

# Resit Exam — Security

February 28, 2017. 12:30 – 15:30

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. **(25 points)** This exercise considers a situation where a group of people, called voters, can cast a vote electronically using RSA. It is assumed that each participant  $X$  has a private key  $d_X$  and a corresponding certificate  $C_X$  containing the associated public key  $e_X$  and the participant's name  $X$ , signed by some Certificate Authority.

We consider the following security properties for the voting process.

- (A) **secrecy**: no-one except voter  $V$  should be able to learn what  $V$  voted.  
(B) **single-vote**: no-one should be able to vote more than once.

Votes are sent to a Polling Station  $PS$  that collects votes. It is assumed that connections do not reveal side-information (like IP addresses).

- (a) **(10 points)** Consider the following protocol.

$$V \longrightarrow PS : \{[\text{vote}]_{d_V}, C_V\}_{e_{PS}}$$

Discuss briefly whether goals (A) and (B) are achieved: how or how not.

Now consider the following more complicated version, where voter  $V$  first gets a blind signature on a fresh nonce  $N$ .

$$V \longrightarrow PS : \{(PS|N) \cdot \{r\}_{e_{PS}}, C_V\}_{e_{PS}} \quad \text{where } r \text{ is a random blinding factor,}$$

and  $|$  is concatenation

$$PS \longrightarrow V : \{[PS|N]_{d_{PS}} \cdot r\}_{e_V}$$

$$V \longrightarrow PS : \{\text{vote}, [PS|N]_{d_{PS}}\}_{e_{PS}}$$

- (b) **(5 points)** Explain what  $PS$  does after receiving the first message, in order to generate the signed name-and-nonce  $[PS|N]_{d_{PS}}$  in the second message.
- (c) **(10 points)** Briefly discuss whether goals (A) and (B) are achieved with this second protocol: how or how not.
2. **(22 points)** A typical organization runs various services a user can log in to, using the same account. A way to provide 'single sign-on' to these services is by using so-called 'tickets'. In such a system, a user requests a ticket from a central *ticket granting server* ( $TGS$ ), and then uses this ticket to authenticate to a *service* ( $S_1, S_2, S_3, \dots$ ). In this question, we will use the following symmetric keys:

- $K_{S_i}$  is a secret key shared between the  $TGS$  and service  $S_i$  ( $TGS$  has a list);
- $K_A$  is shared between user Alice and the  $TGS$ ;
- $K_{AS_i}$  is a (session) key distributed to  $A$  and  $S_i$ , freshly generated by the  $TGS$ .

Protocol 1:

$A \longrightarrow TGS : A, S_i$   
 $TGS$  generates a fresh  $K_{AS_i}$   
 $TGS \longrightarrow A : K_A \oplus K_{AS_i}, K_{S_i} \oplus K_{AS_i} (= ticket)$   
 $A \longrightarrow S_i : K_{AS_i}\{A\} (= authenticator), K_{S_i} \oplus K_{AS_i} (= ticket)$

- (a) (5 points) Protocol 1 contains a crucial flaw; show how Alice can obtain the secret key of a service.

Now consider Protocol 2, where we instead use some secure block cipher to encrypt the session key.

Protocol 2:

$A \longrightarrow TGS : A, S_i$   
 $TGS$  generates a fresh  $K_{AS_i}$   
 $TGS \longrightarrow A : K_A\{K_{AS_i}\}, K_{S_i}\{K_{AS_i}\} (= ticket)$   
 $A \longrightarrow S_i : K_{AS_i}\{A\} (= authenticator), K_{S_i}\{K_{AS_i}\} (= ticket)$

After completing the protocol, the service uses the session key to decrypt the *authenticator*. If that works, authentication is considered successful. To prevent replay attacks, assume each service stores every ticket it has seen, and will never accept the same ticket twice.

- (b) (7 points) Assume Alice (A) starts using Protocol 2 to authenticate to service  $S_1$ . Show how an attacker Eve (E) can make Alice authenticate to  $S_2$  instead.  
(c) (4 points) Prevent the problem that is abused in (b).

We started by assuming that the TGS and each user share a secret key (e.g.  $K_A$ , for Alice). However, typically they only share a *password*. This is then used to derive a shared secret key. Assume this is done by computing  $K_A = h(\text{password}_A)$ , where  $h$  is some fast cryptographically secure hash function.

