ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Family Name: .....

First Name: .....

Section: .....

Cryptography and Security Course

(Crypto Part)

Final Exam

February 17th, 2005

Duration: 1 hour 45 minutes

This document consists of 12 pages.

Instructions

Electronic devices are not allowed.

Answers must be written on the exercises sheet.

This exam contains 2 *independent* exercises.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered. Potential errors in these sheets are part of the exam.

You have to put your full name on *each* page.

I A Hybrid Cryptosystem Using RSA and DES

The boss of a small company wants to secure all digital exchanges among the computers of the employees. As he is stingy, he does not want to hire a cryptographer and decides to set up a complete system by himself (he borrowed a textbook in the library). More precisely, he wants to use RSA and DES in order to build a hybrid cryptosystem. Such a scheme assumes that each employee of the company has a private key and that the associated public key is known to all the other employees. Figure 1 illustrates an example of the set up of a secure communication between Alice and Bob (two employees of the company). The principle is first, to establish a DES secret key (the *session* key) to be used in a session, second, to encrypt every message of the session with this session key. We denote by (n_A, e_A) and (n_B, e_B) the RSA public keys of Alice and Bob respectively, and by d_A and d_B the corresponding private keys. The session key will simply be denoted k. As the boss of the company wants to achieve a high level of security, he decides to use 2048-bit RSA moduli.

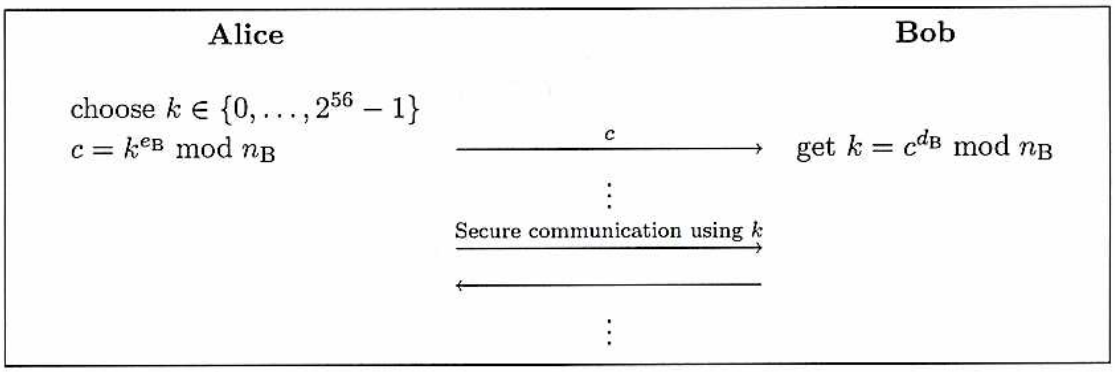


Figure 1: Alice and Bob using the hybrid cryptosystem to secure their communications

1. What are the sizes of the primes p and q? Give a reason why Bob could want to choose a *small* public exponent e_B.

2. Bob chooses $e_B = 3$. Does this scheme provide good security in this case? Why?
   Hint: Look at the size of k^{e_B}.

Bob now chooses $e_B = 2^{16} + 1$. Suppose that Eve (another employee of the company) can eavesdrop the communication and thus learn the value of c.

