Université Libre de Bruxelles
Faculté des Sciences

January 26, 2022

INFO-F-405: Introduction to cryptography

This assessment may be done with the use of written or printed personal notes, slides, books, 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 question of your choice. If you decide to do so, you have to write clearly and explicitly which question you skip. The skipped question will then not be graded and the total of the points will be rescaled accordingly.

Principles

Question 1 (8%): Suppose that (Gen, E, D) is an IND-CPA secure symmetric-key encryption scheme with diversification, i.e., there is no known way of winning the IND-CPA game other than with a probability negligibly close to $\frac{1}{2}$ or with over-astronomical resources. Let the key space be $\{0,1\}^n$ with $n \geq 128$, the plaintext space to be arbitrary and the diversifier space be the set of non-negative integers $\mathbb{N}$. Would the following scheme E' be IND-CPA secure?

$$E'_k(d,m) = E_k(d,m) \parallel E_k(d+2^{32}, m)$$

If it is IND-CPA secure, please justify briefly. If it is *not* IND-CPA secure, please specify how an adversary can win the IND-CPA game, and how much effort is necessary.

Secret-key cryptography

A company is selling videogames. Next to the game on a physical media, they offer the users to register their copy online. Registered users can then enjoy playing over the network and benefit from other advantages. We assume that each copy of a videogame has a unique serial number S and comes with an authentication code C computed as a function of S and of K, a secret key known only to the company and used for all the games they sell. When registering, the user then has to give the pair (S, C) of his/her copy to the registration server.

The company built their authentication function on a block cipher $\mathcal{B}$ with the following signature :

$$\mathcal{B} : \{0,1\}^{128} \times \{0,1\}^{256} \to \{0,1\}^{128} : \mathcal{B}_K(X) \mapsto Y$$

where the X is the input block, Y is the output block and K is the secret key. They also chose a security parameter $\alpha \in [0, 128]$.

For each copy of the game, the company

- chooses a new serial number $S \in \{0,1\}^{128}$,
- computes the authentication code C by first applying the block cipher to S then keeping only the first α bits of the output, i.e., $C = \lfloor \mathcal{B}_K(S) \rfloor_\alpha$, and

- prints (S, C) on a sticker and attaches it to the videogame.

Question 2 (8%): How does the registration server check that the pair (S, C) is legitimate?

Question 3 (8%): A hacker would like to generate a valid pair (S, C) with a serial number different from that of his own copy. What is the name of this type of attack? If he makes random guesses, how many attempts do you expect him to submit before finding a valid one depending on the security parameter α?

Question 4 (8%): If we assume that the best known attack on $\mathcal{B}$ that retrieves the secret key is an exhaustive search, what is easier between finding a valid code by random-guessing and retrieving the secret key? Why?

In order to increase the security of the authentication, the company decides to use bigger parameters, more specifically, a serial number of 256 bits, $S \in \{0, 1\}^{256}$. As the input is now larger than the block size of $\mathcal{B}$, the company defines a new mode of operation inspired from the *Electronic Code Book* (ECB) mode: C is obtained by

- applying the block cipher to S_1, the first 128 bits of S, keeping only the first α bits of the output,
- applying the block cipher to S_2, the last 128 bits of S, keeping only the first α bits of the output, and
- concatenating the results of these last two steps, i.e., $C = \lfloor \mathcal{B}_K(S_1) \rfloor_\alpha \parallel \lfloor \mathcal{B}_K(S_2) \rfloor_\alpha$.

Question 5 (8%): Instead of increasing the security, this variant completely breaks the system's security. Assuming he has seen one or more valid valid pairs (S_i, C_i), explain how a hacker could generate a valid (S, C) with a different serial number without knowing the secret key K. If the hacker has seen N valid pairs (S_i, C_i), how many new valid pairs can he create? (You can make assumptions on the way the serial numbers S_i are generated.)

Hashing

A team is setting up a new blockchain. They are designing a new hash function H for it and considering to use the sponge construction.

Question 6 (8%): For their application, they estimate that preimage resistance is more important than collision resistance. They would like to have at least 160 bits of preimage resistance and at least 100 bits of collision resistance. What is the minimum number of bits that the hash function must output? Please justify! Also, if they use the sponge construction, what capacity can they choose, and what is the smallest permutation they can use?

We can view this blockchain abstractly as a set of transactions $\{T_i\}$. For a transaction T_i to be valid, it has to come with a solution x_i to the following a puzzle: The bit string value x_i must be chosen such that the digest $H(T_i \| x_i)$ starts with 0^k, i.e., the first k bits are all 0. (Here, we do not fix k as it will typically vary with the life of the blockchain.)

Question 7 (8%): Modeling H as a random oracle, what is the probability, for a fixed T and randomly-chosen x, that the output of $\mathcal{RO}(T \| x)$ starts with 0^k? What about after t attempts with different candidates x, approximately? As a function of k, how many attempts do we need before we can solve this puzzle, approximately? If we want the puzzle to take a time equivalent to about 2^{30} attempts, what is the corresponding value k?

