Elements of Cryptography

Xavier Servot *

March 2019

1 Motivations

Alice and Bob want to communicate on a public channel, i.e. a channel where everyone could listen to what information they send each other. We want to understand how they still can communicate secretly in that configuration.

The concept behind it is called a one-way trapdoor function, which combined with randomly generated secret keys from both Alice and Bob, can provide a framework for effective and secure communications.

The RSA algorithm implements a one-way trapdoor function that makes use of arithmetic to ensure secrecy. To learn about how it functions, we first dive into group theory and arithmetic.

2 Group theory

2.1 First definitions

Definition 2.1 (Commutative groups)**.** Let G be a set and let

$$*: \begin{array}{c} G \times G \to G \\ (a, b) \mapsto a * b \end{array}$$

be an operator on G. $(G, *)$ is a *commutative group* if

- $\exists e \in G$ such that $\forall g \in G, g * e = e * g = g$ (Existence of the neutral element)

*EPFL

- $\forall g \in G, \exists g^{-1} \in G$ such that $g*g^{-1} = g^{-1}*g = e$ (Existence of inverses)
- $\forall g, g' \in G, g * g' = g' * g$ (Commutativity)
- $\forall g_1, g_2, g_3 \in G, g_1 * (g_2 * g_3) = (g_1 * g_2) * g_3$ (Associativity)

Definition 2.2 (Group homomorphism)**.** Let $(G, *), (H, .)$ be groups. Then $\phi\colon G \to H$ is a *group homomorphism* if

- $\forall g, g' \in G, \phi(g * g') = \phi(g).\phi(g')$
- $\phi(e_G) = e_H$
- $\forall g \in G, \phi(g^{-1}) = \phi(g)^{-1}$

Definition 2.3 (Image)**.** Let $(G, *), (H, .)$ be groups, and $\phi\colon G \to H$ an homomorphism. The *image* of ϕ is

$$\operatorname{Im}\phi = \{\phi(g)\colon g \in G\}$$

The image is a subgroup of H.

Definition 2.4 (Kernel)**.** Let $(G, *), (H, .)$ be groups, and $\phi\colon G \to H$ an homomorphism. The *kernel* of ϕ is

$$\ker\phi = \{g \in G\colon \phi(g) = e_H\}$$

The kernel is a subgroup of G.

To get the hang of these notions, we have the following theorem

Theorem 2.1. Let $(G, *), (H, .)$ be groups, and $\phi\colon G \to H$ an homomorphism. Let $h \in H$. We consider the set

$$\phi^{-1}(\{h\}) = \{g \in G\colon \phi(g) = h\}$$

Let g_0 be an elements of this set. Then the set is either empty or:

$$\phi^{-1}(\{h\}) = g_0 * \ker\phi$$

Proof. Let g_0 be an element of the set. Then, if g_1 is another element of the set, we have:

$$\phi(g_0) = \phi(g_1) = h \Rightarrow \phi(g_1 * g_0^{-1}) = e_H$$

Thus $g_1 * g_0^{-1} \in \ker \phi \Rightarrow g_1 \in g_0 \ker \phi$.

Conversely, let g_0 be an element of the set, and $g_1 \in \ker \phi$, then

$$\phi(g_0 * g_1) = \phi(g_0).\phi(g_1) = h.e_H = h$$

Thus the sets $g_0 * \ker \phi$ and $\phi^{-1}(\{h\})$ are included in each other and are thus equal. □

Theorem 2.2 (Kernel-Image theorem for groups)**.** Let $(G, *), (H, .)$ be groups, and $\phi \colon G \to H$ an homomorphism. Then

$$|G| = |\operatorname{Im} \phi| \cdot |\ker \phi|$$

Proof. We consider $\operatorname{Im} \phi$: $\forall h \in \phi(G), \exists g_0 \in G$ such that $\phi(g_0) = h$. Also,

$$\phi^{-1}(\{h\}) = g_0 * \ker \phi$$

And thus we know that for each image of ϕ, there are $|\ker \phi|$ elements of G which map to that element via the application ϕ. Thus, as each element of G maps to only one element of $\operatorname{Im} \phi$,

$$|G| = |\operatorname{Im} \phi| \cdot |\ker \phi|$$

Another way to say that is that the sets of pre-images of two different images are separate thus:

$$G = \bigsqcup_{h \in \operatorname{Im} \phi} \phi^{-1}(\{h\})$$

and the application $g \mapsto g_0 * g$ is injective thus we have the equality. □

Definition 2.5 (Normal subgroup)**.** $K \subset G$ is *normal* if $\forall g \in G$, considering the application

$$Ad_g \colon h \in H \mapsto g.h.g^{-1}$$

we have $g.K.g^{-1} = K$ i.e. the group is invariant by the transformation Ad_g.

