a) (**10 points**) What do Alice and Bob have to keep and what to throw away at stage n? Which property of the hash function h is essential?

Begin Secret Info: .....

At stage $n+1$, after using key $K_n = h^n(K_{AB})$, they should hash this key K_n to get $K_{n+1} = h(K_n) = h^{n+1}(K_{AB})$ for the next message, and throw K_n away. Pre-image resistance is essential of h, so that K_n cannot be re-constructed from $K_{n+1} = h(K_n)$.

Grading (total 10):	
aspect:	*points*
throw away K_n	*3*
keep K_{n+1}	*3*
pre-image resistance	*4*

End Secret Info .....

(b) (**10 points**) With this scheme Alice and Bob wish to achieve both *forward* and *backward* secrecy. Forward secrecy means that if ever key n leaks, messages $n+1$ and higher are still protected. Backward secrecy means the opposite, so if ever key n leaks, messages $n-1$ and lower are still protected. Does this scheme provide Alice and Bob with, forward secrecy, backward secrecy, or both?

Begin Secret Info: .....

They get backward *secrecy, since the keys K_i for $i < n$ are thrown away, and cannot be re-constructed from K_n.*
They DO NOT get forward *secrecy, since the keys $K_{n+i} = h^i(K_n)$ can be computed from K_n.*

Grading (total 10):	
aspect:	*points*
backward yes	*2*
correct explanation	*3*
forward no	*2*
correct explanation	*3*

End Secret Info .....

3. (**32 points**) (public key crypto, security goals)
This exercise considers digital (money) cheques. Assume that with each bank account a public key pair is associated, where only the bank account holder knows the private key. The bank has its own private key.

If person X requests a cheque from his/her bank, the bank returns a signed message of the following form.

$$C_0 = [(X\text{'s account number}, \text{amount})]_{\text{bank's private key}}$$

In doing so the bank takes 'amount' form X's bank account. Person X can now pass the cheque to person Y by giving him/her

$$C_1 = [(Y\text{'s account number}, C_0)]_{X\text{'s private key}}$$

If Y wants, now he/she can pass the cheque to person Z by giving

$$C_2 = [(Z\text{'s account number}, C_1)]_{Y\text{'s private key}}$$

Et cetera.

(a) (**8 points**) Suppose you receive a cheque at stage C_n. What do you have to verify in order to gain certainty about the integrity of this payment to you? Be very precise!

Begin Secret Info: .....

i. *that your bank account is included in C_n*
ii. *that all intermediate cheques C_i carry a valid signature, with a key corresponding to the relevant bank account, so that these intermediate steps involve proper cheque transfers*
iii. *when you follow the path back to the first cheque C_0, that it contains the right amount*
iv. *and also that C_0 carries the bank's signature*

Grading (total 8):	
aspect:	*points*
each	*2*

End Secret Info .....

(b) (**4 points**) What privacy-sensitive information do you learn when you receive a cheque at stage n?

Begin Secret Info: .....
You learn who paid the amount in C_0 to whom in the whole path leading up to stage n.

Grading (total 4):	
aspect:	*points*
who-paid-what-to-whom	*4*

End Secret Info .....

(c) (**6 points**) What does the bank verify when eventually cheque C_n is cashed? What happens when C_n is found correct by the bank?

Begin Secret Info: .....

i. *The bank performs the same checks as described above*
ii. *The amount in cheque C_0 is transferred to the bank account in cheque C_n.*

Grading (total 6):	
aspect:	*points*
each aspect	*3*

End Secret Info .....

(d) (**6 points**) Consider the problem of double (or multiple) spending. Explain briefly:

i. what the problem is

ii. who can detect it at what stage in the chain (assuming that there is no communication between the parties involved)?

iii. whether the perpetrator (*dader*) can be identified, and if so, by whom?

Begin Secret Info: .....

i. If I have cheque C_n, then I can include it in multiple successor cheques, to different people

ii. Only the bank: When cheques are cashed, the bank can see that C_0 is included in multiple paths

iii. Yes. The bank can then see where the path forked, and who started including C_i in multiple successor cheques.

Grading (total 6):	
aspect:	*points*
each aspect	*2*

End Secret Info .....

(e) (**8 points**) Modify the scheme so that every cheque C_k also includes a transaction identifier I_k (*e.g.* a counter) which is fresh in this chain. In order to prevent double spending, what message should I now send to the bank when I receive cheque C_{n+1}, and what answer do I expect? Think/explain carefully!

Begin Secret Info: .....

When I receive cheque C_{n+1} I should take out the identifier I_n of cheque C_n and send the pair C_0, I_n (or C_n, I_n) to the bank, with the question if it has seen I_n before in the chain of C_0. If so, the cheque C_n has already been passed on to someone else. Thus, the bank has to keep a record of all nonces that have occurred in cheques.

Grading (total 8):	
aspect:	*points*
take the identifier from previous cheque	*4*
right message pair to the bank	*2*
bank verification	*2*

End Secret Info .....

4. (**26 points**) (RSA, extended gcd, modes of encryption)
Consider the RSA crypto-system, with primes $p = 5, q = 11$ and public exponent $e = 29$.