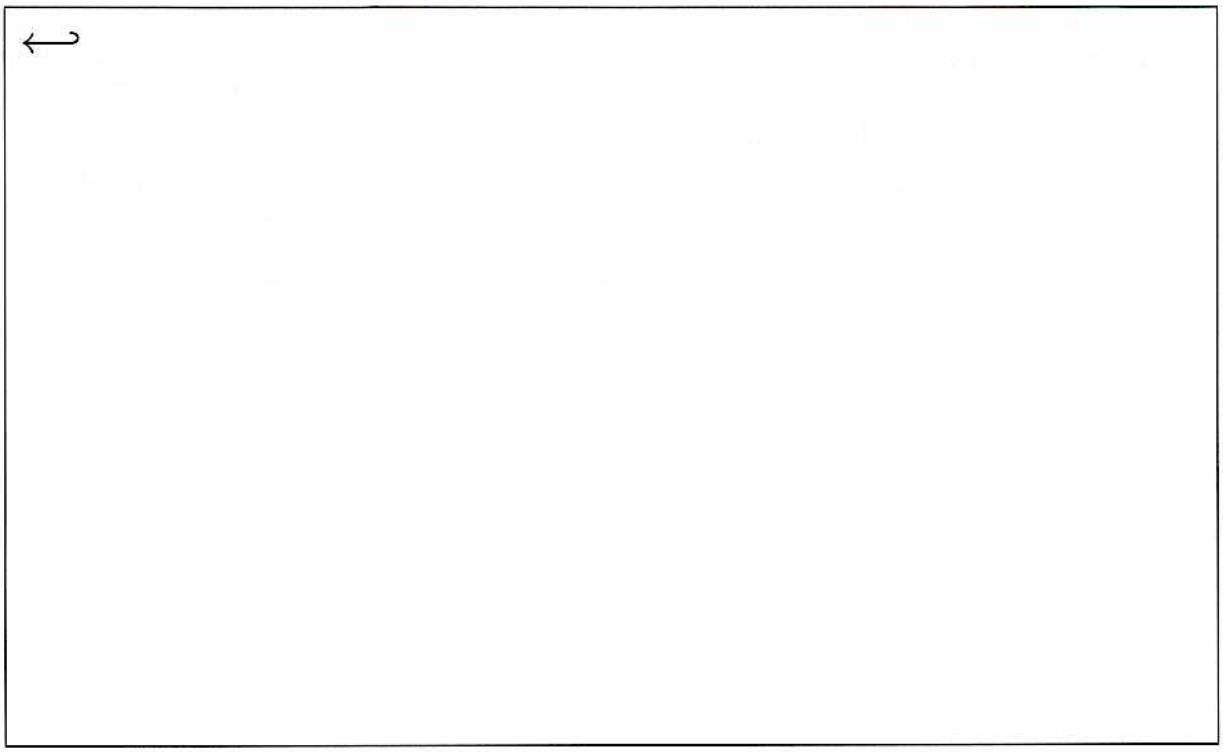
3. Give a brute force algorithm that would (in principle) allow Eve to recover k. What is its complexity? Can it display any wrong key (i.e., a key different from k)?

Suppose now that the DES key k chosen by Alice (considered as an integer of 56 bits) can be written as $k = k_1 \cdot k_2$ where k_1 and k_2 are both integers of 28 bits.

4. Eve decides to store in a table the value of $(k_1, k_1^e \bmod n)$ for every possible value of k_1. Explain how she can mount a kind of meet-in-the-middle attack (using this table) in order to recover k.
   Hint: Express $k_1^{e_B} \bmod n$ in terms of c,k_2,e_B, and n, and exploit this relation.

5. What is the number of modular exponentiations needed to compute the table? What is the size of the table? Once the table is computed, how many modular exponentiations are required to recover the key?

↪



In order to reduce the memory requirement, Eve decides to use a hash function. Consequently, instead of storing $(k_1, k_1^{e_B} \bmod n)$, she now stores $(k_1, h(k_1^{e_B} \bmod n))$, where $h : \{0,1\}^* \to \{0,1\}^N$ is a cryptographic hash function.

6. What is the size of this new table if the hash function that Eve decides to use is MD5? How many collision(s) should she expect?

In order to thwart the attack, the boss (who is a real geek) suggests to only use prime numbers for the DES keys, so that it is not possible to find two number to write k as $k_1 \cdot k_2$ (where k_1 and k_2 are 28 bits long).

7. Compute the approximate number of DES keys that satisfy this condition. What are the time and space complexities of a typical time-memory tradeoff against this scheme?

