Université Libre de Bruxelles 2024-2025
Faculté des Sciences

INFO-F-405: Introduction to cryptography

4. Public-key techniques

Going public

Exercise 1

The characters in the text below make use of public-key cryptography. They each prepared a pair of public/private keys and they all know everyone else's public key. Complete the following sentences and scratch the invalid alternatives (as much as possible without looking at the slides or at your notes).

To send a confidential file to Xavier, Yasmina _____ it with her/Xavier's _____ key. When she receives it, she _____ it with his/her _____ key.

Elena sends to her husband Fabrice the list of items to buy at the grocery store. The list is not confidential, but she wants to avoid their daughter Gabi from adding chocolates or other extra items to it. To do so, she _____ the list with her/Fabrice's/Gabi's _____ key before she sends it. When he receives it, Fabrice _____ the _____ with his/Elena's/Gabi's _____ key.

Exercise 2

Thelma and Louise recently met for the first time and would like to do business together. After their meeting, because of the COVID-19, they are forced to stay at home and can communicate only by phone or via the Internet. They wish to start discussing their business project, but want to do so confidentially. Hopefully, they already exchanged their phone numbers and email addresses, and are both knowledgeable about cryptography. However, their project is highly sensitive and they need to protect their conversation against even active adversaries, that is, people who can afford to tamper with Internet connections. We nevertheless assume that the adversaries are not going to cut their communication lines, as this would be too obvious and as they would otherwise miss all opportunities of spying on the two business women.

Please explain how Thelma and Louise can set up a secure communication channel using public-key cryptography.

Exercise 3

Alice has recently arrived 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.

a. 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?

b. The next day, Charles does not intercept connections to Hydra anymore. What will Alice notice when she connects to Hydra? Why?

c. 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?

d. 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 ?

Exercise 4

Lets say (Gen, E, D) is a semantically secure public-key encryption system. Could the algorithm E be deterministic?

Rivest-Shamir-Adleman

Exercise 5

RSA Encryption

a. Consider the key generation algorithm of RSA. Let $p = 11$ and $q = 17$ be two given prime numbers. What is the corresponding RSA modulus n? Assuming that the public exponent is $e = 3$, what is the corresponding private exponent d in the private key?

b. Consider the encryption algorithm of the RSA "textbook encryption" scheme. Assuming that the plaintext is $m = 15$ and the public key (n, e) is as in question a, compute the resulting ciphertext c.

c. Consider the decryption algorithm of the RSA "textbook encryption" scheme. Using a computer, check that the decryption of the ciphertext c from the previous question indeed yields the original plaintext $m = 15$.

d. Consider the figure depicting RSA-OAEP in the slides and assume a 16-bit RSA modulus (hence $n = 16$ in the notation of that figure), 8 bits of randomness (hence $k0 = 8$) and 4 bits of redundancy (hence $k1 = 4$).

   The functions $G, H : \{0,1\}^* \to \{0,1\}^*$ used in RSA-OAEP should be instantiated with extendable output functions (or hash functions with the MGF1 mode). For simplicity, however, we will assume in this question that they take a very simple form. With $\kappa = k0 = n - k0 = 8$, G and H are defined as follows:

   $$G : b_0 \dots b_{\kappa-1} \mapsto b_{\kappa-1} \dots b_0 \qquad \text{and} \qquad H : b_0 \dots b_{\kappa-1} \mapsto \overline{b_0} \dots \overline{b_{\kappa-1}},$$

   where $\overline{b_i} = 1 - b_i$ for all $0 \leq i < \kappa$.

   Let the plaintext be $m = 1010$ and the randomness be $r = 1110\,0101$. Compute the resulting input $(X \| Y)$ of the exponentiation and give its numerical value in decimal.

e. Let's say we want to set up an RSA system for k users. How many primes do we have to generate? What security issues we might create if we use less prime numbers?

Exercise 6

In the RSA "textbook signature" scheme, a message m is signed in the following way under the private key (n, d):

$$m \to (m, s) = (m, m^d \bmod n),$$

Is there any way of creating a forgery?

