**Mathematics, KTH**
Bengt Ek
November 2015

Supplementary material for SF2736, Discrete mathematics:

# The RSA cryptosystem and primality tests

Secret codes (i.e. codes used to make messages unreadable to "outsiders") were formerly mainly something for diplomats, spies, and the military, but nowadays they have become very important in connection with computer communication.

By a **cryptosystem**

$$\begin{tikzcd}\mathcal{M} \arrow[r, shift left, "E"] & \mathcal{C} \arrow[l, shift left, "D"]\end{tikzcd}$$

we shall mean two finite sets $\mathcal{M}$ (the possible messages, plaintexts) and $\mathcal{C}$ (containing the corresponding possible ciphertexts) and two functions $E : \mathcal{M} \to \mathcal{C}$ for encryption and $D : \mathcal{C} \to \mathcal{M}$ for decryption, such that

$$D(E(m)) = m \text{ for all } m \in \mathcal{M}.$$

We shall assume that $\mathcal{M} = \mathcal{C} = \mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$ for some (large) $n$. If the messages we want to encrypt are in some other form, for instance ordinary text, they can be expressed as large integers using ASCII code for the symbols involved (and breaking long messages into smaller pieces to stay within $\mathbb{Z}_n$).

In a classical cryptosystem (such as replacement ciphers ("change every a into a k, every b into an s, and so on") and many other, more sophisticated, ones) knowing $E$ implied knowing $D$. They therefore both had to be kept secret. ("In principle" that is the case for any cryptosystem, since if you know $E$ you could try all $m \in \mathcal{M}$ until $E(m)$ is the ciphertext you want to read, but in practice that is not possible, because $\mathcal{M}$ is very large).

In 1976 another kind of cryptosystem, now known as **public-key cryptosystems** (Sw. **kryptosystem med offentlig nyckel**) was suggested by W. Diffie och M. Hellman. In such a system every user, $A$ say, would have his own encryption function $E_A$ and his own decryption function $D_A$. They are so "complicated" that even if you know $E_A$, it is very hard to find $D_A$. All the $E_A$ can be made public and every participant keeps his own $D_A$ secret. That means that anybody can produce a ciphertext to $A$, but only $A$ can read it.

One of the first, and probably the best known, such system, is the RSA (so named after the inventors, Ron Rivest, Adi Shamir, and Leonard Adleman) system. Since it is based on primes and their mathematics, it deserves a place in our course.

## The RSA system

Let $p$ and $q$ be different (big) primes and $n = p \cdot q$, $m = \phi(n)$ ($\phi$ is Euler's function). Since $\mathbb{Z}_n \approx \mathbb{Z}_p \times \mathbb{Z}_q$ (by the Chinese remainder theorem) and $(a, b) \in \mathbb{Z}_p \times \mathbb{Z}_q$ is invertible iff $a, b \neq 0$, we find the number of invertible elements in $\mathbb{Z}_n$ to be $m = \phi(n) = (p-1)(q-1)$.

Now let $x \in \mathbb{Z}_n$. By Euler's theorem $x^m = 1$ (in $\mathbb{Z}_n$) if(f) $\gcd(x, n) = 1$, i.e. iff $p, q \nmid x$. Suppose that $x \in \mathbb{Z}_n$ corresponds to $(a, b) \in \mathbb{Z}_p \times \mathbb{Z}_q$ (i.e. $f(x) = (a, b)$ with the notation in the material on the Chinese remainder theorem). Then for any $k \in \mathbb{N}$, $f(x^{km+1}) = (a^{km+1}, b^{km+1})$. If $a \neq 0$, $a^{km+1} = (a^{p-1})^{k(q-1)} \cdot a = 1^{k(q-1)} \cdot a = a$ (in $\mathbb{Z}_p$) and $0^{km+1} = 0$, so $a^{km+1} = a$ for all $a \in \mathbb{Z}_p$. In the same way $b^{km+1} = b$ (in $\mathbb{Z}_q$), so $f(x^{km+1}) = (a, b) = f(x)$ for alla $x \in \mathbb{Z}_n$, since $f$ is injective this gives the

