UNIVERSITÉ LIBRE DE BRUXELLES
FACULTÉ DES SCIENCES

January 25, 2023

INFO-F-405: Introduction to cryptography

This assessment may be done with the use of written or printed personal notes, slides, books, etc., but without any electronic devices (e.g., laptop, tablet, phone) and without internet access. Also, the assessment is strictly personal and you are not allowed to communicate with other students or other people during this assessment.

This exam is made of 11 questions and includes a portfolio at the end. Please ensure that we can split your answers to questions 1-7 from those of questions 8-11 by using separate sheets of paper.

Principles

■ **Question 1** (10%): With your own words, please explain what are confidentiality and authenticity in cryptography. Then, does an encryption scheme provide authenticity and why (not)? And finally, does an authentication scheme provide confidentiality and why (not)?

■ **Question 2** (12%): 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 $\mathbb{N}$. Let H be a cryptographic hash function. Would the following schemes $E^{(1)}$, $E^{(2)}$ and $E^{(3)}$ be IND-CPA secure?

$$E_k^{(1)}(d,m) = E_k(H(d),m)$$
$$E_k^{(2)}(d,m) = E_k(d,m) \,\|\, H(m)$$
$$E_k^{(3)}(d,m) = E_k(H(k\|m),m) \,\|\, H(k\|m)$$

For each scheme, if it is IND-CPA secure, please justify briefly, and specify if there are properties that H needs to satisfy for this. Otherwise, please specify how an adversary can win the IND-CPA game, and how much effort is necessary.

■ **Question 3** (12%): A hypothetical encryption scheme, called RC404, has been standardized with only the following security claim: *"RC404 resists against key recovery attacks with a security strength of 128 bits in the known plaintext model."*

For each of the following attacks, please state whether it contradicts the claim and why (not).

a. A method that recovers the key with a success probability of one in a billionth ($\approx 2^{-30}$), requires 2^{100} known plaintext-ciphertext pairs and runs in about 2^{100} times the execution time of RC404.

b. A method that recovers with certainty the first byte of the plaintext (although not the next ones) from the ciphertext only, with a running time of a few hours on a modern PC.

c. A method that recovers the key with certainty after seeing the ciphertexts corresponding to the $10! \approx 2^{22}$ plaintexts composed of the 10 letters "ALGORITHMS" in any order, with a running time of a few hours on a modern PC.

What criticism can you make on RC404 and on its security claim?

Secret-key cryptography

Rijndael / AES

■ **Question 4** (6%): What is a block cipher? How do you use it in an encryption or authentication scheme, in general? Even if the best attack against a block cipher is the exhaustive key search, is it enough to guarantee the security of an encryption/authentification scheme that uses this block cipher? Why (not)?

■ **Question 5** (10%): Consider two executions of the block cipher AES-128 with the same secret key. The first one has input block X and the second input block X'. If X and X' differ only in *two bytes*, at least how many bytes will differ after two rounds? And at most? Note that the position of the two bytes is not specified, and so you may specify it to be able to reach the minimum or the maximum.

Hashing

SHA-512/256 is a hash function standardized by the NIST. In a nutshell, it works like SHA-512, except that the output is truncated to 256 bits. In more details, it uses the Merkle-Damgård construction on top of SHA-512's compression function. Hence, it has a chaining value of 512 bits and a message block size of 1024 bits. The output consists of the first 256 bits of the final chaining value. At the time of this writing, there has been no weakness found in its compression function.

■ **Question 6** (10%): In the light of these specifications of SHA-512/256,

a. what is its preimage resistance?

b. what is its second preimage resistance?

c. what is its collision resistance?

d. is producing an internal collision a good strategy to find a collision in this hash function, and why (not)?

■ **Question 7** (10%): For authenticating transactions, we would like to use a message authentication code $\mathrm{MAC}_K(\text{message})$, but we hesitate between different options. Here K is a 128-bit key and the message can be of any size. For each option, please state whether it is secure and briefly justify your answer.

a. $\text{MAC}_K(\text{message}) = \text{SHA-512}(K \| \text{message})$

b. $\text{MAC}_K(\text{message}) = \text{SHA-512/256}(K \| \text{message})$

c. $\text{MAC}_K(\text{message}) = \text{HMAC-SHA-512}_K(\text{message})$

d. $\text{MAC}_K(\text{message}) = \text{HMAC-SHA-512/256}_K(\text{message})$

Which one is the most efficient and secure option in your opinion?

Public-key cryptography

A tripartite key-exchange

Anya, Bert and Chloe want to communicate using AES but to do this, they first need to agree upon a common secret that they could use to derive the AES secret key. They want to use the Diffie-Hellman key exchange but this protocol allows to share a secret between two parties only. The goal of this exercise is to create a variant of Diffie-Hellman in which three people can obtain the same secret.

Here, we fix a prime p and a generator g of the group $\mathbb{Z}_p^*$. Anya's, Bert's and Chloe's private/public key pairs $(\mathsf{SK}, \mathsf{PK})$ are $(a, A = g^a), (b, B = g^b), (c, C = g^c)$ with a, b, c in $[1, \ldots, p-2]$. The reduction modulo p is implicit as we work in $\mathbb{Z}_p^*$.

■ **Question 8** (8%): Propose a protocol based on Diffie-Hellman that would allow all three people to share the common secret g^{abc}. *Hint:* The parties can first work in pairs. What is the cost of your protocol compared to the original Diffie-Hellman protocol ?

Generic group notation

Let G be a group of order $|G| = q$ and g is a generator of this group.

■ **Question 9** (6%): Rewrite the two-party Diffie-Hellman algorithm in the generic group notation so that it can be applied, e.g., to elliptic curves. What is the expression of the common secret with the generic group notation?

