

# Security Protocols and Application — Final Exam

Philippe Oechslin and Serge Vaudenay

28.7.2019

Family Name: .....

Given Name: .....

SCIPER: .....

- duration: 3h00
- no document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will not answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

## 1 Finding Malicious Domain Parameters

Let  $n = 2^e d + 1$  where  $e$  and  $d$  are positive integers and  $d$  is odd. Let  $a$  be an integer such that  $1 \leq a < n$ . We say that  $n$  is a *pseudoprime to base  $a$*  if and only if

$$a^d \bmod n = 1 \quad \text{or} \quad \exists i \in \{0, 1, \dots, e-1\} \quad (a^{2^i d} + 1) \bmod n = 0$$

We also define

$$S(n) = \{a \in \{1, 2, \dots, n-1\}; n \text{ is a pseudoprime to base } a\}$$

It was proven that  $\#S(n) \leq \frac{\phi(n)}{2^{m-1}}$ , where  $m$  is the number of pairwise different prime factors of  $n$ .

**Q.1** Explain the acronyms CDH, TLS, PAKE, ECDH.

**Q.2** Explain what is a safe prime, a smooth number, and by which efficient algorithm we can compute discrete logarithms in a smooth ordered cyclic group.

**Q.3** Explain what are Diffie-Hellman parameters and which mathematical properties we should normally verify on those parameters.



**Q.4** Compute  $S(33)$ .



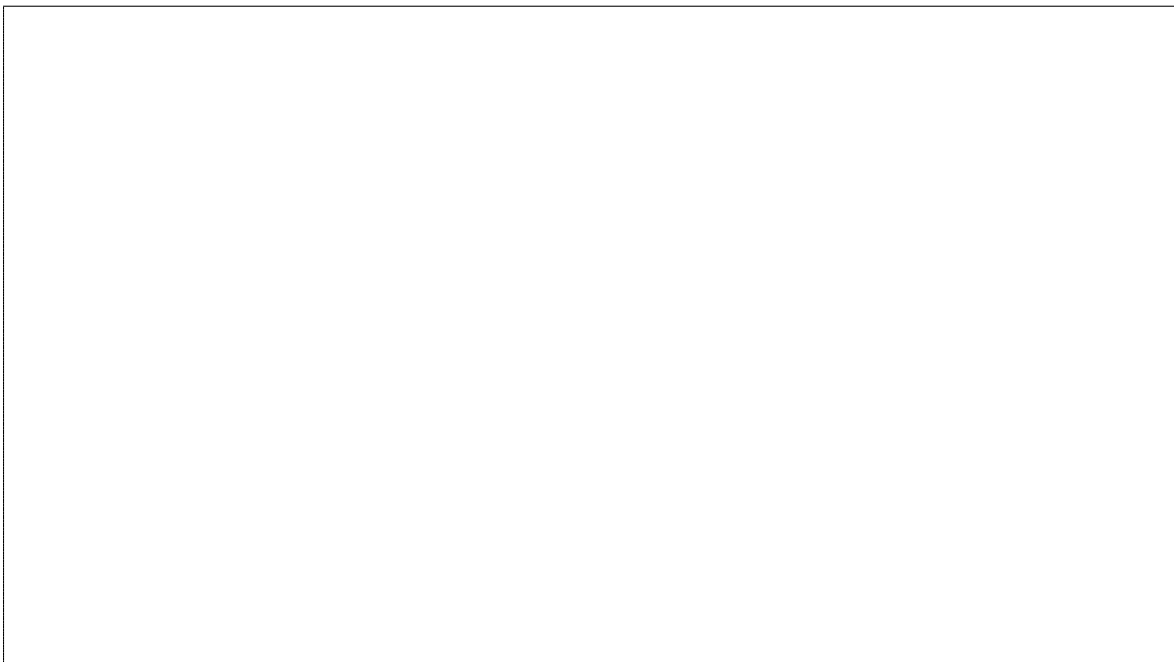
**Q.5** Depending on  $\#S(n)$  and the number  $t$  of iterations, what is the probability of the Miller-Rabin primality test to be wrong when  $n$  is a composite number?



**Q.6** Explain the following quote:

“The primality test that OpenSSL uses [...] performs  $t$  rounds of random-base Miller-Rabin testing, where  $t$  is determined by the bit-size of  $p$  and  $q$ . Since  $p$  and  $q$  are 1 024 and 1 023 bits respectively,  $t = 3$  rounds of Miller-Rabin are performed, at least in versions prior to OpenSSL 1.1.0i (released 14th August 2018). From version 1.1.0i onwards,  $t$  was increased to 5, with the aim of achieving 128 bits of security instead of 80 bits.”

How was  $t$  computed?