The notion of normal subgroup is important as it is the basis for our notation for the group $\mathbb{Z}/n\mathbb{Z}$. We are going to see this later.

2.2 Lagrange Theorem

Theorem 2.3. Let $(G, .)$ be a finite group and $H \subset G$ a subgroup. Then $|H|$ divides $|G|$.

Proof. Let's consider the sets of form $g.H$. As $e_H \in H$, we have

$$G = \bigcup_{g \in G} g.H$$

Let's consider the intersection of such sets: if $g.H \cap g'.H \neq \varnothing$ then $\exists h, h' \in H$ such that

$$g.h = g'.h' \Rightarrow g'.h'.h^{-1} = g \Rightarrow g \in g'.H$$

thus $g'.H \subset g.H.H = g.H$ and conversely, thus if two of these sets do intersect, they are equal. We can thus index $\cup_g g.H$ by the set of elements of G which give different sets $g.H$, without changing that the reunion amounts to G.

In other words, $|\{g.H : g \in G\}| \leq |G|$, and if $\{g.H : g \in G\} = \{H_1, \ldots, H_k\}$, we can say that

$$G = \bigsqcup_{i \in \{1,\ldots,k\}} H_i$$

and most importantly $\forall i \in \{1, \ldots, k\}, \exists g_i$ such that $H_i = g_i.H$ and thus

$$G = \bigsqcup_{i \in \{1,\ldots,n\}} g_i.H$$

we thus thus have

$$|G| = \sum_{i=1}^{k} |g_i.H|$$

Now, the application $h \mapsto g_i.h$ is bijective (because it has an inverse $h \mapsto g_i^{-1}.h$) thus $|g_i.H| = |H|$ and

$$|G| = k|H|$$

□

2.3 Generating set of a group

Proposition 2.1. Let $(G, .)$ a group and $A \subset G$. There exists a subgroup $H \subset G$ which is minimal for the property $A \subset H$, in the sense that every subgroup that contains A contains H. We note this groupe $\langle A \rangle$. We say that A generates H.

Proof. Let $G_* = \{H \text{ subgroup of } G \colon A \subset H\}$. We have $G \subset G_*$. Let

$$\langle A \rangle = \bigcap_{H \in G_*} H$$

$A \subset \langle A \rangle$, we now need to verify that it is a group. Let $g, g' \in \langle A \rangle$. In particuliar, $\forall H \in G_*, g, g' \in H$. As H is a subgroup, we also have $\forall H \in G_*, g.g'^{-1} \in H \Rightarrow g.g'^{-1} \in \langle A \rangle$. □

This notion is very interesting because it has a relation with exponentiation.

Definition 2.6. Let $(G, .)$ a group and $g \in G$. We will define a function $\exp_g \colon \mathbb{Z} \to G$ by induction. Let $\exp_g(0) = e_G$. Then $\exp_g(n+1) = g.\exp_g(n)$. Also, let $\exp_g(-n-1) = g^{-1}.\exp(-n)$. We can deduct from this definition the following properties, which I am sure you are familiar with:

$$\exp_g(-n) = \exp_g(n)^{-1} \quad \exp_g(n+m) = \exp_g(m).\exp_g(n)$$

These properties imply that $\exp_g$ is group homomorphism from $(\mathbb{N}, +)$ to $(G, .)$. We will also use the notation

$$g^n := \exp_g(n)$$

And

$$g^{\mathbb{Z}} := \{g^k \colon k \in \mathbb{Z}\} \subset G$$

Proposition 2.2.

$$\langle \{g\} \rangle = g^{\mathbb{Z}}$$

Proof. As $\langle \{g\} \rangle$ is a group that contains g, it also contains g^{-1} and $\forall n \in \mathbb{Z}, g^n \in \langle \{g\} \rangle$. Thus $g^{\mathbb{Z}} \subset \langle \{g\} \rangle$. Also, $g^{\mathbb{Z}}$ verifies every property of a group, thus if $\langle \{g\} \rangle \setminus g^{\mathbb{Z}} \neq \varnothing$, we have that $\langle \{g\} \rangle$ is not the smallest group that contains $\{g\}$ anymore, which is absurd. □

This proof gives an intuition for an alternative definition of $\langle A \rangle$.

