Queen Mary
University of London

MAS 335 Cryptography

Notes 7: Public-key cryptography Spring 2008

In this section we will describe the RSA cryptosystem, the most popular public-key cryptosystem at present. We need some number-theoretic background first.

Number theory

The RSA enciphering function has the form $T_e : x \mapsto x^e \bmod n$ for some suitable n and e. In order to be able to decipher, we must be assured that this function is one-to-one. So we first discuss the number-theory required for this question.

Euler and Carmichael

Recall Euler's phi-function $\phi(n)$ whose value is the number of elements of $\mathbb{Z}/(n)$ (the integers mod n) which are coprime to n. We calculated this function back in Notes 2:

Theorem 19 *(a) If $n = p_1^{a_1} \cdots p_r^{a_r}$, where p_i are distinct primes and $a_i > 0$, then $\phi(n) = \phi(p_1^{a_1}) \cdots \phi(p_r^{a_r})$.*

(b) If p is prime and $a > 0$, then $\phi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1)$.

A well-known theorem of Fermat (often called *Fermat's Little Theorem*) asserts that, if p is prime, then $a^{p-1} \equiv 1 \pmod{p}$ for any number a not divisible by p. This theorem was generalised by Euler as follows:

Theorem 20 *If $\gcd(a,n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.*

Proof Let $x_1, x_2, \ldots, x_m$ be all the elements of $\mathbb{Z}/(n)$ which are coprime to n, where $m = \phi(n)$. Suppose that $\gcd(a,n) = 1$. Then a has an inverse b mod n, so that $ab \equiv 1 \pmod{n}$. Now let $y_i = ax_i \bmod n$, and consider $y_1, \ldots, y_m$. We have

- $\gcd(y_i, n) = 1$, since $\gcd(a,n) = \gcd(x_i,n) = 1$;

- $y_1, \ldots, y_m$ are all distinct: for if $y_i = y_j$, then $by_i = by_j$, that is, $bax_i \equiv bax_j \pmod n$, or $x_i \equiv x_j \pmod n$, so $x_i = x_j$.

Thus, the set $\{y_1, \ldots, y_m\}$ is the same as the set $\{x_1, \ldots, x_m\}$ (possibly in a different order), so their products are the same:

$$\prod x_i = \prod y_i \equiv \prod ax_i = a^m \prod x_i \pmod n.$$

Since the x_i are coprime to n, so is their product, and it has an inverse mod n. Multiplying the equation by this inverse we get $a^m \equiv 1 \pmod n$, as required.

One very important fact about Fermat's Little Theorem is that it cannot be improved:

Theorem 21 *Let p be prime. Then there exists a such that $a^{p-1} \equiv 1 \pmod p$ but no smaller power of a is congruent to 1 mod p.*

Such an element a is called a *primitive root* or *primitive element* mod p. Since all its powers up to the $(p-2)$nd are distinct, we see that every non-zero element of $\mathbb{Z}/(p)$ can be expressed as a power of a. This is very similar to a theorem we stated without proof for finite fields in Notes 5; the proof given here easily adapts to the result for finite fields as well. (Note that the integers modulo a prime do form a field; this is used in the proof.)

For example, the powers of 3 mod 7 are

$$3^1 \equiv 3, \quad 3^2 \equiv 2, \quad 3^3 \equiv 6, \quad 3^4 \equiv 4, \quad 3^5 \equiv 5, \quad 3^6 \equiv 1 \pmod 7$$

so that 3 is a primitive root of 7. But 2 is not a primitive root of 7, since $2^3 \equiv 1$.

Example We begin with a couple of examples to get the feel of the problem. Suppose that $p = 17$. Then the order of every non-zero element mod p divides 16. If there is no primitive element (one of order exactly 16), then the order of every element would divide 8. But the polynomial $x^8 - 1$ has at most 8 roots in the field $\mathbb{Z}/(17)$, so this can't be the case.