Exercise 7

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 $k' = \text{SHA1}(m)$. 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.

Questions

a. 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.

b. 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?

c. 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?

d. 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?

e. 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?

f. 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.

Discrete logarithm problem in $\mathbb{Z}_p^*$

Exercise 8

The discrete logarithm problem is based on the multiplicative group $(\mathbb{Z}_p^*, \times)$. What happens if we would use the additive group $(\mathbb{Z}_p, +)$ instead? *Hint:* Start with writing out the problem in that group.

Exercise 9

ElGamal encryption

a. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$, $p = 23$. The order of $\mathbb{G}$ is $|\mathbb{G}| = 11$. Let $g = 4$ be a generator of $\mathbb{G}$. Consider the key generation algorithm of the ElGamal encryption scheme (and of all DLP-based schemes). Assuming that the private key is $a = 3$, compute the corresponding public key A.

b. Consider the encryption algorithm of the ElGamal encryption scheme. Assume that the plaintext $m = 3$, the random exponent (ephemeral private key) chosen by the encryption algorithm is $k = 2$, and $\mathbb{G}$, g, and A are as in question a. Compute the resulting ciphertext (K, c).

c. Consider the decryption algorithm of the ElGamal encryption scheme. Assume that the ciphertext is $(K, c) = (6, 22)$ and $\mathbb{G}$, g, a, and A are as in question a. Compute the resulting plaintext m.

Exercise 10

What is the additive group notation for $\mathbb{Z}_p^*$? Given a prime number p, let the group $\mathbb{G}$ be the set of integers between 1 and $p - 1$ (like $\mathbb{Z}_p^*$) equipped with the group operation $+$ be the multiplication modulo p. (Yes, an "addition" symbol to denote a multiplication in this case!) The neutral element of $\mathbb{G}$ is denoted O, i.e., $O = 1$, and the inverse of an element A is denoted $-A$.

The *scalar multiplication* refers to the process of repeating the group operation onto itself a given number of times. We denote $[n]A$ the group element obtained by repeating the group operation on n copies of A. So, $[0]A = O$, $[1]A = A$, $[2]A = A + A$, $[3]A = A + A + A$, etc, and $[-1]A = -A$, $[-2]A = (-A) + (-A)$, etc.

For instance, let $p = 23$ as above, let A and B be group elements $A = 4$ and $B = 6$. Then $A + B = O$ as $4 \times 6 \equiv 1 \pmod{23}$. So A and B are each other's inverse, i.e., $A = -B$. Also, $[5]A = 12$ since $4^5 \equiv 1024 \equiv 12 \pmod{23}$. And $[-1]A = -A = 6$.

ElGamal encryption in additive group notation Assuming a group $\mathbb{G}$, a generator $G \in \mathbb{G}$, rewrite the key generation process to generate Alice's public-private key pair in additive group notation. Then, given a plaintext message $M \in \mathbb{G}$, rewrite the ElGamal encryption scheme in the same notation. Finally, rewrite the decryption and explain why it correctly recovers the plaintext.

Exercise 11

Let's say Alice and Bob both choose a planet that they want to visit (in our solar system). They want to check if they choose the same planet without giving out their choice. Let's say Alice choses the planet a and Bob the planet b. Alice and Bob agree on the following scheme:

- They publicly choose a prime p and generator g of $G = \mathbb{Z}_q^*$
- Alice chooses random x and y in $\mathbb{Z}_p$ and sends to Bob $(A_0, A_1, A_2) = (g^x, g^y, g^{xy+a})$
- Bob choose random r and s in $\mathbb{Z}_p$ and sends back to Alice $(B_1, B_2) = (A_1^r \times g^s, \left(\frac{A_2}{g^b}\right)^r \times A_0^s)$

How Alice can check if she had chose the same planet as Bob?