(a) (**12 points**) What is the associated private key? Write down all the intermediate steps, including the steps of the extended-greatest-common-divisor calculation.

Begin Secret Info: .....
We have $\phi = (5-1)\cdot(11-1) = 40$, so we seek x, y with $40 \cdot x + 29 \cdot y = 1$.

n	m	rem	div	y	$x - y \cdot div$
40	*29*	*11*	*1*	*8*	$-3 - 8 = -11$
29	*11*	*7*	*2*	*-3*	$2 + 3 \cdot 2 = 8$
11	*7*	*4*	*1*	*2*	$-1 - 2 \cdot 1 = -3$
7	*4*	*3*	*1*	*-1*	$1 + 1 = 2$
4	*3*	*1*	*1*	*1*	$0 - 1 = -1$
3	*1*	*0*	*3*	*0*	*1*

Hence $40 \cdot 8 + 29 \cdot (-11) = 240 - 239 = 1$. *Thus* $d = -11 = 40 - 11 = 29 \mod 40$. *Double check:* $40 \cdot (-21) + 29 \cdot 29 = -840 + 841 = 1$. *So, the associated private key is* $(55, 29)$.

Grading (total 12):	
aspect:	*points*
correct ϕ	*2*
correct goal	*2*
correct outcome	*2*
correct computation	*6*

End Secret Info .....

(b) (**14 points**) Encrypt the message "papa" in ECB mode, with encoding 'a' = 2, 'b' = 3, etc. Explain how you proceed.

Begin Secret Info: .....

$$
\begin{aligned}
\{a\}_{(n,e)} &= \{2\}_{(55,29)} \\
&= 2^{29} \mod 55 \\
&= 2 \cdot 2^{2 \cdot 14} \mod 55 \\
&= 2 \cdot (2^2)^{14} \mod 55 \\
&= 2 \cdot 4^{14} \mod 55 \\
&= 2 \cdot 16^7 \mod 55 \\
&= 2 \cdot 16 \cdot 16^6 \mod 55 \\
&= 32 \cdot (16^2)^3 \mod 55 \\
&= 32 \cdot 256^3 \mod 55 \\
&= 32 \cdot 256 \cdot 256^2 \mod 55 \\
&= 32 \cdot 36 \cdot 36^2 \mod 55 && \textit{since } 256 = 220 + 36 = 4 \cdot 55 + 36 \\
&= 52 \cdot 1296 \mod 55 && \textit{since } 32 \cdot 36 = 1152 = 1100 + 52 = 20 \cdot 55 + 52 \\
&= 52 \cdot 31 \mod 55 && \textit{since } 1296 = 1265 + 31 = 23 \cdot 55 + 31 \\
&= 1612 \mod 55 \\
&= 17 && \textit{since } 1612 = 1595 + 26 = 29 \cdot 55 + 17
\end{aligned}
$$

Since the encoding of the letter p is 17, we get $\{p\}_{(n,e)} = \{2^{29}\}_{(55,29)} = \{[2]_{(55,29)}\}_{(55,29)} = 2$. *Hence the ECB-encryption of "papa" is 17-2-17-2.*

For marking purposes, here is the encryption of 'p' = 17:

$$
\begin{aligned}
\{p\}_{(n,e)} &= \{17\}_{(55,29)} \\
&= 17^{29} \mod 55 \\
&= 17 \cdot (17^2)^{14} \mod 55 \\
&= 17 \cdot 289^{14} \mod 55 \\
&= 17 \cdot 14^{14} \mod 55 && \textit{since } 289 = 275 + 14 = 5 \cdot 55 + 14 \\
&= 17 \cdot (14^2)^7 \mod 55 \\
&= 17 \cdot 196^7 \mod 55 \\
&= 17 \cdot 31^7 \mod 55 && \textit{since } 196 = 165 + 31 = 3 \cdot 55 + 31 \\
&= 17 \cdot 31 \cdot (31^2)^3 \mod 55 \\
&= 32 \cdot 961^3 \mod 55 && \textit{since } 17 \cdot 31 = 527 = 495 + 32 = 9 \cdot 55 + 32 \\
&= 32 \cdot 26^3 \mod 55 && \textit{since } 961 = 935 + 26 = 17 \cdot 55 + 26 \\
&= 32 \cdot 26 \cdot 26^2 \mod 55 \\
&= 7 \cdot 676 \mod 55 && \textit{since } 32 \cdot 26 = 832 = 825 + 7 = 15 \cdot 55 + 7 \\
&= 7 \cdot 16 \mod 55 && \textit{since } 676 = 660 + 16 = 12 \cdot 55 + 16 \\
&= 112 \mod 55 \\
&= 2.
\end{aligned}
$$

Grading (total 14):	
aspect:	*points*
ECB understood	*2*
repeated squaring	*2*
correct encryption of 'a'	*5*
correct encryption of 'p'	*5*

End Secret Info .....

Begin Secret Info: .....
Finally we have a ***toetsmatrix****, relating the original goals of the course to the exercises:*

goal	***exercise***
recognise security goals	*1,2,3*
hash functions	*1,2*
public key crypto	*1,3,4*
secret key crypto	*2*
protocols	*1,2,3*

End Secret Info .....