Number theory and (public key) cryptography.

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

For this, Bob needs to have a way to encrypt his message, and Alice needs a way to decrypt it. Normally, it seems they will need to agree beforehand on how to do this.

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

For this, Bob needs to have a way to encrypt his message, and Alice needs a way to decrypt it. Normally, it seems they will need to agree beforehand on how to do this. This is both risky (the agreement can be intercepted) and cumbersome.

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

For this, Bob needs to have a way to encrypt his message, and Alice needs a way to decrypt it. Normally, it seems they will need to agree beforehand on how to do this. This is both risky (the agreement can be intercepted) and cumbersome. (Think of Bob as a customer and Alice as a bank, and the work involved if every customer has to agree on a 'key' with the bank.)

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

For this, Bob needs to have a way to encrypt his message, and Alice needs a way to decrypt it. Normally, it seems they will need to agree beforehand on how to do this. This is both risky (the agreement can be intercepted) and cumbersome. (Think of Bob as a customer and Alice as a bank, and the work involved if every customer has to agree on a 'key' with the bank.)

Public key cryptography circumvents this problem: The receiver (the bank, Alice) decides on the method in a way that we shall describe. Alice tells Bob, and the whole world, how to encrypt, but keeps the *decryption key* for herself.

We have two actors: The *sender* wants to send an encrypted message to the *receiver*. In popular accounts (like here) the sender is usually called Bob, and the receiver is called Alice, so Bob wants to send a message to Alice.

For this, Bob needs to have a way to encrypt his message, and Alice needs a way to decrypt it. Normally, it seems they will need to agree beforehand on how to do this. This is both risky (the agreement can be intercepted) and cumbersome. (Think of Bob as a customer and Alice as a bank, and the work involved if every customer has to agree on a 'key' with the bank.)

Public key cryptography circumvents this problem: The receiver (the bank, Alice) decides on the method in a way that we shall describe. Alice tells Bob, and the whole world, how to encrypt, but keeps the *decryption key* for herself. Bob encrypts his message. After that, only Alice can read it, because only she has the key.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock. She sends the box and the **open** padlock to Bob, but keeps the key. Bob puts his message in the box and closes with the padlock.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock. She sends the box and the **open** padlock to Bob, but keeps the key. Bob puts his message in the box and closes with the padlock. After that he sends the closed box to Alice, who is the only one that can open it.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock. She sends the box and the **open** padlock to Bob, but keeps the key. Bob puts his message in the box and closes with the padlock. After that he sends the closed box to Alice, who is the only one that can open it.

This is a good way to remember the general idea, but of course no boxes are used. The boxes are replaced by a mathematical procedure.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock. She sends the box and the **open** padlock to Bob, but keeps the key. Bob puts his message in the box and closes with the padlock. After that he sends the closed box to Alice, who is the only one that can open it.

This is a good way to remember the general idea, but of course no boxes are used. The boxes are replaced by a mathematical procedure.

There are now many methods for public key cryptography. They are all based on the fact that some mathematical operations that are easy to carry out may be very hard to reverse.

Baby version

The real baby version of the procedure is as follows: Alice has a box and a padlock. She sends the box and the **open** padlock to Bob, but keeps the key. Bob puts his message in the box and closes with the padlock. After that he sends the closed box to Alice, who is the only one that can open it.

This is a good way to remember the general idea, but of course no boxes are used. The boxes are replaced by a mathematical procedure.

There are now many methods for public key cryptography. They are all based on the fact that some mathematical operations that are easy to carry out may be very hard to reverse.

History

One example is multiplication (relatively easy), versus factorization (very hard).

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799?

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know.”

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know.” (The answer is $8616460799 = 89681 \cdot 96079$.)

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know.” (The answer is $8616460799 = 89681 \cdot 96079$.)

The RSA-method that I will describe was published by R Rivest, A Shamir and L Adelman in 1977.

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know.” (The answer is $8616460799 = 89681 \cdot 96079$.)

The RSA-method that I will describe was published by R Rivest, A Shamir and L Adelman in 1977. It was first discovered by C Cocks in 1973, but was immediately classified for 24 years. This was probably the first version of public key or assymetric encryption.

History

One example is multiplication (relatively easy), versus factorization (very hard).

In 1874, the British economist and logician W S Jevons wrote: “Can the reader say what two numbers multiplied together will produce the number 8616460799? I think it unlikely that anyone but myself will ever know.” (The answer is $8616460799 = 89681 \cdot 96079$.)

The RSA-method that I will describe was published by R Rivest, A Shamir and L Adelman in 1977. It was first discovered by C Cocks in 1973, but was immediately classified for 24 years. This was probably the first version of public key or assymetric encryption.

We now turn to the mathematics behind it, which is precisely what is illustrated by Jevons’ example: It is easy to multiply two big prime numbers, but very hard to find these prime numbers if you only know their product.

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Example: $\phi(6) = 2$ since only 1 and 5 are relatively prime to 6.