Exercise 12

Schnorr Signature

a. In contrast to RSA with full-domain hashing, the Schnorr signature scheme is probabilistic, i.e., different executions of the signing algorithm on the same input (a, m) results in different signatures $\sigma = (s, e)$. How many different Schnorr signatures can exist for a single message $m \in \{0, 1\}^*$?

b. Let q the size of the group, i.e., $q = p - 1$ if g is a generator of $\mathbb{Z}_p^*$. Assume that during the generation of some Schnorr signature $\sigma = (s, e)$ the random exponent (or ephemeral private key) $k \in \mathbb{Z}_q$ becomes known to an adversary $\mathcal{A}$. How can $\mathcal{A}$ use σ and k in order to forge a Schnorr signature σ' for any message m' of its choice?

c. Assume that the same random exponent (or ephemeral private key) $k \in \mathbb{Z}_q$ used in the signing algorithm of the Schnorr scheme was used multiple times. More precisely, we have two signatures $\sigma_1 = (s_1, e_1)$ and $\sigma_2 = (s_2, e_2)$ on two different messages $m_1 \neq m_2$ that were generated using the same exponent $k_1 = k_2 = k$. How can an adversary $\mathcal{A}$ use σ_1 and σ_2 in order to forge a Schnorr signature σ' for any message m' of its choice?

d. To make k random and unique per signature, the signer generates a random $k \in \mathbb{Z}_q$ upon the first signature, and then increments it ($k \leftarrow k + 1$) for each new signature. Is that secure? If so, please justify. If not, what is the problem and how can you fix it?

e. Propose two ways to pick k in a secure way. *Hint:* One probabilistic and one deterministic.

Elliptic curve cryptography

Exercise 13

Elliptic curve in $\mathrm{GF}(11)$**.** In this exercise, all the arithmetic operations are understood modulo 11, although not explicitly written. Let E be the curve over $(\mathrm{GF}(11))^2$ satisfying the Weierstrass equation

$$y^2 = x^3 - 3x + 7.$$

a) List the points on the curve. To do this, we advise the following steps:

   - Build a table of squares modulo 11. For each value y, compute y^2. Sort the table per value y^2. How many squares modulo 11 are there? For a given value y^2, how many corresponding values y can there be? When more than 1, how do these values relate to each other?
   - For each value x, compute $x^3 - 3x + 7$. Using the table of squares modulo 11, look up the possible value(s) of y (if any) that satisfy $y^2 = x^3 - 3x + 7$.

b) How many points does E have? What is the largest prime-order group we can get? What is the cofactor?

c) Which point has order 1? Which point has order 2? (Hint: think about the elliptic curves over the reals.)

Exercise 14

ElGamal signature with elliptic curves The purpose of this exercise is to translate the ElGamal signature scheme as expressed with modular exponentiation into an elliptic curve-based scheme.

First, we recall the original scheme here. Given a prime p and a generator g over $\mathbb{Z}_p^*$, we proceed as follows.

- **Signature** of message $m \in Z_2^*$ by Alice with her private key a:
    - Compute $h = \mathrm{hash}(m)$
    - Choose randomly an integer $k \in [1, p-2]$
    - Compute $r = g^k \bmod p$
    - Compute $s = k^{-1}(h - ar) \bmod (p-1)$
        - If $s = 0$, restart with a new k
    - Send (r, s) along with m

- **Verification** of signature (r, s) on m with Alice's public key $A = g^a \bmod p$:
    - Compute $h = \mathrm{hash}(m)$
    - Check $A^r r^s \overset{?}{\equiv} g^h \pmod{p}$

Now, let E be an elliptic curve over $\mathrm{GF}(p)$ and let $G \in E$ be a generator of prime order q. Rewrite the scheme above by replacing operations in $\mathbb{Z}_p^*$ with operations in E. (Hint: to compute the equivalent of r, first compute $R = [k]G$ then let r be the x-coordinate of R.)

What is the equivalent of operations modulo $p - 1$?

Exercise 15

A company that installs and maintains an open-source operating system is creating an automatic update mechanism for its customers. Regularly, the company publishes an *update pack* containing the latest changes to be made to the operating system, and each computer connected to the Internet can fetch it. The company would like to guarantee the integrity of these update packs so as to avoid the installation of malicious software on their customers' computers. Fortunately, confidentiality is not required, as it is open-source software.