Next consider $p = 37$. The orders of all elements must divide 36; if the order of an element is smaller than 36, then it must divide either 12 or 18. But there are at most $12 + 18$ such elements (arguing as above), so there must be a primitive element.

In general we need to refine this crude counting a bit. Here is the general proof.

Proof Let a be any element with $\gcd(a,p)=1$. We define the *order* of a mod p to be the smallest positive integer m such that $a^m \equiv 1 \pmod{p}$. In the proof below, we write equality in place of congruence mod p for brevity, so that this condition will be written $a^m = 1$.

Now the order of any element divides $p-1$. For suppose that a has order m, where $p-1 = mq+r$ and $0<r<m$. Then

$$1 = a^{p-1} = (a^m)^q \cdot a^r = a^r,$$

contradicting the definition of m. So $r=0$ and m divides $p-1$.

Given a divisor m of $p-1$, how many elements of order m are there? Let $f(m)$ be this number. Now we have:

- $f(m) \leq \phi(m)$ for all m dividing $p-1$. For this is clearly true if $f(m)=0$, so suppose not. Then there exists some element a with order m. Now the elements $a^0 = 1, a^1, \ldots, a^{m-1}$ are all distinct and satisfy the polynomial equation $x^m - 1 = 0$. But a polynomial of degree m over a field has at most m roots; so these are all the roots. Now it is easy to see that a^r has order m if and only if $\gcd(r,m)=1$; so there are exactly $\phi(m)$ elements of order m in this case.

- $\sum\limits_{m|p-1} f(m) = p-1$. This is because each of the $p-1$ non-zero elements of $\mathbb{Z}/(p)$ has some order!

- $\sum\limits_{m|p-1} \phi(m) = p-1$. This follows from the fact that the number of integers a with $0 \leq a \leq p-1$ and $\gcd(a,p-1) = (p-1)/m$ is precisely $\phi(m)$, which is quite easy to see; check it for yourself using the fact that $\gcd(a,p-1) = (p-1)/m$ if and only if $\gcd(am/(p-1),m) = 1$.

From these three statements it follows that $f(m) = \phi(m)$ for all m dividing $p-1$. In particular, $f(p-1) = \phi(p-1) > 0$, so there are some elements which have order $p-1$, as required.

Our proof actually shows us a little more: the number of primitive roots of the prime p is $\phi(p-1)$. For example, $\phi(7-1) = \phi(2\cdot 3) = 2$, so there are two primitive roots of 7, namely 3 and 5.

Now it is not true that Euler's extension of the little Fermat theorem is best possible. For example, suppose that $\gcd(a,35)=1$. Then $\gcd(a,7)=1$, so $a^6 \equiv 1 \pmod{7}$. Similarly, $\gcd(a,5)=1$, so $a^4 \equiv 1 \pmod{5}$. From this we deduce that $a^{12} \equiv 1 \pmod{7}$ and $a^{12} \equiv 1 \pmod{5}$, so $a^{12} \equiv 1 \pmod{35}$. On the other hand, $\phi(35) = \phi(7)\cdot\phi(5) = 6\cdot 4 = 24$, so Euler only guarantees that $a^{24} \equiv 1 \pmod{35}$.

Carmichael's lambda-function $\lambda(n)$ is defined to be the least number m such that $a^m \equiv 1 \pmod n$ for all a such that $\gcd(a,n) = 1$. It follows from Euler and the argument we used above that $\lambda(n)$ always divides $\phi(n)$, but it may be strictly smaller; for example, $\phi(35) = 24$ but $\lambda(35) = 12$. (We can see that $\lambda(35)$ cannot be less than 12 since, for example, $2^6 \equiv 29 \pmod{35}$ and $2^4 \equiv 16 \pmod{35}$.)