- (d) (6 points) Assume Alice has not chosen a very strong password (e.g. only five characters) and an eavesdropper listens in on a run of Protocol 2. Show how an attacker can retrieve the password using an *offline attack*. Clearly state how the correct password can be recognized.
3. (16 points) Alice and Bob will execute the Diffie-Hellman protocol. The domain parameters are modulus  $p = 101$  and generator  $g = 19$ . Alice uses 39 as her secret number and Bob uses 48 as his secret number.
- (a) (3 points) What is the purpose of the Diffie-Hellman protocol, i.e., what do Alice and Bob want to achieve?
- (b) (3 points) Describe the protocol with messages exchanged and computations executed on each side.

- (c) **(4 points)** The basic Diffie-Hellman protocol is vulnerable to a (wo-)man-in-the-middle attack because Bob cannot verify that the message he receives actually comes from Alice and vice versa. Explain how to extend the protocol such that Bob can be sure that what he receives comes from Alice (we do not require the other way round in this question). Also clearly explain what must be set up for that.
- (d) **(3 points)** Compute Alice's public number that plays a role in the protocol
- (e) **(3 points)** Do the computation Bob will do at the end of the protocol and say what it is that he has computed
4. **(15 points)** Take  $p = 11$  and  $q = 19$  as prime numbers for RSA.
- (a) **(4 points)** Compute the modulus  $n$  and the result  $\varphi(n)$  of applying the Euler function  $\varphi$  to  $n$ .
- (b) **(6 points)** Assume we choose as public exponent  $e = 17$ . Find the corresponding private exponent  $d$ . Describe your computations.

Assume we have found an USB stick with someone's RSA private key on it that was stored as  $p, q$  and  $d$ . We don't know whose key it is and to find out we wish to find the corresponding modulus and public exponent. After doing that we will be able to compare it with public keys we can find on the web.

- (c) **(5 points)** The private key data on the USB stick is  $p = 23$ ,  $q = 19$  and private exponent 317. Explain how to compute the modulus and especially the public exponent and then do the actual computations.
5. **(22 points)** Consider a company A-Corp, which has a database with user records. These records include password data. Furthermore, users can set a password hint themselves. This hint can be shown to users to remind them of their password. A-Corp was hacked and the database was made public, containing the columns `User`, `Password Data` and `Hint`, as shown in the table on the next page. This question is about what could be in the `Password Data` column.

- (a) **(4 points)** After the hack, A-Corp sent an email to its customers. In this email it stated the following:

*"The attackers may have obtained access to your username and encrypted password."*

If this statement is true, what is A-Corp doing wrong in the way they store the password? Explain why that is a problem.

Many users use insecure passwords. Since the database also includes password hints, for some users we can easily guess the password. On the next page, we display an extract of the leaked database. The table contains password data and password hints for a number of users with easily guessable passwords. Next to it is a column that was added by us. It contains the passwords we can guess, based on the password hints.

User	Password Data	Hint	Password guess
Alice	0581c5efa4097c14	numbers 1-6	123456
Bobby	0d136609370da3aa 20f990b0e3f0980b	2*name+1996	bobbybobby1996
Claire96	20f990b0e3f0980b	birth year	1996
Dave	42eaa074b069b227 2b713b4cca277edc	9 down	987654321
ErinB	2fca9b003de39778 2b713b4cca277edc	password + 1	password1
Frank	2fca9b003de39778 2fca9b003de39778	password*2	passwordpassword
Gerald	e5b172997af51dd1 13f43e1a74dd0e98 1433f6b89ca2a82d 08e1b5be844b0abb	XKCD 936	correcthorse- battery staple

Longer passwords give longer encryptions, but the length of the encrypted password is always a multiple of a fixed number (8 bytes in this case).

- (b) (5 points) What mode of encryption is used? Explain your answer.
- (c) (3 points) What is the problem with this mode?
- (d) (10 points) A different company, called Bee Inc., also has a database with user records. Bee Inc. hashes their passwords. Again, we can guess passwords using the password hints:

User	Password Data	Hint	Password guess
Arnold	110edf2294fb8bf4	numbers 123456	123456
Bart	110edf2294fb8bf4	==123456	
Charlie	110edf2294fb8bf4	c'est "123456"	
Denise	e5d8efed9088db0b	double password	passwordpassword
Erica	e5d8efed9088db0b	password twice	
Francine	2fca9b003de39778	the password is password	password
Gabe	2fca9b003de39778	rhymes with assword	
Harry	ecba98cca55eabc2	sixxone	111111
Iris	ecba98cca55eabc2	1*6	
Jessica	ecba98cca55eabc2	sixones	

We see that an important technique for securely storing passwords is not used.

- i. What is the name of this technique?
- ii. How does it work?
- iii. Why is it important to use this technique?