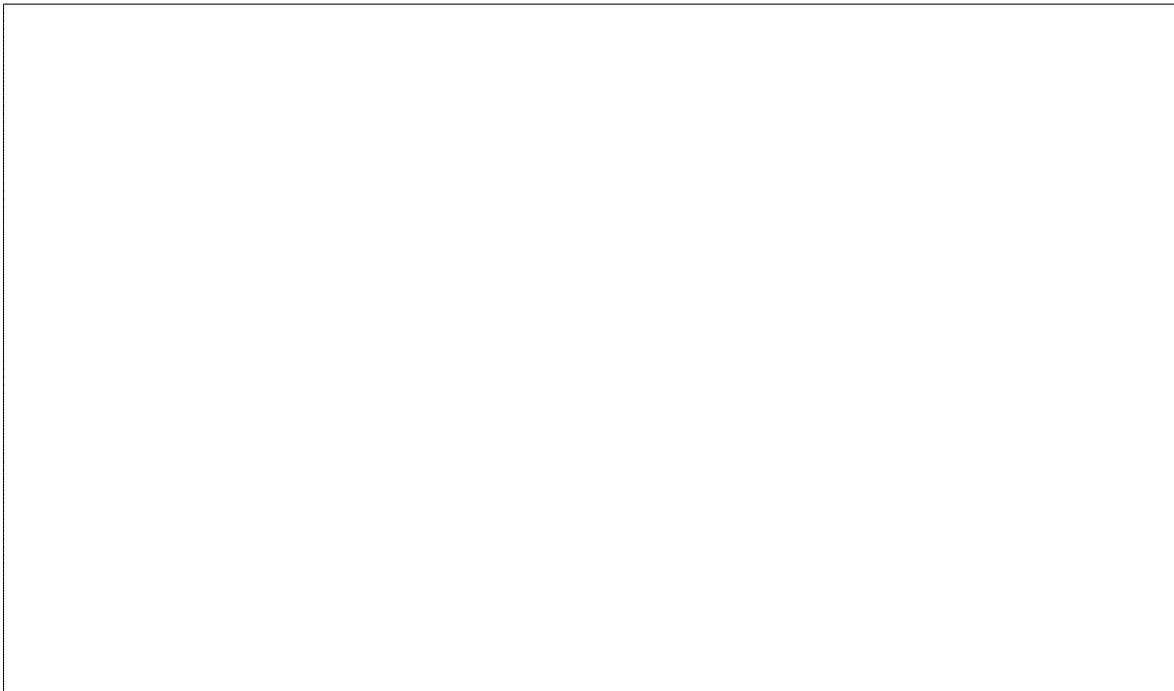
**Q.7** The quote of the previous question continues as follows:

“For the DH parameter set [there is] a probability of approximately  $1/2^8$  of being declared prime by a single round of Miller-Rabin testing. Hence this DH parameter set will be accepted by DH\_check as being valid with probability approximately  $2^{-24}$  (and the lower probability of  $2^{-40}$  since version 1.1.0i of OpenSSL).”

Why is this not a contradiction with the previous quote?



**Q.8** Is this attack a threat to the Diffie-Hellman protocol? If not, when could it be a threat?





Family Name: .....

Given Name: .....

SCIPER: .....

## 2 NSEC5 and Zone Enumeration

### 2.1 NSEC and NSEC3

**Q.1** NSEC and NSEC3 have a weakness that NSEC5 aims to eliminate. Answer the following 3 questions:

- What is this weakness ?
- What advantage does NSEC3 give regarding this weakness ?
- Why is this not sufficient ?

## 2.2 NSEC5 properties

In NSEC5, PSR stands for Primary-Secondary-Resolver systems. Explain the following properties for a PSR system:

**Q.2** Completeness:

**Q.3** Soundness:

**Q.4** Privacy in NSEC5 is defined using f-zero knowledge proofs (f-zk proofs). Explain what the f means and what it is in NSEC5



### 2.3 NSEC5 signatures

NSEC5 uses two key pairs, the primary and secondary keys. They are used for two different types of signatures. Let's call them primary signatures and secondary signatures.

**Q.5** How many primary and how many secondary signatures must the primary resolver generate when setting up a zone with  $N$  host names ?

**Q.6** How many primary and how many secondary signatures must the secondary server generate when answering a request ?

**Q.7** How many primary and how many secondary signature verifications must the resolver carry out to verify the answer ?

## 2.4 NSEC5 attacks

**Q.8** Looking at the answers of the last two questions, describe a method for creating a denial of service on the secondary server. What is the cost for the attacker ?

**Q.9** Describe a method that allows an attacker to know the number of names that exist in a domain

**Q.10** If a secondary server is compromised by an attacker, can the attacker

- a) know all existing names of the domain ?
- b) fake a positive response for a name that is not in the domain?
- c) fake a negative response for a name that is in the domain?

Justify

**Q.11** What attack could an attacker carry out if he was in possession of the private key of a secondary server?



**Q.12** There is a very small probability that a fully functioning secondary server can not generate a proof of non-existence of a name. In what situation does this happen ?