Theorem 22 *(a) If $n = p_1^{a_1} \cdots p_r^{a_r}$, where p_i are distinct primes and $a_i > 0$, then $\lambda(n) = \operatorname{lcm}\{\lambda(p_1^{a_1}), \ldots, \lambda(p_r^{a_r})\}$.*

(b) If p is an odd prime and $a > 0$, then $\lambda(p^a) = \phi(p^a) = p^a - p^{a-1} = p^{a-1}(p-1)$.

(c) $\lambda(2) = 1$, $\lambda(4) = 2$, and $\lambda(2^a) = 2^{a-2}$ for $a \geq 3$.

The fact that $\lambda(p) = p-1$ for all primes p is a consequence of Theorem 21. Fermat tells us that $a^{p-1} \equiv 1 \pmod p$ for all a coprime to p, and the theorem tells us that no smaller exponent will do.

Suppose that n is the product of two distinct primes, say $n = pq$. The theorem asserts that $\lambda(n) = \operatorname{lcm}(p-1, q-1)$. To show this, let $m = \operatorname{lcm}(p-1, q-1)$. Now, for any integer x coprime to n, we have $x^{p-1} \equiv 1 \pmod p$, and so $x^m \equiv 1 \pmod p$, since $p-1$ divides m. Similarly $x^m \equiv 1 \pmod q$. By the Chinese Remainder Theorem, $x^m \equiv 1 \pmod n$.

In the converse direction, suppose that a is primitive element mod p, and b a primitive element mod q. Use the Chinese Remainder Theorem to find c such that

$$c \equiv a \pmod p, \qquad c \equiv b \pmod q.$$

Then it is easy to see that the order of c mod n is a multiple both of $p-1$ and of $q-1$, and hence of m. So m is the smallest positive number such that $x^m \equiv 1 \pmod n$ for all x coprime to n; that is, $\lambda(n) = m$.

We will not need the other cases of the above theorem.

For example, we have $\lambda(35) = \operatorname{lcm}(\lambda(7), \lambda(5)) = \operatorname{lcm}(6,4) = 12$, as we found earlier.

Power maps

Now consider the transformation

$$T_e : x \mapsto x^e \bmod n.$$

First, we shall simply consider this transformation acting on the set

$$U(n) = \{x \in \mathbb{Z}/(n) : \gcd(x,n) = 1\}$$

of x with $\gcd(x,n) = 1$. (Note that if $\gcd(x,n) = 1$ then $\gcd(x^e,n) = 1$ for all e.)

Proposition 23 *The transformation T_e is one-to-one on $U(n)$ if and only if e satisfies* $\gcd(e,\lambda(n)) = 1$*. So the number of e for which T_e is one-to-one is $\phi(\lambda(n))$.*

We will prove this just in the reverse direction. Suppose that $\gcd(e,\lambda(n)) = 1$. Then there exists d with $de \equiv 1 \pmod{\lambda(n)}$. Then, since $x^{\lambda(n)} = 1$, we have $x^{de} = x$ for all x coprime to n; that is, $T_d T_e$ is the identity map, and so T_e has an inverse.

For example, for $n = 35$, the invertible maps are T_1, T_5, T_7 and T_{11}. The map T_{13} is equal to T_1 on $U(35)$ since $x^{12} \equiv 1 \pmod{35}$ for any $x \in U(35)$.

This condition is not sufficient for T_e to be one-to-one on the whole of $\mathbb{Z}/(n)$. For example, take $n = 9$. Then $\lambda(n) = \phi(n) = 6$, and the number $e = 5$ satisfies $\gcd(e,\lambda(n)) = 1$. Now the fifth powers mod 9 are given in the following table:

x	0	1	2	3	4	5	6	7	8
$x^5 \bmod 9$	0	1	5	0	7	2	0	4	8

So, in accordance with Proposition 23, T_5 is one-to-one on $\{1,2,4,5,7,8\}$ (the numbers coprime to 9); but it maps all the others to zero.