In the following, we describe a protocol that the company uses to sign the update packs and for the customers to check that the update packs are genuine. However, we voluntarily added some mistakes (including omissions) to its definition. Some of them are functional, that is, they prevent the protocol from working correctly, while others introduce security flaws.

Please *list all the mistakes* you find, and for each, *justify* why it is incorrect and *propose a correction.*

Let $\mathcal{E}$ be a standard elliptic curve over $\mathrm{GF}(p)$ (for some prime p) that is used by the company and its customers. They also agreed on a base point $G \in \mathcal{E}$ with prime order q, i.e., $[q]G$ is the neutral element. Note that the uppercase letters refer to points on $\mathcal{E}$, while lowercase letters are integers, and UP is a string of bits that represents an update pack.

One-time setup at the company:

1. The company secretly generates its secret key c randomly and uniformly in $[1 \dots p-1]$.
2. The company computes its public key $C = [c]G$, and publishes it via some PKI process. (This aspect is out of scope of this question. In the sequel, we assume that all the public keys are correctly authenticated and can be trusted.)
3. The company secretly generates its secret value n_C randomly and uniformly in $[1 \dots p-1]$.

Company's procedure to sign and post an update pack UP

1. Compute $k = \mathrm{hash}(n_C)$.
2. Compute $R = [k]G$.
3. Compute $e = \mathrm{hash}(R||\texttt{UP})$.
4. Compute $s = k + ec \bmod p$.
5. Post $(\texttt{UP}, e, s)$ on the company's update repository.

Customer's procedure to retrieve and install an update pack

1. Retrieve $(\texttt{UP}, e, s)$ from the company's update repository.
2. Compute $C = [c]G$.
3. Compute $R' = [s]G - [e]C$.
4. Compute $e' = \mathrm{hash}(R'||\texttt{UP})$.
5. The customer installs UP.

Exercise 16

Projective coordinates. Instead of representing points on the curve with (x, y), called *affine coordinates*, one can use an alternate representation using three coordinates $(X : Y : Z)$ called *projective coordinates*.

When $Z \neq 0$, a point in projective coordinates $(X : Y : Z)$ represents (x, y) in affine coordinates with $x = XZ^{-1}$ and $y = YZ^{-1}$. The projective coordinates are *redundant*: For any $\lambda \neq 0$, the projective coordinates $(X : Y : Z)$ and $(\lambda X : \lambda Y : \lambda Z)$ represent the same point.

In projective coordinates, the Weierstrass equation becomes

$$Y^2 Z = X^3 + aXZ^2 + bZ^3 \tag{1}$$

a) How can we represent the point at infinity O in projective coordinates? Check that it satisfies the Weierstrass equation above. (Hint: intuitively, the point at infinity is the vertical direction. Over the reals, it can be viewed as having affine coordinates $(0, \pm\infty)$. In the projective plane, points at infinity have $z = 0$ by definition.)

b) In affine coordinates, the addition of two points $P + Q$ in the general case, i.e., assuming $P \neq Q \neq -P$ and $P \neq O \neq Q$, works as follows. We have $(x_P, y_P) + (x_Q, y_Q) = (x, y)$ with

$$\begin{aligned} s &= \frac{y_P - y_Q}{x_P - x_Q} \\ x &= s^2 - x_P - x_Q \\ y &= s(x_P - x) - y_P \end{aligned}$$

Given the projective coordinates of P and Q, say, $(X_P : Y_P : Z_P)$ and $(X_Q : Y_Q : Z_Q)$, write the result of the addition in projective coordinates $(X : Y : Z)$. As the projective coordinates are redundant, the value Z can be freely chosen; you may for instance fix $Z = 1$.

c) The projective coordinates are interesting in practical implementations for their higher efficiency: The operations on the points can be expressed *without inversions*, which are otherwise costly compared to additions, subtractions and multiplications. Rewrite the addition of points above, but using only additions, subtractions and multiplications (no inversions). (Hint: set Z so as to cancel any inversions.)