**Theorem:**
Let $p$ and $q$ be distinct primes, $n = p \cdot q$, and $m = (p-1)(q-1)$.
If $x \in \mathbb{Z}_n$ and $s \in \mathbb{N}$ satisfies $s \equiv_m 1$, then

$$\boldsymbol{x^s = x} \text{ in } \boldsymbol{\mathbb{Z}_n,}$$
$$\text{i.e., for all } \boldsymbol{x \in \mathbb{Z},\quad x^s \equiv x \pmod{n}.}$$

That gives the following way **to construct an RSA system:**

1. Take two different primes $p$, $q$.
2. Let $\boldsymbol{n = p \cdot q}$ and $\boldsymbol{m = (p-1)(q-1)}$.
3. Find $e \in \mathbb{N}$ with $\gcd(e, m) = 1$, and $d \in \mathbb{N}$ with $\boldsymbol{e \cdot d \equiv 1 \pmod{m}}$.
4. Let $E, D : \mathbb{Z}_n \to \mathbb{Z}_n$ be given by $\boldsymbol{E(x) = x^e}$ and $\boldsymbol{D(x) = x^d}$.
   so $E(x) \equiv x^e \pmod{n}$, $D(x) \equiv x^d \pmod{n}$
5. Publish $n$ and $e$, keep $d$ secret (and throw away $m$).

Note that the modules are different. We calculate $E$, $D$ using the module $n$ and find $d$ with the module $m$.

To find $e$ in step 3, try different values and check the condition $\gcd(e, m) = 1$ with the Euclidean algorithm. That then also gives the corresponding $d$.

Since $D(E(x)) = (x^e)^d = x^{ed} = x^{de} = (x^d)^e = E(D(x)) = x$, by the theorem above, $D = E^{-1}$ in this system.

$D$ is hard to find (i.e., $d$ is hard to find knowing $n$ and $e$), because it is (we believe) hard to factor large integers (in a reasonable time). In practice, around 150 digits (in base 10) in $p$ and $q$ seems to be considered sufficient.

**Example.** Taking the (not-so-big) primes $p = 17$, $q = 23$, we find $n = 17 \cdot 23 = 391$ and $m = (17-1)(23-1) = 352$.
With (for instance) $e = 15$ ($\gcd(352, 15) = 1$, since $352 = 23 \cdot 15 + 7$, $15 = 2 \cdot 7 + 1$), we get $d = 47$ ($1 = 15 - 2 \cdot 7 = 15 - 2(352 - 23 \cdot 15) = -2 \cdot 352 + 47 \cdot 15$, so $15 \cdot 47 \equiv_{352} 1$).
To encrypt the very secret message $x = 367$, we compute $E(367) = 367^{15} = 114$ (in $\mathbb{Z}_{391}$). So the ciphertext is 114. Decrypting with $D$ we get $D(114) = 114^{47} = 367$ (also in $\mathbb{Z}_{391}$).

In the example the computations involved $367^{15}$, which has 39 digits in base 10. In a real system one would have x with hundreds of digits and also $e$, so $x^e$ would be far to big to compute even in a big computer.
Instead one can compute succesive squares, $x$, $x^2$, $x^4$, $x^8, \ldots$, in each step taking the result $\pmod{n}$ (i.e. computing in $\mathbb{Z}_n$), so one never has to handle numbers larger than $n^2$, which is feasible in a computer. Then the correct $x^i$ are multiplied (still $\pmod{n}$ in each step) to get the result.

**Example continued.** We could compute $D(114) = 114^{47}$ like this in $\mathbb{Z}_{391}$:
$x = 114$,
$x^2 = 12\,996 = 93$,
$x^4 = 93^2 = 8\,649 = 47$,
$x^8 = 47^2 = 2\,209 = 254$,
$x^{16} = 254^2 = 64\,516 = 1$, (That $x^{16} = 1$ is an "accident".)
$x^{32} = 1^2 = 1$.
Since $47 = (101111)_2$, we find $x^{47} = x^{32} \cdot x^8 \cdot x^4 \cdot x^2 \cdot x = 1 \cdot 254 \cdot 47 \cdot 93 \cdot 114 = = 11\,938 \cdot 93 \cdot 114 = 208 \cdot 93 \cdot 114 = 19\,344 \cdot 114 = 185 \cdot 114 = 21\,090 = 367$ (in $\mathbb{Z}_{391}$).