However, there is a special case where we can guarantee that T_e is invertible on $\mathbb{Z}/(n)$:

Proposition 24 *Let n be the product of distinct primes. If* $\gcd(e,\lambda(n)) = 1$*, then $T_e : x \mapsto x^e \bmod n$ is one-to-one on $\mathbb{Z}/(n)$.*

Here is the proof in the case that n is the product of two primes. (This is the only case that is required for RSA, but the proof can be modified to work in general.)

We use the fact that $x^p \equiv x \pmod{p}$ for any prime p. (If p doesn't divide x, this follows from Fermat's little theorem; if $p \mid x$ it is trivial.) Hence $x^{k(p-1)+1} \equiv x \pmod{p}$ for any $k > 0$.

Now, if d is the inverse of $e \bmod \lambda(n)$, then $de \equiv 1 \pmod{\lambda(n)}$, and hence $de \equiv 1 \pmod{p-1}$, since $p-1$ divides $\lambda(n)$. From the preceding paragraph, we see that $x^{de} \equiv x \pmod{p}$. Similarly $x^{de} \equiv x \pmod{q}$, and so $x^{de} \equiv x \pmod{n}$, by the Chinese Remainder Theorem.

For example, suppose that $n = 15$. Then $\lambda(n) = \mathrm{lcm}(2,4) = 4$, and we can choose $e = 3$. The table of cubes mod 15 is:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$x^3 \bmod 15$	0	1	8	12	4	5	6	13	2	9	10	11	3	7	14

We see that T_3 is indeed one-to-one.

The RSA cryptosystem

Preliminaries

The system depends on the following problems. The easy problems are all in P. Unfortunately the hard problems are not known to be NP-complete!

Easy problems

(1) Test whether an integer N is prime.

(2) Given a and n, find $\gcd(a,n)$ and (if it is 1) find an inverse of a mod n.

(3) Calculate the transformation $T_e : x \mapsto x^e \bmod N$.

Hard problems

(4) Given an integer N, factorise it into its prime factors.

(5) Given an integer N, calculate $\lambda(N)$ (or $\phi(N)$).

(6) Given N and e, find d such that T_d is the inverse of T_e mod N.

Notes about the easy problems My job is to persuade you that they are easy, not to give formal proofs that they belong to the class P.

Problem (1): Note that trial division does not solve this problem efficiently. For a number N requiring n bits of input is one which has n digits when written in base 2, and hence is of size roughly 2^n; its square root is about $2^{n/2}$, and trial division would require about half this many steps in the worst case. Only in 2002 was an algorithm found which solves this problem in a polynomial number of steps, by Manindra Agrawal, Neeraj Kayal and Nitin Saxena at the Indian Institute of Technology, Kanpur. However, the result had been widely expected, since 'probabilistic' algorithms which tested primality with an arbitrarily small chance of giving an incorrect answer have been known for some time. We will consider the question of primality testing further in the next part of the notes.

Problem (2): This is solved by Euclid's algorithm, as we have seen.

Problem (3). On the face of it, this problem seems hard, for two reasons:

- First, the number x^e will be absolutely vast, with about $e \log x$ digits (and remember that the number of digits of e is part of the size of the input; if e has 100 digits, then x^e has too many digits to write down even if the whole universe were our scrap paper).

- Second, we have on the face of it to perform $e-1$ multiplications to find

$$x^e = x \cdot x \cdot x \cdots x \qquad e \text{ factors.}$$

But these difficulties can both be overcome:

Proposition 25 *The number a^e mod n can be computed with at most $2\log_2 e$ multiplications of numbers smaller than n and the same number of divisions by n; this can be done in a polynomial number of steps.*

The first difficulty is easily dealt with: we do all our calculations mod n. Thus, to calculate ab mod n, where $a,b < n$, we calculate ab as an integer, and take the remainder on division by n. We never have to deal with a number larger than n^2 in the calculation.

We can reduce this number of multiplications required from $e-1$ to at most $2\log_2 e$ as follows.

