HAI709I : Fondements cryptographiques de la sécurité, Université de Montpellier, 2024

23/09/2024. Lecture 3.

1 Polynomials in modular arithmetic.

Proposition 1 (Proven last week in the class)**.** *Let*

$$L(x) = c_0 + c_1 x + \ldots + c_d x^d$$

be a polynomial of degree d with integer coefficients. Assume that for some integer a

$$L(a) = 0 \mod n.$$

Then there exists a polynomial $R(x)$ of degree $d-1$ with integer coefficients such that

$$L(x) = (x-a)R(x) \mod n$$

Theorem 1. *Let n be a prime number and $c_0, \ldots, c_d$ be be integer numbers. Then the polynomial*

$$L(x) = c_0 + c_1 x + \ldots + c_d x^d \mod n$$

cannot have more than d roots in $\{0, 1, \ldots, n-1\}$ (unless all c_i are equal to zero $\mod 0$).

Proof. Assume that $x_1, x_2, \ldots, x_d, x_{d+1}$ are roots of $L(x)$, i.e., for all i

$$L(x_i) = 0 \mod n,$$

and $x_i \neq x_j \mod n$ for $i \not\equiv j$. Since x_1 is a root of $L(x)$, we can apply Proposition 1 and conclude that

$$L(x) = (x - x_1)L_1(x) \mod n$$

for some polynomial with integer coefficients $L_1(x)$ of degree less than d. For each $j = 2, \ldots, d+1$ we have $L(x_j) = 0 \mod n$. Hence,

$$(x_j - x_1)L_1(x_j) \mod n.$$

As $x_j - x_i \neq 0 \mod n$ and n is a prime number, we conclude that $L_1(x_j) = 0 \mod n$. In other words, $x_2, \ldots, x_{d+1}$ are roots of $L_1(x)$.

Now we use the fact that x_2 is a root of $L_1(x)$ and apply again apply Proposition 1. We conclude that

$$L_1(x) = (x - x_2)L_2(x) \mod n$$

for some polynomial with integer coefficients $L_2(x)$, and degree of $L_2(x)$ is at most $d-2$. We observe that $x_3, \ldots, x_{d+1}$ are roots $L_2(x)$. By repeating this argument d times we conclude that

$$L(x) = (x - x_1) \cdot (x - x_2) \cdot \ldots \cdot (x - x_d) \cdot L_d(x) \mod n,$$

where $L_d(x)$ is a polynomial of degree zero (i.e., a constant). Thus, for some integer number a we have

$$L(x) = a(x - x_1) \cdot (x - x_2) \cdot \ldots \cdot (x - x_d) \mod n.$$

Observe that $a \neq 0 \mod n$ unless all coefficients c_j of $L(x)$ are equal to zero modulo n. Now we substitute x_{d+1} in $L(x)$ and obtain

$$L(x_{d+1}) = a(x_{d+1} - x_1) \cdot (x_{d+1} - x_2) \cdot \ldots \cdot (x_{d+1} - x_d) \mod n$$

As n is a prime number, a product of non-zero factor cannot result in the value 0 modulo n. We come to a contradiction with the assumption that x_{d+1} is another root of $L(x)$. □

Theorem 2. *Let n be a prime number, $x_0, \ldots, x_d$ be integer numbers (pairwise different modulo n) and let $y_0, \ldots, y_d$ be any integer numbers. Then there exists a unique polynomial*

$$L(x) = c_0 + c_1 x + \ldots + c_d x^d$$

of degree at most d with coefficients $c_i \in \{0, 1, \ldots, n-1\}$ such that $L(x_i) = y_i$ for $i = 0, \ldots, d-1$.

Proof. First, we prove the uniqueness. Assume that there exist two different polynomials of degree at most d (with integer coefficients) $L(x)$ and $R(x)$ such that

$$L(x_i) = R(x_i) = y_i,\ i = 0, \ldots, d.$$

Then, the difference $L(x) - R(x)$ is a polynomial of degree at most d with $(d+1)$ roots $x_0, \ldots, x_d$ (in the arithmetic modulo n). This contradicts Theorem 1.