Theorem 2.4. Let $(G, .)$ a group and $A \subset G$.

$$\langle A \rangle = \{a_1^{n_1}.\cdots.a_k^{n_k} : \forall i \in \{1, \ldots, k\}, a_i \in A, n_i \in \mathbb{Z}\}$$

Proof. Let $\exp_A(\mathbb{Z}) = \{a_1^{n_1}.\cdots.a_k^{n_k} : \forall i \in \{1, \ldots, k\}, a_i \in A, n_i \in \mathbb{Z}\}$. Proving that this set is a group is easy. When we multiply two elements of this group together, it gives another element of the required form. Also, we have an inverse by multiplying by -1 every exponent in the expression. The existence of the identity is proven by $a^0 = e_G \subset \exp_A(\mathbb{Z})$.

Now, $\langle A \rangle$ contains $\exp_A(\mathbb{Z})$ by construction, thus by the same argument as the previous proposition, $\langle A \rangle = \exp_A(\mathbb{Z})$. □

2.4 Order of an element

From here, we will note $\langle g \rangle$ for $\langle \{g\} \rangle$.

Definition 2.7. Let $(G, .)$ a group and $g \in G$. If $\langle g \rangle$ is finite, we define the order of g ($\operatorname{ord} g$) to be $|\langle g \rangle|$. Otherwise, we define the order to be ∞.

Proposition 2.3. If G is a finite group, $\operatorname{ord} g$ divides $|G|$.

Proof. Direct implication of Lagrange theorem. □

Proposition 2.4. Let $(G, .)$ a finite group. Let $g \in G$. If $\operatorname{ord} g \in \mathbb{N}$, then $\operatorname{ord} g$ is the smallest non-trivial integer solution of

$$g^n = e_G$$

Proof. We are interested in $\ker \exp_g$. In particular, we are interested in the smallest non trivial integer in this set, i.e. not 0. Now, $\exp_g : \mathbb{Z} \to G$, thus the application is not injective, as $\mathbb{Z}$ is infinite and G is finite. Thus $\ker \exp_g \neq \{0\}$. Then let $N \in \ker \exp_g$ the smallest integer in the kernel that is not 0.

Let $n \in \ker \exp_g$. We perform the euclidean division of n by N: $n = Nk + r$, $r \in \{0, \ldots, N-1\}$. And

$$g^r = g^{n-Nk} = g^n.(g^N)^{-k} = e_G$$

thus if an integer is in the kernel, it is a multiple of N.

Now, to show that $\operatorname{ord} g \leq N$, we have to show that every one of the following elements is distinct:

$$g_0 = e_G, \ldots, g_{N-1}$$

If two of those elements were not distinct, we would have

$$g_k = g_l \Leftrightarrow g_{k-l} = e_G$$

Thus $k-l = Nq$. Now $k, l \in \{0, \ldots, N-1\}$ thus $k-l \in \{-N+1, \ldots, N-1\}$ and $q = 0$. Which implies that $k = l$.

Now let's show that $g_{\mathbb{Z}} = \{g_0, \ldots, g_{N-1}\}$. Let $g^n \in g^{\mathbb{Z}}$. Then, we perform the euclidean division of n by N: $n = Nq + r$. And in particular we have

$$g^n = (g^N)^q . g^r = g^r$$

Thus $\forall n \in Z$, $g^n \in \{g_0, \ldots, g_{N-1}\} \Rightarrow g^{\mathbb{Z}} = \{g_0, \ldots, g_{N-1}\}$, as every element is distinct. □

2.5 Cyclic groups

Definition 2.8. A group G generated by a unique element $g \in G$ is cyclic i.e.

$$G = \langle g \rangle$$

Proposition 2.5. Let $(G, .)$ cyclic. If it has an infinite cardinal, it is isomorphic to $\mathbb{Z}$. Otherwise, let n be its cardinal, it is isomorphic to $\mathbb{Z}/n\mathbb{Z}$.