Write e in base 2: $e = 2^{a_1} + 2^{a_2} + \cdots + 2^{a_k}$. Suppose that a_1 is the greatest exponent. Then $k \leq a_1 + 1$ and $a_1 \leq \log_2 e$.

By $a_1 - 1$ successive squarings, calculate $x^2, x^{2^2}, \ldots, x^{2^{a_1}}$.

Now

$$x^e = x^{2^{a_1}} \cdot x^{2^{a_2}} \cdots x^{2^{a_k}}$$

can be obtained by $k-1$ further multiplications. The total number of multiplications required is $a_1 + k - 2 < 2\log_2 e$.

This informal description of the algorithm can be translated into a formal proof that problem (3) can be solved in polynomial time.

For example, let us compute $123^{321} \pmod{557}$.

First we find by successive squaring

i	0	1	2	3	4	5	6	7	8
2^i	1	2	4	8	16	32	64	128	256
123^{2^i} mod 557	123	90	302	413	127	533	19	361	540

Now $321 = 2^8 + 2^6 + 1$, so two further multiplications mod 557 give

$$123^{321} = 123^{256} \cdot 123^{64} \cdot 123 \equiv 540 \cdot 19 \cdot 123 \equiv 234 \cdot 123 \equiv 375 \pmod{557}.$$

Notes about the hard problems Problems (4)–(6) are not known to be NP-complete, so it is possible that they may not be as hard as we would like. However, centuries of work by mathematicians has failed to discover any 'easy' algorithm to factorise large numbers. (We will see later that the advent of quantum computation would change this assertion!)

We will be concerned only with numbers N which are the product of two distinct primes p and q. So we really need the special case of (4) which asks:

> Given a number N which is known to be the product of two distinct prime factors, find the factors.

Even this problem is intractable at present.

However, if we know that N is the product of two distinct primes, then problems (4) and (5) are equivalent, in the sense that knowledge of a solution to one enables us to solve the other.

Proposition 26 *Suppose that N is the product of two distinct primes. Then, from any one of the following pieces of information, we can compute the others in a polynomial number of steps:*

- *the prime factors of N;*
- $\phi(N)$*;*
- $\lambda(N)$.

For suppose first that $N = pq$ where p and q are primes (which we know). Then $\phi(N) = (p-1)(q-1)$ can be found by simple arithmetic. Also, $\lambda(N) = \mathrm{lcm}(p-1, q-1) = (p-1)(q-1)/\gcd(p-1, q-1)$; the greatest common divisor can be found by Euclid's Algorithm, and the rest is arithmetic.

Suppose that we know $\phi(N)$. Then we know the sum and product of p and q, (namely, $p+q = N-\phi(N)+1$ and $pq = N$); and so the two factors are roots of the quadratic equation

$$x^2 - (N - \phi(N) + 1)x + N = 0,$$

which can be solved by arithmetic (using the standard algorithm for finding the square root).

The case where we know N and $\lambda(N)$ is a bit more complicated. Suppose that p is the larger prime factor. Then $\lambda(N) = \mathrm{lcm}(p-1, q-1)$ is a multiple of $p-1$, and divides $\phi(N)$. Let $r \equiv N \bmod \lambda(N)$ be the remainder on dividing N by $\lambda(N)$. Then

- $N - \phi(N) \equiv r \pmod{\lambda(N)}$, since $\lambda(N) \mid \phi(N)$;

- $N-\phi(N) = p+q-1 < 2\lambda(N)$, since $\lambda(N) \geq p-1 > q$ (assuming that $N > 6$).

So $N-\phi(N) = r$ or $N-\phi(N) = r+\lambda(N)$. We can solve the quadratic for each of these two possible values of $\phi(N)$; one of them will give us the factors of N.

