Introduction to cryptology
TD#1

2018-W04

Exercise 1: Finite fields, $\mathbf{F}_2$, $\mathbf{F}_2^n$

A *commutative group* $(G, \star)$ is defined by a non-empty set of elements G and an operation $\star$ that satisfies the following properties:

- G is closed by $\star$: $\forall a, b \in G,\ a \star b \in G$
- There is a neutral (or identity) element: $\exists e \in G$ s.t. $\forall a \in G,\ a \star e = a$
- The law is associative: $\forall a, b, c \in G,\ a \star (b \star c) = (a \star b) \star c$
- Every element of G is invertible: $\forall a \in G, \exists b$ s.t. $a \star b = e$
- The law is commutative: $\forall a, b \in G,\ a \star b = b \star a$

A group is usually noted either additively ($\star$ is denoted $+$; the inverse of an element a is denoted $-a$; the neutral element is noted 0) or multiplicatively ($\star$ is denoted $\cdot$, $\times$ or $*$; the inverse of an element a is denoted $1/a$ or a^{-1}; the neutral element is noted 1). Common examples of groups are $(\mathbf{Z}, +)$ (the relative integers with addition), $(\mathbf{R}, +)$ (the real numbers with addition) or $(\mathbf{R}\backslash\{0\}, \times)$ (the real numbers except zero with multiplication). Note that $(\mathbf{N}, +)$ is not a group (only 0 has an inverse).

A *field* $(k, +, \times)$ is defined by a set of elements k and two operations (here $+$ and $\times$) such that:

- $(k, +)$ is a commutative group.
- $(k\backslash\{0\}, \times)$ (also written $k^\times$ for short) is a commutative group, where 0 is the neutral element of $(k, +)$.
- The element 0 is absorbing for $\times$, i.e. $\forall a \in k,\ a \times 0 = 0$.
- The law $\times$ is distributive over $+$: $\forall a, b, c \in k,\ a \times (b + c) = a \times b + a \times c$.

Common examples of fields are $(\mathbf{R}, +, \times)$ (the real numbers with addition and multiplication) and $(\mathbf{Q}, +, \times)$ (the rational integers with addition and multiplication). A counter-example is $(\mathbf{Z}, +, \times)$, as only 1 has a multiplicative inverse.

It is not necessary for a field to have an infinite number of elements. Examples of *finite fields* are the integers modulo a prime number p, which have p elements. Such fields are usually noted $\mathbf{Z}/p\mathbf{Z}$ or $\mathbf{F}_p$.

Q. 1: Which of the followings are fields: $(\mathbf{N}, +, \times)$, $(\mathbf{C}, +, \times)$, $(\mathbf{Z}/7\mathbf{Z}, +, \times)$, $(\mathbf{Z}/33\mathbf{Z}, +, \times)$?

Q. 2: Give the addition and multiplication tables of the field $\mathbf{F}_2$ (we will denote its only two elements by 0 and 1). What happens when an element of the field is added to itself an even number of times? An odd number?

*

Let k be a field; a k-vector-space E is a set of vectors endowed with an *inner addition law* ($+$, used to add vectors together) and an *outer multiplication law* ($\times$, $\cdot$, ..., used to multiply a vector by a scalar from k). A vector-space must satisfy the following:

- $(E,+)$ is a commutative group.
- The multiplication law must be such that $\forall \vec{u}, \vec{v} \in E$, $\forall \lambda, \mu \in k$:
  - $\lambda(\vec{u}+\vec{v}) = \lambda\vec{u} + \lambda\vec{v}$.
  - $(\lambda\mu)\vec{u} = \lambda(\mu\vec{u})$.
  - $(\lambda+\mu)\vec{u} = \lambda\vec{u} + \mu\vec{u}$.
  - $1\vec{u} = \vec{u}$.

Two vectors $\vec{u}$, $\vec{v}$ are *linearly independent* if $\nexists \lambda \in k$ s.t. $\lambda\vec{u} + \vec{v} = \vec{0}$, where $\vec{0}$ is the neutral element for $(E,+)$. More generally, n vectors are linearly independent if there is no linear combination of any of them that sums to the zero vector. The *dimension* $\dim(E)$ of a vector-space E is equal to the maximum size of a family of linearly independent vectors of E. Note that this dimension is not necessarily finite.

A *basis* of a finite-dimensional vector-space is a family of $n = \dim(E)$ linearly-independent vectors $\vec{b_0}, \ldots, \vec{b_{n-1}}$ such that all vectors of E can be written as a linear combinations of elements of the basis: $\forall \vec{u} \in E, \exists \lambda_0 \ldots \lambda_{n-1} \in k$ s.t. $\vec{u} = \lambda_0\vec{b_0} + \ldots + \lambda_{n-1}\vec{b_{n-1}}$.

An example of vector-space of dimension 2 is $\mathbf{R}^2$. Two possible bases for this vector-space are $\begin{pmatrix}1 & 0\end{pmatrix}^t$, $\begin{pmatrix}0 & 1\end{pmatrix}^t$ and $\begin{pmatrix}1 & 2\end{pmatrix}^t$, $\begin{pmatrix}1 & 1\end{pmatrix}^t$. An example of vector-space of infinite dimension is $\mathbf{R}[X]$, the polynomials with coefficients in $\mathbf{R}$. A possible (infinite) basis for this vector-space is $1, X, X^2, X^3, \ldots$.