Proof. If G is infinite, we consider the morphism

$$\exp_g : \begin{array}{c} \mathbb{Z} \to G \\ n \mapsto g^n \end{array}$$

which has a kernel $\{0\}$ and is thus sujective and injective i.e. an isomorphism

Otherwise, let $\operatorname{ord} g = n$. We consider the morphism

$$\exp_g : \begin{array}{c} \mathbb{Z}/n\mathbb{Z} \to G \\ n \mapsto g^n \end{array}$$

which has a kernel of the form $k.\{n\}$, and as $n = 0$ in $\mathbb{Z}/n\mathbb{Z}$, its kernel is $\{0\}$ and it is thus a bijection. □

We now have a sense of how groups, in particular cyclic ones, are connected to arithmetic.

3 Arithmetic

We are not going to go over basic definitions and properties of arithmetic. Let's just say that in our notations, $[a]_n$ is the equivalence class of a in $(\mathbb{Z}/n\mathbb{Z}, +/.)$, or whatever group we are working with in that paragraph.

3.1 Euclide's algorithm

We are going to consider the *greatest common divisor* of two numbers a and b, noted $\gcd(a, b)$ or $(a; b)$ or $a \wedge b$. These properties will be useful in this section:

- Let $d := \gcd(a, b)$. Then $\exists k_a, k_b \in \mathbb{N}$ such that $a = dk_a, b = dk_b$. Thus

  $$a + b = d(k_a, k_b) \Rightarrow \gcd(a, b) | \gcd(a, a + b)$$

  Conversely, let $d' := \gcd(a, a+b)$. Then $\exists k_a, k_{a+b}$ such that $a = d'k_a, a+b = d'k_{a+b}$. Thus

  $$b = d'(k_{a+b} - k_a) \Rightarrow \gcd(a, a + b) | \gcd(a, b)$$

  Thus $\gcd(a, b) = \gcd(a, a + b)$

- More generally, by the same argument,

  $$\gcd(a, b) | \gcd(ka + lb, k'a + l'b)$$

  and if $d = \gcd(ka + lb, k'a + l'b)$, we have in particular that

  $d|l'(ka + lb) - l(k'a + l'b) = (l'k - lk')a \quad d|k'(ka + lb) - k(k'a + l'b) = (k'l - kl')b$

  Thus

  $$d | \gcd((l'k - lk')a, (-l'k + lk')b) = |l'k - lk'| \gcd(a, b)$$

  And in general,

  $$|l'k - lk'| = 1 \Rightarrow \gcd(a, b) = \gcd(ka + lb, k'a + l'b)$$

We can now start constructing the algorithm, which uses the fact that if the euclidean division of a by b gives $a = bk + r$, we have by induction:

$$\gcd(a, b) = \gcd(a - b, b) = \ldots = \gcd(a - bk, b) = \gcd(r, b)$$

By convention, in $\mathbb{Z}$, 0 is divisible by every integer, and thus

$$\gcd(a, 0) = a$$

So now intuitively, starting from $\gcd(a, b) = \gcd(b, r) = \gcd(rk' + r', r) = \gcd(r, r') = \ldots$, if we keep doing this till we get a null rest, we get that the rest just before the null one is the gcd, as stated by the property above. Thus for example, if $r' = 0$ we'd have $\gcd(a, b) = \gcd(r, r') = \gcd(r, 0) = r$.

Now we are also sure that we are eventually going to get a null rest: for $a >= b \in \mathbb{N}$, $a = bk + r$ with $r < b$ and $\gcd(a, b) = \gcd(b, r)$, and this argument can be applied at each step thus we obtain a strictly decreasing series of numbers ($r < b$, and $a > b$ otherwise the gcd is trivial).

Now, knowing that, we can go further in this algorithm with the following theorem

Theorem 3.1 (Bézout)**.** Let $a, b \in \mathbb{N}$, and $d := \gcd(a, b)$. $\exists u, v \in \mathbb{Z}$ such that

$$au + bv = d$$

Proof. Intuitively, at each step we have an euclidean division of the form

$$r_{i-2} = r_{i-1}q_i + r_i$$

where we have $r_{-2} = a$ and $r_{-1} = b$ if r_0 designates the rest of the euclidean division of a by b. Thus right before the last step of the algorithm, let's stay step k, we have

$$r_{k-2} = r_{k-1}q_k + d \Leftrightarrow d = r_{k-2} - r_{k-1}q_k$$

And at the step $k - 1$ we have

$$r_{k-3} = r_{k-2}q_{k-1} + r_{k-1} \Leftrightarrow r_{k-1} = r_{k-3} - r_{k-2}q_{k-1}$$

Thus we can inject the equation of the step $k - 1$ into the equation of the step k to get an expression of d that only depends on the coefficients r_{k-2} and r_{k-3}.