Example Suppose that $N = 589$ and $\lambda(N) = 90$. Now 589 mod 90 = 49. Trying $\phi(N) = 540$, we get that the prime factors of N are the roots of the quadratic

$$x^2 - 50x + 589 = 0,$$

so that

$$p,q = 25 \pm \sqrt{625-589} = 25 \pm 6 = 31, 19.$$

There is no need to try the other case.

Example Suppose that $N = 21$ and $\lambda(N) = 6$. Then $N-\phi(N) = 3$ or 9. In the first case the quadratic is $x^2 - 4x + 21 = 0$, which has imaginary roots. In the second, it is $x^2 - 10x + 21 = 0$, with roots 3 and 7. Note that we only need the second case if $q-1$ divides $p-1$, since otherwise $\lambda(N) \geq 2(p-1)$.

Remark In fact we have $2\lambda(N) \mid \phi(N)$ (why?) so we can calculate $N \equiv N - \phi(N)$ mod $2\lambda(N)$ and hence work out in advance which of the two cases is the right one.

Finally, we remark that, if $\phi(N)$ or $\lambda(N)$ is known, then problem (6) is easy. For we choose e to be the inverse of d mod $\lambda(N)$, using Euclid's Algorithm.

In the other direction, if we know a solution to problem (6) (that is, if we know d and e such that T_d is the inverse of T_e mod N), we can often factorise N. The algorithm is as follows. We assume that N is the product of two primes (neither of them being 2).

> Let $de - 1 = 2^a \cdot b$, where b is odd. Choose a random x with $0 < x < N$.
>
> First, calculate $\gcd(x,N)$. If this is not 1, we've found a factor already and we can stop.
>
> If $\gcd(x,N) = 1$, we proceed as follows. Let $y = x^b \bmod N$. If $y \equiv \pm 1 \pmod{N}$, the algorithm has failed. Repeatedly replace y by y^2 mod N (remembering the preceding value of y – more formally, $z := y$ and $y := y^2$ mod N) until $y \equiv \pm 1 \pmod{N}$.
>
> If $y \equiv -1 \pmod{N}$, the algorithm has failed.
>
> However, if $y \equiv 1 \pmod{N}$, then we have found z such that $z^2 \equiv 1 \pmod{N}$ and $z \not\equiv \pm 1 \pmod{N}$. Then $\gcd(N, z+1)$ and $\gcd(N, z-1)$ are the prime factors of N.

Remarks:

- The chance that $\gcd(x,N) \neq 1$ is very remote. However, we should make this test, since the rest of the algorithm depends on the assumption that the gcd is 1.

- The loop where we do $z := y$ and $y := y^2 \bmod N$ will be repeated at most a times. For we know that $\lambda(N)$ divides $de-1$, so that $x^{de} \equiv x \pmod{N}$. Since x is coprime to N, it has an inverse, and so $x^{de-1} \equiv 1 \pmod{N}$. But $x^{de-1} = x^{2^a \cdot b} \equiv y^{2^a}$, where $x \equiv x^b$, so after a successive squarings we certainly have 1; the loop will terminate no later than this step.

- If $z^2 \equiv 1 \pmod{N}$, then N divides $z^2 - 1 = (z+1)(z-1)$. Both the factors lie between 1 and $N-1$, so $\gcd(N, z+1)$ and $\gcd(N, z-1)$ are proper divisors of N. They are coprime, so they must be the two prime factors of N.

- It can be shown that, choosing x randomly, the probability that the algorithm succeeds in factorising N is approximately $1/2$. So, by repeating a number of times with different random choices of x if necessary, we can be fairly sure of finding the factorisation of N.

- This algorithm is closely related to the Miller–Rabin test for primality. Suppose we run the algorithm with N being prime. Then it must *always* produce the answer $y \equiv -1 \pmod{N}$. But if N is not prime, then we *expect* to get different values for y at least half of the time. Therefore, if we repeat the experiment many times, with different values of x, and always get $y \equiv -1 \pmod{N}$, then we take this as strong evidence that N is prime. Of course, this is a statistical argument, not a mathematical one! It is not hard to find examples of composite numbers which the Miller–Rabin test thinks are prime.