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Example: $\phi(6) = 2$ since only 1 and 5 are relatively prime to 6.
$\phi(7) = 6$ since 1,2,3,4,5,6 are relatively prime to 7.

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Example: $\phi(6) = 2$ since only 1 and 5 are relatively prime to 6. $\phi(7) = 6$ since 1,2,3,4,5,6 are relatively prime to 7. In fact $\phi(p) = p - 1$ if p is prime.

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Example: $\phi(6) = 2$ since only 1 and 5 are relatively prime to 6. $\phi(7) = 6$ since 1,2,3,4,5,6 are relatively prime to 7. In fact $\phi(p) = p - 1$ if p is prime.

Proposition

Let $n = pq$ where both p and q are prime. Then

$$\phi(n) = (p - 1)(q - 1).$$

Euler totient function

Let n be a natural number. Then $\phi(n)$, the *Euler totient function* is defined as the number of a:s less than n that are relatively prime to n, i e have no common factors with n.

Example: $\phi(6) = 2$ since only 1 and 5 are relatively prime to 6. $\phi(7) = 6$ since 1,2,3,4,5,6 are relatively prime to 7. In fact $\phi(p) = p - 1$ if p is prime.

Proposition

Let $n = pq$ where both p and q are prime. Then

$$\phi(n) = (p - 1)(q - 1).$$

In fact, if n and m are relatively prime, then

$$\phi(nm) = \phi(n)\phi(m).$$

I can't resist giving a proof of this, based on the *Chinese remainder theorem*.

I can't resist giving a proof of this, based on the *Chinese remainder theorem*.

Theorem

Let n_1 and n_2 be relatively prime, and let a_1, a_2 be arbitrary integers. Then there is an integer x such that

$$x = a_1 mod(n_1)$$

and

$$x = a_2 mod(n_2).$$

To find x, solve

$a_1 - a_2 = tn_1 + sn_2.$

I can't resist giving a proof of this, based on the *Chinese remainder theorem*.

Theorem

Let n_1 and n_2 be relatively prime, and let a_1, a_2 be arbitrary integers. Then there is an integer x such that

$$x = a_1 mod(n_1)$$

and

$$x = a_2 mod(n_2).$$

To find x, solve

$a_1 - a_2 = tn_1 + sn_2$. Always possible since n_1 and n_2 are relatively prime.

I can't resist giving a proof of this, based on the *Chinese remainder theorem*.

Theorem

Let n_1 and n_2 be relatively prime, and let a_1, a_2 be arbitrary integers. Then there is an integer x such that

$$x = a_1 mod(n_1)$$

and

$$x = a_2 mod(n_2).$$

To find x, solve

$a_1 - a_2 = tn_1 + sn_2$. Always possible since n_1 and n_2 are relatively prime.

Put

$$x = a_1 - tn_1 = a_2 + sn_2.$$

Then x solves the problem. (Why?)

Moreover solutions are almost unique: If y is another solution to the equation, then $x - y$ is divisible by nm.

Moreover solutions are almost unique: If y is another solution to the equation, then $x - y$ is divisible by nm.

If x and y solve the equation then $x - y$ is divisble by n and m, hence by nm since n and m are relatively prime. This proves uniqueness.

Moreover solutions are almost unique: If y is another solution to the equation, then $x - y$ is divisible by nm.

If x and y solve the equation then $x - y$ is divisble by n and m, hence by nm since n and m are relatively prime. This proves uniqueness.

Let $1 \leq a_1 < n_1$ and $1 \leq a_2 < n_2$, and let x be the unique solution to the Chinese remainder problem such that $1 \leq x < n_1 n_2$.

Moreover solutions are almost unique: If y is another solution to the equation, then $x - y$ is divisible by nm.

If x and y solve the equation then $x - y$ is divisble by n and m, hence by nm since n and m are relatively prime. This proves uniqueness.

Let $1 \leq a_1 < n_1$ and $1 \leq a_2 < n_2$, and let x be the unique solution to the Chinese remainder problem such that $1 \leq x < n_1 n_2$. One checks that x is relatively prime to $n_1 n_2$ if and only if a_1 is relatively prime to n_1 and a_2 is relatively prime to n_2.

Moreover solutions are almost unique: If y is another solution to the equation, then $x - y$ is divisible by nm.

If x and y solve the equation then $x - y$ is divisble by n and m, hence by nm since n and m are relatively prime. This proves uniqueness.

Let $1 \leq a_1 < n_1$ and $1 \leq a_2 < n_2$, and let x be the unique solution to the Chinese remainder problem such that $1 \leq x < n_1 n_2$. One checks that x is relatively prime to $n_1 n_2$ if and only if a_1 is relatively prime to n_1 and a_2 is relatively prime to n_2. So, the map

$$p : (a_1, a_2) \to x$$

is a bijection between the set of pairs (a_1, a_2) relatively prime to n_1 and n_2 respectively and the set of x relatively prime to $n_1 n_2$. This proves the proposition

$$\phi(n_1 n_2) = \phi(n_1)\phi(n_2).$$

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$.