Q.3: Give a basis for the four-dimensional vector-space $\mathbf{F}_2^4$. What happens when an element of the vector-space is added to itself an even number of times? An odd number?

Q.4: Write the identity matrix of $\mathcal{M}_4(\mathbf{F}_2)$, the vector-space of matrices of dimension 4 over $\mathbf{F}_2$.

Q.5: In the following, we let $M = \begin{pmatrix}0 & 1 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 1\\ 0 & 0 & 0 & 1\end{pmatrix}$, $\vec{u} = \begin{pmatrix}1 & 1 & 0 & 1\end{pmatrix}$, $\vec{v} = \begin{pmatrix}1\\1\\1\\0\end{pmatrix}$ be respectively a matrix, a row vector, and a column vector of dimension four over $\mathbf{F}_2$.

Compute $\vec{u}M$ and $M\vec{v}$. Is M an invertible matrix?

Exercise 2: The smartwatch fall problem

There are many possible formulations for this classical exercise (some of them more gory than others). We will adopt the following.

An alpinist is testing a new smartwatch, and wants to determine up to what height (in metres) it can fall without breaking (we assume that an upper-bound n (e.g. 100m) is known). Our alpinist does not want to break too many watches to find out the answer, but also wishes to avoid making too many trials (as each of them requires to climb up to a certain height, drop the watch, and climb or rappel down to the ground to find out if it is broken). You need to help him/her in determining the following tradeoffs.

Q.1: Describe an algorithm that breaks one watch and needs n drops in the worst case.

Q.2: Describe an algorithm that breaks up to $\lceil \log_2(n) \rceil$ watches and needs $\mathcal{O}(\log_2(n))$ drops in the worst case.

Q.3: Describe an algorithm that breaks two watches and needs $\mathcal{O}(\sqrt{n})$ drops in the worst case.

Note. The last algorithm is an instance of a "baby-step giant-step" method, that is also useful to compute discrete logarithms (i.e. logarithms in a finite group).

Exercise 3: One-time pad

Q.1: Let $x \in \mathbf{F}_2$ and r be an element drawn uniformly at random from $\mathbf{F}_2$ (i.e. $\Pr[r = 0] = \Pr[r = 1] = 0.5$. What is:

- $\Pr[x + r = 0]$?
- $\Pr[x + r = 1]$?
- $\Pr[x = 0 | x + r = 0]$?
- $\Pr[x = 0 | x + r = 1]$?
- $\Pr[x = 1 | x + r = 0]$?
- $\Pr[x = 1 | x + r = 1]$?

Hint. You may use Bayes' formula: $\Pr[A|B] = \Pr[B|A] \Pr[A] / \Pr[B]$.

Q.2: Let $\vec{x} \in \mathbf{F}_2^n$ and r be an element drawn uniformly at random from $\mathbf{F}_2^n$. For each element $\vec{y}$, $\vec{z}$ of $\mathbf{F}_2^n$, what is $\Pr[\vec{x} + \vec{r} = \vec{y}]$? $\Pr[\vec{x} = \vec{z} | \vec{x} + \vec{r} = \vec{y}]$?

Q.3: Assume that $x \in \{0,1\}^n$ is a message written as binary data. Assume that $r \in \{0,1\}^n$ is drawn uniformly at random among all binary strings of length n. Explain why observing $x \oplus r$ (the bitwise XOR of x and r) does not reveal any information about x.

Q.4: We will informally say that a cipher $\mathscr{C}$ is *perfectly secure* if observing the *ciphertext* $c = \mathscr{C}(p)$ does not reveal any new information about the *plaintext* p.

Let p and k be n-bit strings. Under what condition on k is the cipher $\mathscr{C} : p \mapsto p \oplus k$ perfectly secure?

Q.5: Let $\mathscr{C}$ be a perfectly secure cipher as above. Is its concatenation $\mathscr{C}^2 : p||p' \mapsto p \oplus k||p' \oplus k$ perfectly secure?

Exercise 4: Operand size blow-ups

Q.1: Let x and y be two n-bit natural integers. What is the maximum number of bits necessary to represent their sum?

Q.2: Let x and y be two n-bit natural integers. What is the maximum number of bits necessary to represent their product?

Q.3: On a 64-bit processor with a 64-to-64-bit unsigned integer adder and a 64-to-64-bit unsigned integer multiplier, what is the maximum operand size that guarantees that the result of any addition or multiplication can be represented on the CPU? Do you think that this is large enough to implement public-key cryptography? How would you implement addition and multiplication for larger number?

Exercise 5: Key size of an ideal block cipher

A binary block cipher with κ-bit keys and n-bit blocks is a family of permutations $\mathscr{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. That is, for every $k \in \{0,1\}^\kappa$, $\mathscr{E}(k,\cdot)$ is an invertible mapping over binary strings of length n. An *ideal* block cipher is a cipher such that for every key k, $\mathscr{E}(k,\cdot)$ is drawn uniformly at random among all possible such invertible mappings.

Q.1: What is (in function of n) the minimum key size (in bits) of an ideal block cipher?

Q.2: Give a lower and upper-bound numerical estimates for $n = 128$. Compare with an existing block cipher (if you know any).

Hint. You may use the inequalities $n! > (n/3)^n$, $n! < (n/2)^n$ (valid for large n). Recall also that $\log(ab) = \log(a) + \log(b)$; $\log(a^b) = b\log(a)$.