Example Suppose that $N = 589$ and we are told that the private exponent corresponding to $e = 7$ is $d = 13$. Now $de - 1 = 90 = 2 \cdot 45$. Apply the algorithm with $x = 2$. We do have $\gcd(2, 589) = 1$. Now $y = 2^{45} \bmod 589 = 94$. At the next stage, $z = 94$ and $y = z^2 \bmod 589 = 1$. So the factors of 589 are $\gcd(589, 95) = 19$ and $\gcd(589, 93) = 31$ (these gcds are found by Euclid's algorithm).

Implementation

Bob chooses two large prime numbers p_B and q_B. This involves a certain amount of randomness. It is known that a fraction of about $1/(k \ln 10)$ of k-digit numbers are prime. Thus, if Bob repeatedly chooses a random k-digit number and tests it for primality, in mk trials the probability that he has failed to find a prime number is exponentially small (as a function of m). Each primality test takes only a polynomial number of steps. The chances of success at each trial can be doubled by the obvious step of choosing only odd numbers; and excluding other small prime divisors such as 3 improve the chances still further. We conclude that in a polynomial number of steps (in terms of k), Bob will have found two primes, with an exponentially small probability of failure.

Knowing p_B and q_B, Bob computes their product $N_B = p_B q_B$. He can also compute $\lambda(N_B) = \mathrm{lcm}(p_B - 1, q_B - 1)$. He now computes a large 'exponent' e_B satisfying $\gcd(e_B, \lambda(N_B)) = 1$ (again by choosing a random e_B and using Euclid's algorithm). The application of Euclid's algorithm also gives the inverse of e_B mod $\lambda(N_B)$, that is the number such that T_{d_B} is the inverse of T_{e_B}, where

$$T_{e_B} : x \mapsto x^{e_B} \pmod{N_B}.$$

Proposition 24 shows that the maps are inverses on all of $\mathbb{Z}/(N_B)$, since N_B is the product of two primes.

Bob publishes N_B and e_B, and keeps the factorisation of N_B and the number d_B secret.

If Alice wishes to send a message to Bob, she first transforms her message into a number x less than N_B. (For example, if the message is a binary string, break it into blocks of length k, where $2^k < N_B$, and regard each block as an integer in the interval $[0, 2^k - 1]$ written to the base 2. Now she computes $z = T_{e_B}(x)$ and sends this to Bob.

Bob deciphers the message by applying the inverse function T_{d_B} to it. This gives a number less than N_B and congruent to x mod N_B. Since x is also less than N_B, the resulting decryption is correct.

If Eve intercepts the message z, she has to compute $T_{d_B}(z)$, which is a hard problem (problem (6) above). Alternatively, she could compute d_B from the published value of e_B. Since d_B is the inverse of e_B mod $\lambda(N_B)$, this requires her to calculate $\lambda(N_B)$, which is also hard (problem (5)). Finally, she could try to factorise N_B: this, too, is hard (problem (4)). So the cipher is secure.

Since the plaintext and ciphertext are both integers smaller than N_B, and the encryption function is a bijection, the RSA system supports digital signatures.

If Alice and Bob have both chosen a key, then Alice can sign her message to Bob by the method for digital signatures that we described earlier. That is, Alice 'decrypts' with her secret key T_{d_A} before encrypting with Bob's public key; after decrypting, Bob then 'encrypts' with Alice's public key to get the authenticated message.

Remark We saw that, if we know d and e such that T_d is the inverse of T_e mod N, then we have a very good chance of factorising N. The moral of this is that, if your RSA key is broken (that is, if Eve comes to know both d and e), it is not enough to keep the same N and choose different d and e, since you must assume that Eve now knows the factors of N. You must begin again with a different choice of two primes p and q.