Université Libre de Bruxelles
Faculté des Sciences

August 19, 2021

INFO-F-405 Introduction to cryptography

This assessment may be done with the use of personal notes, slides, books, web pages, etc. However, the assessment is strictly personal and you are not allowed to communicate with other students or other people during this assessment.

Optionally, you may skip one sub-question of your choice. If you decide to do so, you have to write clearly and explicitly which sub-question you skip, e.g., "I am skipping sub-question 1A." The skipped question will then not be graded and the total of the points will be rescaled accordingly.

Question 1

(5/20) Practical aspects of cryptography Alice has recently landed in Belgium for a one-year Erasmus at ULB/VUB. Just like any other student, she has been granted access to the university supercomputer, *Hydra*, to run experiments for her Master's Thesis.

We will assume that the connection protocol with Hydra consists in a Diffie-Hellman key exchange followed by the symmetric encryption of the communications. Upon the first connection, Alice's computer requests Hydra's long-term public key and stores it for the subsequent connections. Then for each connection, Alice's computer generates an ephemeral key pair and sends the public key to Hydra. Finally, each party generates a secret key from its private key and the public key received from the other party.

Carelessly, Alice connects to Hydra with this protocol from her student residence.

Sub-question 1A. Let us assume that an attacker (Charles) has control over the network of Alice's student residence. Charles knows that many students will try to connect to Hydra. How can Charles intercept and read all communications between Alice's computer and Hydra in the clear without being noticed?

Sub-question 1B. The next day, Charles does not intercept connections to Hydra anymore. What will Alice notice when she connects to Hydra? Why?

Sub-question 1C. In SSH, the first time you connect to a host, you get prompted with a *fingerprint* and asked whether that *fingerprint* matches what you expect from the remote host you are trying to reach. What is this fingerprint? To what is it applied? What kind of attack does it prevent?

Sub-question 1D. Let us assume that there is a trusted authority such as the Belgian government in the loop. Could the ULB/VUB prevent attacks such as the one Charles is able to perform via this trusted authority ?

Question 2

(5/20) Secret-key cryptography An automated teller machine (ATM) distributes cash to the customers of a bank. It is connected to the bank's server, which is in charge of authorizing and tracking transactions. The bank and the ATM share a k-bit secret key K. When authorizing, the bank sends the ATM a triplet $(u, m, \texttt{tag})$ allowing user u to receive an amount m in cash, and tag is a n-bit message authentication code (MAC) computed under key K over the first two components of the triplet. (This is of course a very simplified view for the sake of this question.)

Sub-question 2A. What is the bank trying to protect against, assuming an adversary who has full access to the network connecting the bank and the ATM?

Sub-question 2B. Let us assume for now that the bank and the ATM use a secure MAC function. What is the minimum key length k that is required to have a security strength of 128 bits, and why?

Sub-question 2C. Let us further assume that the ATM processes not more than one triplet per second and that the secret key K is securely renewed every 24 hours. What is the minimum length n of the MAC such that the probability of fraudulent transaction is less than 10^{-12} per day, and why?

Sub-question 2D. Let us now assume that $n = k$ and that the MAC function is constructed as follows:

$$\text{MAC}_K(\text{message}) = \text{hash}(\text{message}) \oplus K,$$

where $\text{hash}(\cdot)$ is a secure n-bit hash function and $\oplus$ denotes the bitwise mod-2 addition (or XOR). Is this MAC function secure? If so, please justify briefly. If not, please describe an attack (preferably in the light of the EU-CMA game).

Sub-question 2E. Let us consider another MAC function. We assume again that $n = k$, but the MAC function is now constructed as follows:

$$\text{MAC}_K(\text{message}) = \text{hash}(K_1 \| \text{message}) \oplus K,$$

where $\text{hash}(\cdot)$ is a secure n-bit hash function, $\|$ denotes the concatenation and $K = K_1 \| K_2$ with $|K_1| = |K_2| = k/2$. What is the security strength of this MAC function, and why?