## Electronic signatures

The fact that "everybody" can encrypt messages for the user $A$, using the public key $E_A$, gives rise to a new problem. Namely, when $B$ sends a message to $A$, how can $A$ be sure that it was really $B$ who sent it and not an imposter?

That problem is solved by an **electronic signature**, using the fact that $D(E(x)) = E(D(x)) = x$.

Suppose $B$ wants to make sure that $A$ will be certain that it was $B$ who sent the message $x$. She can then, instead of $E_A(x)$ send $D_B(E_A(x))$ (or $E_A(D_B(x))$). Then $A$ can use the public $E_B$ to find $E_B(D_B(E_A(x))) = E_A(x)$ and using his own secret $D_A$ find $x$. Only $B$ could use $D_B$, so only $B$ could be the sender, and only $A$ could read the message, since $D_A$ was needed.

In a similar way, if $A$ wants to make $y$ publicly known, he can send $D_A(y)$ to everybody. Only somebody who knows $D_A$ could be the sender.

## Primality tests

To implement the RSA system, every user must have two very large primes of his own.

There are certainly enough primes. By the famous Prime number theorem, proved in 1896, the density of primes around $N$ is approximately $\frac{1}{\ln N}$ for large $N$, so near $N = 10^{150}$ about one number out of 350 is a prime. That means a computer can find primes rather easily by testing several candidates, provided it has a reasonably efficient way of testing if an integer with around 150 digits (in base 10) is a prime. That is, we need a good **primality test**.

How do we test if a small number, for instance 389, is a prime? The most direct method would be to search for factors, by checking all primes up to 19 (since the next prime, 23, has $23^2 > 389$) (the check does not have to be a full division, for instance $19 \mid 389 \Leftrightarrow 19 \mid (389 - 19) = 370 \Leftrightarrow 19 \mid 37$, (since $\gcd(19, 10) = 1$), so $19 \nmid 389$). 389 is a prime, but that method can not be used for really large numbers (even if we assume that we already know the primes up to $10^{75}$ and can check for divisibility in $10^{-12}$ seconds, the test of one number near $10^{150}$ could take more than $10^{53}$ years). Checking for all possible factors would almost amount to factorizing a large integer, and if we could do that, RSA wouldn't be secure anyway.

But we do know one way to check if $N$ is a prime, without factorizing:

**The Fermat test, base $b$, $1 < b < N$:**

$$\textbf{Is } b^{N-1} \equiv 1 \pmod{N}?$$

By Fermat's (little) theorem, the answer will for all $b$ be "yes" if $N$ is a prime, so if the answer is "no", we are certain that $N$ is not prime. But there are numbers, so-called (Fermat) **pseudoprimes** to base $b$, which pass the Fermat test to base $b$ without being prime.

**Example.** Since $2^{10} = 1024 = 3 \cdot 341 + 1$, $2^{340} \equiv_{341} 1^{34} = 1$, so 341 is a Fermat pseudoprime to base 2. But $3^{340} \equiv_{341} 56 \neq 1$, so 341 is not a Fermat pseudoprime to base 3 (and also not a prime).

As the example shows, the Fermat test would show that 341 is not prime, if we tried it with both base 2 and base 3. A **probabilistic** test for primality of $N$ could then be to take a number of bases $b_1, b_2, \ldots, b_k$ at random and perform the test for all of them. If $N$ is a prime, it would pass all the tests, and it should be improbable that a non-prime would (in the sense that only a very small fraction of all non-prime $N$ of similar size, say, would pass all of them). If we increase the number $k$ of bases, the probability of a non-prime passing the test should become "practically zero". Or should it?

In fact, there are non-primes $N$ which pass **all** Fermat tests with bases $b$ satisfying $\gcd(b, N) = 1$. Such numbers are called **Carmichael numbers** and they are exactly the square-free (i.e., for all primes $p$, $p^2 \nmid N$) composite integers such that if $p$ is a prime with $p \mid N$, then $(p - 1) \mid (N - 1)$. They all contain at least three primes and there are infinitely many of them.