A one-round tripartite Diffie-Hellman using a pairing

We now want to improve our protocol and improve its efficiency. In order to do this, we will use a special kind of function (often used in elliptic curve cryptography) called a *pairing*.

Let E be an elliptic curve, G the group of points on this curve and P a point that is a generator of the group G. Let $q = |G|$ be the order of G, i.e., the number of elements in G. Anya's, Bert's and Chloe's private/public key pairs $(\mathsf{SK}, \mathsf{PK})$ are now $(a, A = [a]P), (b, B = [b]P), (c, C = [c]P)$ where a, b, c are integers in $\mathbb{Z}_q$.

Finally, we define a map $F : \mathbb{Z}_q \times G \times G \to \mathbb{Z}_q^*$ with the following properties. If P is a generator of the group G, then for any points R and Q and any integer x in $\mathbb{Z}_q$, we have

- $F(1, P, P) = u$, where u is some fixed and known element in $\mathbb{Z}_q^*$,
- $F(x, R, Q) = F(1, R, Q)^x$,
- $F(1, [y]R, Q) = F(1, R, Q)^y$,
- $F(1, R, [z]Q) = F(1, R, Q)^z$.

Such a map F is what we call a *pairing*. (*Note:* you can safely assume the pairing F to be already defined and use it abstractly.)

In essence, these properties allow us to compute $F(x, Y, Z)$ for any point $Y = [y]P$ and $Z = [z]P$, and the result can be expressed as a power of the known integer u. Note that every point can be expressed as a multiple of P since we assume P is a generator.

■ **Question 10** (6%): Compute the value of $F(1, A, B)$ and $F(a, P, P)$.

■ **Question 11** (10%): Propose a protocol in which each party obtain a common secret by applying the pairing F only once. What is the expression of this common secret?

Portfolio

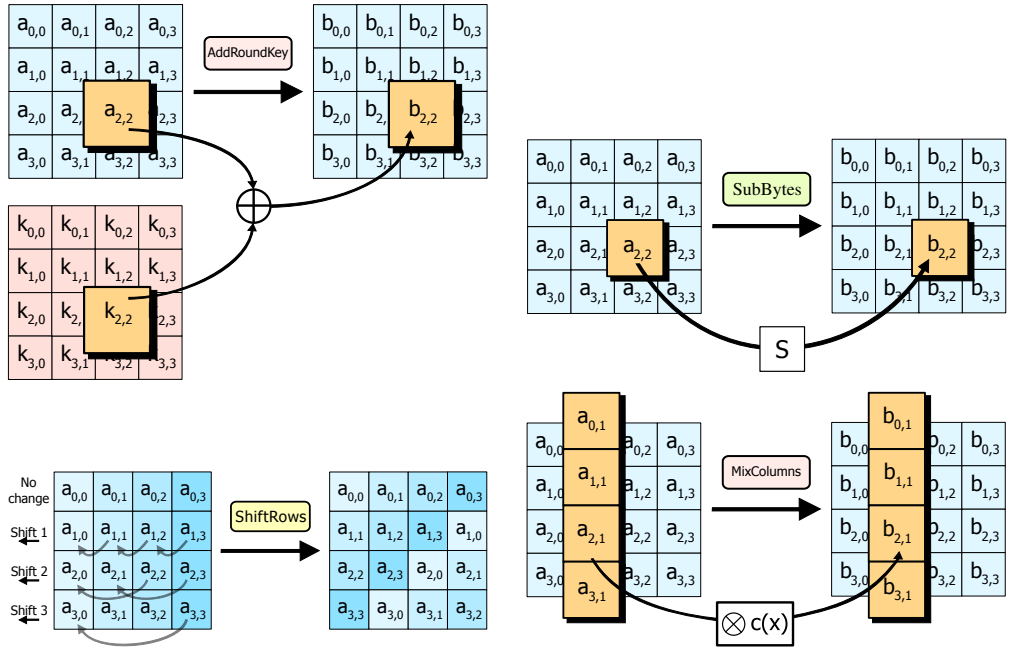
IND-CPA (chosen plaintext, chosen diversifier)

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: $\epsilon = \left|\Pr[\text{win}] - \frac{1}{2}\right|$.)

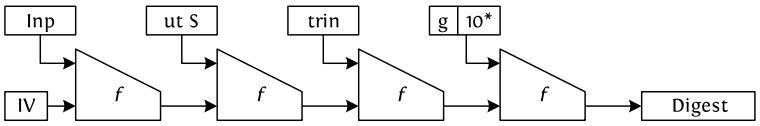
The adversary **must** respect that d **is a nonce**! The values of d used in steps 2, 3 and 5 must all be different.

The step mappings inside Rijndael/AES



MDS property: If N bytes change at the input of MixColumns and as a result M bytes change at its output, then either $N = M = 0$ or $N + M \geq 5$.

The Merkle-Damgård construction



HMAC (simplified)

$$\text{HMAC-}H_K(\text{message}) = H(K \| H(K \| \text{message}))$$

The Diffie-Hellman protocol

In the multiplicative group setting, the public parameters are the group $G = \mathbb{Z}_p^*$ (with p prime) and a generator g.
The keypairs $(\mathsf{SK}, \mathsf{PK})$ of Alice and Bob are respectively $(a, A = g^a)$ and $(b, B = g^b)$, for a, b in $\mathbb{Z}_p^*$.
The key exchange is then described by the following algorithm :

1. Alice sends A to Bob ; Bob sends B to Alice
2. Alice computes B^a ; Bob computes A^b
3. Alice outputs B^a as the shared secret ; Bob outputs A^b as the shared secret