Now it is easy to see that going all the way up to r_{-2} and r_{-1} we'd have an expression of d depending only on those two rests, in the sense that we'd have an equality of the form $au + bv = d$. □

Corollary 3.1.1. if a and b are coprime, i.e. $a \wedge b = 1$, iff $\exists u, v \in \mathbb{Z}$

$$au + bv = 1$$

3.2 Results from group theory

Recall that $(\mathbb{Z}/n\mathbb{Z}_*, .)$ is not always a group, as it needs for every element to have an inverse. Let's note $\mathbb{Z}/n\mathbb{Z}_\times$ as the elements of $(\mathbb{Z}/n\mathbb{Z}_*, .)$ which have inverses, which is not empty as 1 is always in it. This set is a group. Also, let's note $\mathbb{Z}/n\mathbb{Z}_+ := (\mathbb{Z}/n\mathbb{Z}, +)$.

Lagrange theorem tells us that $\mathbb{Z}/n\mathbb{Z}_+$ has cardinality n, and that there are subgroups if n is not prime. Otherwise, subgroups only have cardinality 1 or n. This is not new to us. However, a result that might be trickier to prove involves Euler's totient function ϕ. $\phi(n)$ is the number of positive integers up to n which are coprime with n.

Why we care about this function is this case, is that $\mathbb{Z}/n\mathbb{Z}_\times$ is a very different group than $\mathbb{Z}/n\mathbb{Z}_+$, and the cardinality of the multiplicative group is actually $\phi(n)$. $\phi(n)$ is the number of numbers less than n which are coprime with n. Now, coprime means having an inverse in $(\mathbb{Z}/n\mathbb{Z}, .)$ (Bezout's theorem), and thus belonging to $\mathbb{Z}/n\mathbb{Z}_\times$.

A result in cyclic groups tells us that $\forall a \in \mathbb{Z}/n\mathbb{Z}_\times$,

$$a^{\phi(n)} \equiv 1[n]$$

and a corollary of this theorem is Fermat's little theorem: $\forall p$ prime, $\forall a \in \mathbb{Z}/p\mathbb{Z}_\times$,

$$a^{p-1} \equiv 1[p]$$

as every number less than a prime number p is coprime with p, thus $\phi(p) = p - 1$

3.3 Chinese reminder theorem

3.3.1 Proof

Theorem 3.2. Let $n_1, \ldots, n_k$ be coprime integers. Then the system of equations

$$x \equiv a_i[n_i], i \in \{1, \ldots, k\}$$

has a unique solution mod $N := \prod_{i=1}^{k} n_i$

Proof. Let $N_i := \frac{N}{n_i}$. Then, we verify that N_i and n_i are coprime: if it wasn't the case then n_i would have a common factor with $\prod_{j=1,j\neq i}^{k} n_j$ which is absurde. Thus $\exists s_i, t_i \in \mathbb{Z}$ such that

$$s_i N_i + t_i n_i = 1$$

Then let

$$x = \sum_{i=1}^{k} a_i s_i N_i$$

we have $\forall l \in \{1, \ldots, k\}$

$$\begin{aligned} x &\equiv \sum_{i=1}^{k} a_i s_i N_i \ [n_l] \\ &\equiv a_l s_l N_l + n_l \sum_{i=1}^{k} (a_i s_i \prod_{j=1, j\neq i,l}^{k} n_j) \ [n_l] \\ &\equiv a_l \ [n_l] \end{aligned}$$