The smallest Carmichael numbers are

$$561 = 3 \cdot 11 \cdot 17,\ 1105 = 5 \cdot 13 \cdot 17,\ 1729 = 7 \cdot 13 \cdot 19,\ 2465 = 5 \cdot 17 \cdot 29,$$

$$2821 = 7 \cdot 13 \cdot 31,\ 6601 = 7 \cdot 23 \cdot 41,\ 8911 = 7 \cdot 19 \cdot 67.$$

If the prime factors of a Carmichael number are very big, it is very improbable that we should detect that it is not prime with the probabilistic test described above.

The **Miller-Rabin test** with base $b$ (from 1980) is a modification of the Fermat test which, when performed with several bases, makes it highly improbable that a non-prime should be judged prime.

If $N$ is a prime and $x \in \mathbb{Z}_N$ satisfies $x^2 = 1$, then $x = \pm 1$ (if $N$ is prime, $N \mid (x^2-1) \Rightarrow N \mid (x-1)$ or $N \mid (x+1)$). That is the observation behind

**The Miller-Rabin test, base $b$, $1 < b < N$:**

$$\textbf{Let } N - 1 = u \cdot 2^r, \quad u \textbf{ odd}, \; r \geq 1 \text{ (if } N \text{ is odd).}$$

$$\textbf{Is } b^u \equiv_N 1 \textbf{ or } (b^u)^{2^i} \equiv_N -1 \textbf{ for some } i, \; 0 \leq i < r?$$

If $N$ is prime, the answer will be "yes" and one can show that if $N$ is composite, it will pass the Miller-Rabin test for at most $\frac{N}{4}$ of the bases $b$ with $1 < b < N$.

**Example.** For $N = 561$ and $b = 2$, we find $N - 1 = 560 = 35 \cdot 2^4$ and
$2^{35} \equiv_{561} 263$,
$2^{70} \equiv_{561} 263^2 = 69\,169 \equiv_{561} 166$,
$2^{140} \equiv_{561} 166^2 = 27\,556 \equiv_{561} 67$,
$2^{280} \equiv_{561} 67^2 = 4\,489 \equiv_{561} 1$.
Since $67 \not\equiv_{561} -1$, the Miller-Rabin test shows that 561 is not a prime.
Using the isomorphism (from the Chinese remainder theorem)

$$f : \mathbb{Z}_{561} \to \mathbb{Z}_3 \times \mathbb{Z}_{11} \times \mathbb{Z}_{17},$$

we can see what is happening here:
$f(2^{35}) = f(263) = (2, 10, 8) = (-1, -1, 8)$,
$f(2^{70}) = f(166) = (1, 1, 13)$,
$f(2^{140}) = f(67) = (1, 1, -1)$,
$f(2^{280}) = f(1) = (1, 1, 1)$,
$f(2^{560}) = f(1^2) = (1, 1, 1)$.
That $2^{560} \equiv_n 1$ for $n = 3$, 11, 17, means that 561 passes the Fermat test with base 2, but since $2^{35 \cdot 2^i} \equiv_n -1$ doesn't happen for the same $i$ for all of $n = 3$, 11, 13, $2^{35 \cdot 2^i} \not\equiv_{561} -1$ for all $i$, so 561 does not pass the Miller-Rabin test with base 2.

## Exercises

**1.** A user in an RSA system has the public parameters $(n, e) = (143, 17)$.
**a.** Encrypt the message $x = 71$.
**b.** Find the user's secret decryption parameter $d$.
**c.** Check your $d$ by using it to decrypt the result in a.

**2.** An RSA system uses $n = 1147 (= 31 \cdot 37)$ and $e \geq 332$.
What are the smallest possible value for $e$ and the corresponding $d$?

**3.** An RSA user has the parameter $n = p \cdot q = 57\,656\,617$.
What are the primes $p$ and $q$, if $m = (p-1)(q-1) = 57\,641\,220$?

**4.** Show that if $p$, $q$ are distinct primes, $p^{q-1} + q^{p-1} \equiv 1 \pmod{pq}$.

# Answers

**1a.** $E(x) = 80$, **b.** $d = 113$.

**2.** $e = 337,\ d = 673$.

**3.** $\{p, q\} = \{6427, 8971\}$.