Now we prove the existence of the required polynomial $L(x)$. For each $i = 0, \ldots, d$ we denote

$$L_i(x) = (x - x_1) \cdot (x - x_2) \cdot \ldots \cdot (x - x_{i-1}) \cdot (x - x_{i+1}) \cdot \ldots \cdot (x - x_d)$$

(the polynomial defined as the product of all $(x - x_k)$ for all k except for i). Observe that $L_i(x_k) = 0 \mod n$ for all $k \neq i$ and $\alpha_i := L_i(x_i) \neq 0$. We know that for a prime number n, for every integer $\alpha_i \neq 0 \mod n$ there exists another integer number β_i such that $\alpha_i \cdot \beta_i = 1 \mod n$. Let

$$\hat{L}_i(x) := \beta_i L_i(x).$$

So we have a family of polynomials with integer coefficients $\hat{L}_i(x)$ that are equal to 1 at x_i and equal to 0 in all other points x_k (for $k \neq i$). (As usual, all computations are done modulo n.)

Now we can define the required polynomial $L(x)$ as the sum

$$L(x) := y_0 \cdot \hat{L}_0(x) + y_1 \cdot \hat{L}_1(x) + \ldots + y_d \cdot \hat{L}_d(x).$$

We have obtained a polynomial fo degree at most d such that $L(x_i) = y_i$ for all i. □

2 Shamir secret sharing scheme

We continue the discussion fo secret sharing. We need to distribute a *secret* k among a group of m parties (participants of the secret sharing scheme). Let us fix t (the *threshold*, an integer number between 1 and n). We require that every group of at least t participants could get the secret and every groups of less than t participants should not get *any* information about the secret

Let $\mathcal{K}$ be the space of all potential secrets. In this section $\mathcal{K} = \mathbb{Z}/n\mathbb{Z}$ for some prime number n. We assume that the secret is chosen in $\mathcal{K}$ at random, with the uniform distribution. A secret sharing scheme is a randomized algorithm (*Dealer*) that samples for each $k \in \mathcal{K}$ a probability distribution $p_k(s_1, \ldots, s_m)$

$$\mathrm{Prob}^{(k)}[S_1 = s_1, \ldots, S_m = s_m] = p_k(s_1, \ldots, s_m),$$

the distribution of random *shares* compatible with the key k. These distributions must respect the following two conditions.

(I) For every group of t participants $\{i_1, \ldots, i_t\}$, the random variables $\langle S_{i_1}, \ldots, S_{i_t}\rangle$ contain enough information to reconstruct the secret key k. This means that for every vector of value $(s_{i_1}, \ldots, s_{i_t})$ there can be only one secret $k \in \mathcal{K}$ such that

$$\mathrm{Prob}^{(k)}[S_{i_1} = s_{i_1}, \ldots, S_{i_t} = s_{i_t}] > 0.$$

(II) For every group of $\ell < t$ participants $\{i_1, \ldots, i_\ell\}$, the random variables $\langle S_{i_1}, \ldots, S_{i_\ell}\rangle$ contain *no* information on k. This means that for all $k \in \mathcal{K}$ the restrictions of the distribution

$$\mathrm{Prob}^{(k)}[S_1 = s_1, \ldots, S_n = s_n]$$

on the coordinates $i_1, \ldots, i_\ell$ are identical[1].

Shamir's scheme of secret sharing is defined as follows. First of all, we fix a prime number n (at must be greater than the number of participants m) and some integer numbers x_i for $i = 1, \ldots, m$ (these numbers must be pairwise different modulo n); this information is public. To share a secret k, Dealer chooses at random $c_1, \ldots, c_{t-1}$ in $\mathbb{Z}/n(Z)$ and defines

$$L(x) = k + c_1 x + \ldots c_{t-1} x^t.$$

Every i-th participants receives a private key $y_i := L(x_i)$.

(I) Given t values $y_{i_1}, \ldots, y_{i_t}$, one can apply Theorem 2 and reconstruct the polynomial $L(x)$ such that

$$L(x_{i_1}) = y_{i_1} \mod n, \ \ldots \ , L(x_{i_t}) = y_{i_t} \mod n,$$

and compute the secret key $k = L(0) \mod n$. Thus, every group of (at least) t participants can compute the secret key.

(II) if only $(t-1)$ (or even less) values y_{i_1} are known, than any value of k is compatible with the given conditions

$$L(x_{i_s}) = y_{i_s} \mod n.$$

Moreover, all values of $k \in \mathbb{Z}/n\mathbb{Z}$ are equiprobable. Thus, every group of at most $(t-1)$ participants holds no information on the secret key.