Now, let x_* be another solution. Then

$$x_* \equiv x \ [n_i] \forall i \in \{1, \ldots, k\}$$

i.e. $\exists k_i$ such that $x_* - x = k_i n_i$. Thus

$$(x_* - x)^k \equiv 0 \ [N] \Leftrightarrow x_* \equiv x \ [N]$$

□

3.3.2 The isomorphism described by the Chinese reminder theorem

Now, let $n_1, \ldots, n_k$ pairwise coprimes. What is going to be interesting to us is that if we take k rings $(\mathbb{Z}/n_i\mathbb{Z}, +, .)$, then for any $(a_1, \ldots, a_k) \in \mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_k\mathbb{Z}$, there exists only one solution $a \in \mathbb{Z}/N\mathbb{Z}$ such that

$$a \equiv a_i \ [a_i] \forall i \in \{1, \ldots, k\}$$

Thus there exists a mapping from $\mathbb{Z}/N\mathbb{Z}$ to $\mathbb{Z}/n_1\mathbb{Z} \times \cdots \times \mathbb{Z}/n_k\mathbb{Z}$, that we can verify to be an ismorphism. First, it is bijective as the solution always exists and is unique. Then, it is an morphism (of groups, we are not going to prove all conditions for an isomorphism of ring) as

- $a = 1$ is a solution (and unique as proved by the thm.) when $a_i = 1 \forall i$.
- if a is solution of $a \equiv a_i \ [n_i]$ and b is solution of $b \equiv b_i \ [n_i]$, we can easily check that $ab \equiv a_i b_i \ [n_i]$ by multiplying.
- the system of equations associated with the solution a^{-1} is $x \equiv a_i^{-1} \ [n_i]$, given that for the system $x \equiv a_i \ [n_i]$ has solution a.

What the chinese reminder theorem says in this sense is important. Say you want to find a number congruent to $y \, [a.b]$, and a and b are coprime. Well, it is the same as finding a number congruent to $y \, [a]$ and $y \, [b]$. i.e. we can change the set we are working with for convenience.

Also as we saw, we have an efficient solution for mapping a system of equations to its solution. And mapping a solution to its system is even easier, by just taking the mod for the base of each line of the system.

3.3.3 Chinese reminder theorem and Fermat's little theorem

Let p, q be distinct primes. Let $k := r(p-1)(q-1)$, where $r \in \mathbb{N}$, i.e. k is a multiple of $p-1$ and $q-1$. Then we have

$$a^k \equiv 1 \, [p]$$
$$a^k \equiv 1 \, [q]$$

Now, if we take the expressions to any exponent $l \in \mathbb{N}$, it still is congruent to 1 to their respective mod. Thus

$$a^{kl+1} \equiv a \, [p]$$
$$a^{kl+1} \equiv a \, [q]$$

thus by the chinese reminder theorem,

$$a^{kl+1} \equiv a \, [pq]$$

We will see that this is important in the functionning of the RSA algorithm, which is the subject of our next section.

4 How RSA works

4.1 The intuition behind RSA

We are going to describe at a high level how RSA works. We still have our setup where Alice and Bob want to communicate information that no one should be able to decrypt on a public channel

$$\text{Alice} \begin{bmatrix} \text{plaintext } t \\ \text{private secret key } k_A \\ \text{encryption alg. } E_{k_A}(\cdot) \end{bmatrix} \xrightarrow[\text{ciphertext } c=E_{k_A}(t)]{\text{public channel}} \text{Bob} \begin{bmatrix} \text{ciphertext } c = E_{k_A}(t) \\ \text{private secret key } k_B \\ \text{decryption alg. } D_{k_B}(\cdot) \\ \text{decrypted } t = D_{k_B}(E_{k_A}(t)) \end{bmatrix}$$

What we are trying to achieve is find $m, e, d \in \mathbb{N}$ such that $\forall t \in \mathbb{Z}/m\mathbb{Z}$

$$[(t^e)^d]_m = [t]_m$$

m is called the modulus, e called the encoding exponent and d called the decoding exponent. Then Alice would just need to send t^e to Bob and Bob would just apply $\cdot^d$ to what he received.

Of course, this does not tell us how we find e, d, m, but at least we know that d should be kept private, otherwise everyone could decrypt the message. e and m would be public keys to encrypt plaintexts.

4.2 Key generation: Bob's end

(i) We begin by generating large primes p, q at random

(ii) We set $m = pq$, which is the modulus used to decode the information.

(iii) Let k be a multiple of p and q, which Bob should keep secret. There are a few concrete ways to set k: either by setting it to $\phi(pq)$ or to $lcm(p-1, q-1)$.

(iv) Produce the encoding exponent e such that $gcd(e, k) = 1$. There is no need for e to be distinct for each time Bob generates the keys. A common choice for e is $2^{16} + 1$, which is prime.