Question 3

(4/20) Hashing Various sub-questions.

Sub-question 3A. For a given hash function, does second preimage resistance imply collision resistance, or vice-versa, and why?

Sub-question 3B. A friend of yours designed his/her own hash function CATSHA320, with 320 bits of output. It is an iterated hash function that cuts the input message into blocks of 192 bits, processes them sequentially and has a chaining value of 224 bits. Without knowing more about this hash function, what is its expected collision resistance (in number of attempts), and why?

Sub-questions 3C and 3D. Consider SHAKE128, a standard sponge function with permutation f, capacity $c = 256$ bits and rate $r = 1344$ bits. Let us assume that it is used in an application that hashes only small input messages that fit in a single bock of r bits (even after padding[1]), and where the output size is $n = 256$ bits.

- Sub-question 3C. First, write symbolically (i.e., as a mathematical expression) the output as a function of the input message for the restricted use case of this application.
- Sub-question 3D. Then, knowing that f and its inverse f^{-1} can both be evaluated efficiently, is this hash function preimage-resistant? If so, please justify briefly. If not, please describe an attack.

[1] The details of the padding have little relevance in this question, so we may assume that the message M is simply padded as $M||10^*$, i.e., with a single bit 1 followed by the minimum number of bits 0 such that the padded message is r-bit long.

Question 4

(6/20) Public-key cryptography Alice frequently needs to upload files to Bob's server, and they use RSA and hybrid encryption to keep their files confidential. Please find below the specifications of their protocol, similar to RSA-KEM, to which we added a few mistakes.

One-time setup

1. Bob sets $e = 3$.
2. Bob randomly chooses a private prime number p of 256 bits. He checks that $\gcd(e, p) = 1$; otherwise, he repeats with a new prime p.
3. Similarly, Bob randomly chooses another private prime number q of 256 bits. He checks that $\gcd(e, q) = 1$ and $p \neq q$; otherwise, he repeats with a new prime q.
4. Bob computes $d = e^{-1} \bmod \phi(pq)$ and keeps d private.
5. Bob computes $n = pq$ and sends (e, n) to Alice. They check together that Alice received Bob's public key correctly.

When Alice needs to upload a file F

6. Alice chooses a random string m of 512 bits.
7. Alice computes the 160 bits of output of $\mathrm{SHA1}(m)$ and interprets them as the integer k with $0 \leq k \leq 2^{160} - 1$.
8. Alice computes $c = k^e \bmod n$, and she encrypts her file F with a good symmetric encryption scheme using the secret key k resulting in ciphertext G.
9. Alice uploads (c, G).

When Bob receives an encrypted file (c, G)

10. Bob recovers k by computing $k = c^d \bmod n$.
11. Bob chooses a random string m of 512 bits and computes $k' = \mathrm{SHA1}(m)$. If $k' \neq k$, it outputs an error message and aborts.
12. Bob decrypts G with the same good symmetric encryption scheme, using k as secret key, to recover F.
13. Bob stores F on the server.

Sub-question 4A. What is the size in bits of Bob's modulus n that will result from the setup? Does that give sufficient security in 2021? If so, please justify briefly. If not, please recommend which size the primes should have to achieve about 128 bits of security.

Sub-question 4B. On line 4, what is $\phi(pq)$? Can you give a simpler expression? What would happen if $\phi(pq)$ was part of Bob's public key?

Sub-question 4C. There is a mistake in the one-time setup procedure that can cause the computation to fail sometimes. What is it? How can you fix it, and why? Before it is fixed, what is approximately the probability that this procedure fails?

Sub-question 4D. There is a mistake in the way RSA is used that causes a major security issue. What is it? How can you fix it, and why?

Sub-question 4E. There is a silly mistake in Bob's procedure that causes it to fail almost always. What is it? How can you fix it, and why?

Sub-question 4F. Can someone other than Alice upload a file onto Bob's server? If so, please sketch briefly how to modify the protocol so that only Alice can upload a file. Otherwise, please explain briefly how the current protocol achieve this restriction.