Public-key cryptography

Consider the following variant of the ElGamal encryption scheme. Let p be a large prime and g is a generator of $\mathbb{Z}_p^*$. Let XOF be a secure extendable output function. Let $a \in \mathbb{Z}_{p-1}$ be Alice's private key and $A = g^a \bmod p$ her public key.

Encryption of $m \in \{0,1\}^*$ with Alice's public key A:

- Randomly choose a secret integer $k \in [1, p-2]$
- Compute

$$\begin{aligned} K &= g^k \bmod p \\ T &= A^k \bmod p \\ c &= m \oplus \mathsf{XOF}(T) \end{aligned}$$

- Send the ciphertext (K, c) to Alice

Decryption of (K, c) by Alice with her private key a:

$$\begin{aligned} T' &= K^a \bmod p \\ m &= c \oplus \mathsf{XOF}(T') \end{aligned}$$

Note that "$\oplus$" denotes the bitwise modulo-2 addition between two bit strings. We assume that $\mathsf{XOF}(\cdot)$ returns as many output bits as the length of the string it is added to.

Question 8 (6%): Please explain why the decryption procedure correctly recovers the plaintext m.

Question 9 (6%): Consider an adversary attempting a known plaintext attack. If it observes a known plaintext-ciphertext pair $(m, (K, c))$, can it recover the value T that was used during the encryption? Please justify.

Question 10 (8%): If for a given encryption, k is not kept secret, what are the consequences (if any) for the security of this ElGamal variant? Please justify.

Question 11 (8%): If for the encryption of two plaintexts m_1 and m_2 to Alice, the same value k is used, what are the consequences (if any) for the security of this ElGamal variant? Please justify.

Question 12 (8%): Bob wants to send many messages to Alice, but he is a bit lazy. So, instead of choosing an independent random k every time, he randomly chooses a secret integer k_0 once for all and increments it for every new encryption, i.e., he uses $k = k_0 + i \bmod (p-1)$ for the encryption of message number i. What are the consequences (if any) for the security of this ElGamal variant? Please justify.

Question 13 (8%): Let $\mathcal{E}$ be an elliptic curve over GF(p). Let $G \in \mathcal{E}$ be a generator of order q. Rewrite this encryption scheme to use the elliptic curve $\mathcal{E}$.

Portfolio

IND-CPA (chosen plaintext, chosen diversifier)

A scheme $\mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec})$ is **IND-CPA-secure** if no adversary can win the following game for more than a negligible advantage.

1. Challenger generates a key (pair) $k \leftarrow \text{Gen}()$
2. Adversary queries Enc_k with (d, m) of his choice
3. Adversary chooses d and two plaintexts $m_0, m_1 \in M$ with $|m_0| = |m_1|$
4. Challenger randomly chooses $b \leftarrow_R \{0, 1\}$, encrypts m_b and sends $c = \text{Enc}_k(d, m_b)$ to the adversary
5. Adversary queries Enc_k with (d, m) of his choice
6. Adversary guesses b' which plaintext was encrypted
7. Adversary wins if $b' = b$ (Advantage: $\epsilon = \left|\Pr[\text{win}] - \frac{1}{2}\right|$.)

The adversary **must** respect that d **is a nonce**! The values of d used in steps 2, 3 and 5 must all be different.

Sponge construction

The sponge construction is a mode on top of a permutation. It calls a b-bit permutation f, with $b = r + c$, i.e., r bits of *rate* and c bits of *capacity*. A sponge function is a concrete instance with a given f, r, c and implements a mapping from $M \in \{0, 1\}^*$ to $\{0, 1\}^\infty$ (truncated at an arbitrary length):

- $s \leftarrow 0^b$
- $M||10^*1$ is cut into r-bit blocks
- For each M_i do (absorbing phase)
  - $s \leftarrow s \oplus (M_i||0^c)$
  - $s \leftarrow f(s)$
- As long as output is needed do (squeezing phase)
  - Output the first r bits of s
  - $s \leftarrow f(s)$

The differentiating advantage of a random sponge from a random oracle is at most $t^2/2^{c+1}$, with t the time complexity in number of calls to f.

Random oracle

A random oracle is a non-deterministic algorithm that returns uniformly and independently distributed random bits for any input value. The same output bits are returned when the same value is input.

Original ElGamal encryption

Encryption of $m \in \mathbb{Z}_p^*$ with Alice's public key A:

- Randomly choose a secret integer $k \in [1, p-2]$
- Compute

$$K = g^k \bmod p$$
$$c = mA^k \bmod p$$

- Send the ciphertext (K, c) to Alice

Decryption of (K, c) by Alice with her private key a:

$$m = K^{-a} c \bmod p$$