(v) Bob broadcasts the public key (e, m) on the public channel.

(vi) Bob produce the decoding exponent d such that $de + kl = 1$, and keeps (d, m) as the private key.

Now, when Alice wants to send a message to Bob, she asks Bob to generate these keys, and receives (e, m), with which she encodes the message t and sends it to Bob, which will have no problem decrypting it. You now know why the last bits about the Chinese Reminder theorem were so important. Without it, we wouldn't have been able to prove that

$$[(t^e)]_{pq}^d = [t]_{pq}^{ed} = [t]_{pq}^{1-kl} = [t]_{pq}$$

4.3 Digital signature

We want to find a way for Alice to send an information, such that we can verify only she could send such information, i.e. produce a *digital signature*: where only she can sign, and everyone can verify that its her signature.

4.3.1 Hash functions

A hash function is a surjective function from $\mathbb{N}$ to a stream of bits, of fixed length or variable length. It is such that two very close inputs produce very differents hashes. The intuition is that for finding a good candidate for the input of a given hash, the best way is to use brute force.

4.3.2 How it's done

Let's now admit that Alice has already generated her keys for communication, and has broadcasted her public key (e_A, m), such that for sending her a message someone would use the trapdoor one way function $f_A = t^{e_A}[m]$. Let t be the plaintext she wants to sign, and h a hash function, publicly available. Then she defines the digital signature as

$$s := f_A^{-1}(h(t))$$

Now when she sends a message t to *Bob*, she actually sends (t, s), and Bob can verify $f_A(s) = h(t)$. Bob knows that only Alice could produce the inverse of $f_A(s)$.

So now, we Alice and Bob can communicate in privacy and with authenticity. Privacy because Alice ca send $(f_B(t), s)$ and authenticity because Bob can first decrypt $f_B(t)$ using his trapdoor information, and then compare $f_A(s)$ to $h(t)$, to know that it's actually Alice that sent the message to Bob.

4.4 Possible attacks

Trudy wants to know every way she could decrypt the ciphertext, not knowing d.

4.4.1 Factor m into pq

As m is public, Trudy could think that factoring m into its to prime components $m = pq$ is not hard. This would lead us to know k as it is a multiple of pq. As we said, there are a few concrete ways of setting k: $\phi(pq)$ or $lcm(p-1, q-1)$.

Now, knowing that, Trudy could then find d such that $de + kl = 1$, the same way Bob did, and thus get the secret key.

However, the problem of factoring m into its prime components is an extremely hard one, as p and q are chosen extremely large, such that multiplying them is an easy computation but finding the inverse map is unfeasible.

4.4.2 Find t given t^e

Alice broadcasts on a public channel $c = t^e[m]$. Trudy could try to solve (for x): $c = x^e[m]$. Now, this problem is known as the discrete logarithm, and it takes $\sqrt{m}$ operations to compute it. For m large enough, this is an absurd amount of calculations to perform.

For reference, computing the discrete exponentiation $t \mapsto t^e$ is done is $\log_2(m)$ operations. For $m \approx 2^{200}$, we have $\log_2(m) \approx 40$ and $\sqrt{m} \approx 2^{100} \approx 10^{30}$.

The discrete exponentiation is called a one-way trapdoor function, because the computation is done easily in one way but is very hard in the other way (the other way being the discrete log). Yet, it is also easy to do find t given t^e and also (d, m). This is called the trapdoor information, because it makes the inverse computation easy (we only have to perform a discrete exponentiation knowing this information).

4.4.3 Tampering the directory of public keys

We Bob broadcast his public key, it rather means that he will store his public key in public access database. Now, how do we know that no one modified a public key stored in such database? For example, Trudy could replace Bob's public key with hers, and thus when Alice sends a message encrypted with

what she thinks is Bob's public key, she really is encrypting it using Trudy's public key. This means Trudy can decrypt the message Alice meant to send to Bob.

What we do to counter that is use trusted agencies, like Symantec. Symantec ditributes a public key into a hardware where it cannot be modified after, like hardcoded into the hardware.

Symantec digitally signs every entry of the database, using the technique mentioned above. The result is called a certificate. If the information in the database has been changed, we know that the signature won't match and the certificate will be invalid.

Thus, we know that public key in the database is legitimate and we can be confident in using it.