8. Obviously, the scheme is not very well designed. What could be done in order to obtain a better scheme?

II A (Weak) Identification Scheme

An identification scheme is an interactive protocol in which a prover wants to convince a verifier that he knows some private information. It can be used, for instance, in access control. Let s and t be some given security parameters (e.g. $s = 1024$ bits and $t = 160$ bits). We assume that the prover and the verifier have set up some public parameters, that the prover (only) has a private key, and that the verifier has the public key of the prover. Those values are set up as follows:

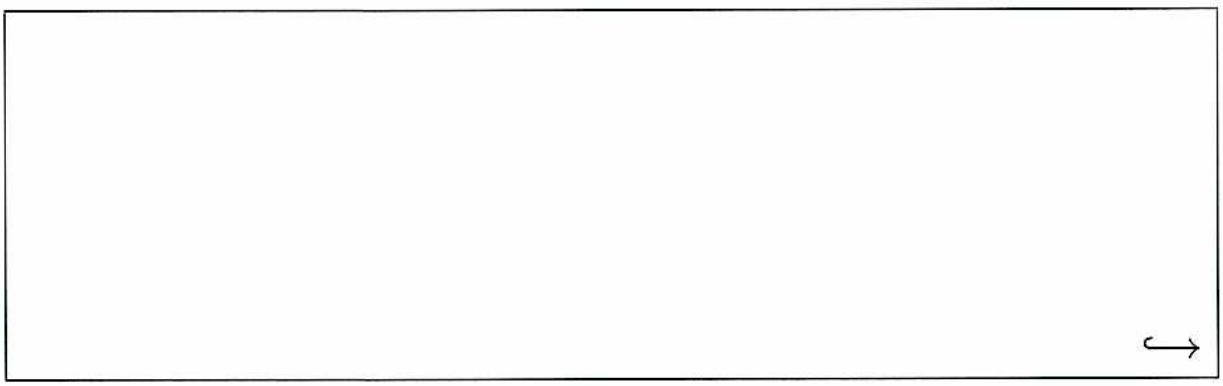
- **Public Parameters**: A large integer n of size s, an element $\gamma \in \mathbb{Z}_n^*$, a prime q of size t
- **Private Key**: An integer $a \in [2, q-1]$
- **Public Key**: $\alpha = \gamma^a \bmod n$

Following the identification scheme on Figure 2, the prover convinces the verifier that he knows the private key without disclosing it.

Prover		**Verifier**
choose $k \in [0, q-1]$ uniformly at random		
$\mu = \gamma^k \bmod n$	$\xrightarrow{\mu}$	
check $e \in [0, q-1]$	$\xleftarrow{e}$	choose $e \in [0, q-1]$
$x = k - a \cdot e$		
Compute r, ℓ such that $x = \ell \cdot q + r$ with $0 \leq r < q$		
$\lambda = \gamma^\ell \bmod n$	$\xrightarrow{r,\lambda}$	check $\mu = \gamma^r \alpha^e \lambda^q \bmod n$ and $r \in [0, q-1]$

Figure 2: The RDSA identification scheme

1.1. What is the complexity of the generation of the public parameters? What is the complexity of the generation of the private/public key pair? Cite all the algorithms that are needed in both cases.



↩

1.2. What is the total bit length of the messages exchanged between the prover and the verifier in the worst case?

1.3. Show that the verification process should work, i.e., show that $\mu = \gamma^r \alpha^e \lambda^q \bmod n$ when the prover and the verifier follow the protocol specifications.

Obviously, no information about the prover's private information should leak, not even to the verifier. We will now see that this scheme is flawed as a *malicious* verifier can recover some of the bits of the private key.

2.1. A malicious verifier chooses $e = 0$. Compute ℓ, r, λ in this case. Does the malicious verifier recover any information about the secret key in this case?

2.2. A malicious verifier chooses $e = 1$. Depending on k and a, compute ℓ, r, λ in this case.

2.3. Deduce from the previous question that the verifier learns one bit of the secret key (with high probability) after a few runs of the protocol.

We denote by $\lfloor x \rfloor$ the greatest integer less than or equal to x. We will see that the verifier can recover *several* bits of the private key.

3.1. Show that $\ell = \lfloor \frac{-a \cdot e}{q} \rfloor + \epsilon$, where $\epsilon = 0$ or 1.

3.2. Deduce from the last question that the size of ℓ is approximatively equal to the size of e. Show that the verifier can exploit this to easily recover ℓ from λ when e is short.

3.3. Show that $\left|a - \frac{-\ell \cdot q}{e}\right| < \frac{2q}{e}$.

3.4. Denoting δ the size of e, show that the last inequality allows the verifier to recover $\delta - 1$ bits of the private key by selecting a short e.