Observe that in Shamir's scheme, for every participant the space of possible private key coincides with the space of secret keys $\mathcal{K} = \mathbb{Z}/n\mathbb{Z}$.

3 Secret sharing scheme for an arbitrary access structure

In the class we discussed a general definition of *secret sharing* for an arbitrary *access structure*. We showed that for any access structure there exists a valid secret sharing scheme (though the general construction suggests to the participants private keys that are much greater than the size of the secret key).

[1] In the construction that we discuss below, the joint distributions $(S_{i_1}, \ldots, S_{i_\ell})$ for non authorized groups are always the uniform distributions of ℓ independent random variables, though the general definition admits more complicated constructions.

4 Attack with a pair of predefined clear messages

Let $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively. Let us consider the following game between an adversary and Alice.

- Adversary uses an algorithm $Adv_1()$ that chooses two clear messages $m_a, m_b \in \mathcal{M}$;
- Alice chooses at random $i \in \{a, b\}$ (with equal probabilities), samples a secret key $k \leftarrow \mathrm{Gen}()$, and computes the encrypted message $e = \mathrm{Enc}(m_i, k)$;
- Adversary computes $j \in \{a, b\}$ using another algorithm $j \leftarrow Adv_2(m_a, m_b, e)$.

The success of the adversary is defined as follows:

$$\textbf{success} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

In words: the adversary prepared a pair of messages m_a, m_b; Alice decides which message to encrypt; then the adversary tries to understand which of the messages was encrypted.

Theorem 3. *If* $\Pi = \langle \mathrm{Gen}(), \mathrm{Enc}(), \mathrm{Dec}() \rangle$ *is a secure encryption scheme, then*

$$\mathrm{Prob}[\textbf{success} = 1] = 1/2.$$

In words: the adversary has no better strategy than simply toss a coin and suggest an answer at random

(We proved this theorem in the class.) In the next lecture we will adapt this attack to define security against an adversary with bounded computational resources.

5 Practical algorithms and admissibly small errors

We say that an algorithm is computationally efficient (feasible) if it stops in time at most $\mathrm{poly}(n)$ for all inputs of size n (for some polynomial $\mathrm{poly}(n)$). This definition applies to deterministic and to randomized algorithms. This definition defines the same class of algorithm for many popular models of computation, such as Turing machines with one or many tapes, random-access machine, and many other models.

We say that a function $f : \mathbb{N} \to \mathbb{R}_+$ is *negligible*, if for any polynomial $\mathrm{poly}(n)$ (that is not identically equal to zero) there is a natural number n_0 such that for all $n > n_0$ we have $|f(n)| < 1/|\mathrm{poly}(n)|$. In words: a negligible function goes to 0 faster than any inverted polynomial.

Exercise 1 (see also DM attached to this lecture)**.**
(a) If $f(n)$ and $g(n)$ are negligible functions, then $f(n) + g(n)$ and $f(n) \cdot g(n)$ are also negligible.
(b) If $f(n)$ is negligible function and C is a real number, then $C \cdot f(n)$ is also a negligible number.
(c) The functions $1/\log n$, $1/\sqrt{n}$, $1/(n+5)^2$, $1/n^{10}$ are not negligible.
(d) The functions e^{-n}, $e^{-n/10}$, $e^{-\sqrt{\log n}}$, $n^{-\log\log n}$ are negligible.

In what follows we will discuss encryption scheme and protocols that are sucre against "realistic" adversary. We usually believe that a scheme is practically secure, if attacks of any adversary which uses an algorithm *computable in polynomial time* succeeds with a *negligible* probability. In the next lectures we will see more precise forms of this definition.

References

[1] H. Tyagi and S. Watanabe. Information-Theoretic Cryptography. Cambridge Univ. Press 2023.

[2] J. Katz, Y. Lindell. Introduction to modern cryptography, CRC Press, 2021

[3] C. Walter. Arithmétique. Univ. de Nice, 2011. Chapitre 3.
https://math.unice.fr/~walter/L1_Arith/

[4] Louis Nebout. Propriétés de $\mathbb{Z}/n\mathbb{Z}$.
https://maths-olympiques.fr/wp-content/uploads/2017/09/arith_zn.pdf