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers.

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers. Then there is an integer x such that

$$x = a_i mod(n_i) \quad i = 1, 2...p.$$

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers. Then there is an integer x such that

$$x = a_i mod(n_i) \quad i = 1, 2...p.$$

Moreover, x is unique modulo N.

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers. Then there is an integer x such that

$$x = a_i mod(n_i) \quad i = 1, 2...p.$$

Moreover, x is unique modulo N.

Exercise: Prove this!

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers. Then there is an integer x such that

$$x = a_i mod(n_i) \quad i = 1, 2...p.$$

Moreover, x is unique modulo N.

Exercise: Prove this! *Hint: Induction. First solve the first two equations and call the solution $x = a_{12}$. Then replace the first two equations by $x = a_{12} mod(n_1 n_2)$ to get a new system with $p - 1$ equations.*

General version of Chinese remainder theorem

Probably, the general version is easier to remember:

Theorem

Let $n_1, n_2, ...n_p$ be relatively prime and let $N = n_1 n_2 ... n_p$. Let $a_1, a_2, ...a_p$ be arbitrary integers. Then there is an integer x such that

$$x = a_i mod(n_i) \quad i = 1, 2...p.$$

Moreover, x is unique modulo N.

Exercise: Prove this! *Hint: Induction. First solve the first two equations and call the solution $x = a_{12}$. Then replace the first two equations by $x = a_{12} mod(n_1 n_2)$ to get a new system with $p - 1$ equations.*

The Chinese remainder theorem was used by Gödel in his famous incompleteness theorem to encode a sequence of numbers ($a_1, a_2, ...a_p$) by one number (x).

Theorem

If m and n are relatively prime, then

$$m^{\phi(n)} = 1 mod(n).$$

In particular, if p is prime and p does not divide m, then

$$m^{p-1} = 1 mod(p).$$

Theorem

If m and n are relatively prime, then

$$m^{\phi(n)} = 1 mod(n).$$

In particular, if p is prime and p does not divide m, then

$$m^{p-1} = 1 mod(p).$$

The last part is called *Fermat's little theorem*.

For the proof we need a lemma:

Lemma

Let m be relatively prime to n. Then there is a number m' such that

$$mm' = 1 mod(n).$$

Bevis.

Look at all the numbers km where k is relatively prime to n too, and $1 \leq k < n$. . They are all relatively prime to n and they are all different modulo n. So, one of them must be equal to 1 modulo n. □

Proof of theorem

Let $1 = k_1 < k_2 ... k_{\phi(n)}$ be all the numbers less than n that are relatively prime to n. Look at

$$mk_1, mk_2, ... mk_{\phi(n)}.$$

Proof of theorem

Let $1 = k_1 < k_2 ... k_{\phi(n)}$ be all the numbers less than n that are relatively prime to n. Look at

$$mk_1, mk_2, ...mk_{\phi(n)}.$$

If $i < j$, then $mk_j - mk_i = m(k_j - k_i)$ is never divisible by n since m does not contain any divisor of n and $k_i - k_j$ is not divisible by n since k_i and k_j are different modulo n.

Proof of theorem

Let $1 = k_1 < k_2 ... k_{\phi(n)}$ be all the numbers less than n that are relatively prime to n. Look at

$$mk_1, mk_2, ...mk_{\phi(n)}.$$

If $i < j$, then $mk_j - mk_i = m(k_j - k_i)$ is never divisible by n since m does not contain any divisor of n and $k_i - k_j$ is not divisible by n since k_i and k_j are different modulo n. So all numbers mk_i are different modulo n. Moreover all mk_i are relatively prime to n. Therefore

$$\{k_1, k_2, ...k_{\phi(n)}\} = \{mk_1, mk_2, ...mk_{\phi(n)}\}$$

if we only look at their residue classes modulo n.

Proof of theorem

Let $1 = k_1 < k_2 ... k_{\phi(n)}$ be all the numbers less than n that are relatively prime to n. Look at

$$mk_1, mk_2, ...mk_{\phi(n)}.$$

If $i < j$, then $mk_j - mk_i = m(k_j - k_i)$ is never divisible by n since m does not contain any divisor of n and $k_i - k_j$ is not divisible by n since k_i and k_j are different modulo n. So all numbers mk_i are different modulo n. Moreover all mk_i are relatively prime to n. Therefore

$$\{k_1, k_2, ...k_{\phi(n)}\} = \{mk_1, mk_2, ...mk_{\phi(n)}\}$$

if we only look at their residue classes modulo n. Hence

$$\prod k_i = m^{\phi(n)} \prod k_i mod(n).$$

Since $\prod k_i$ is relatively prime to n, we can divide by it (by the lemma) and get

$$1 = m^{\phi(n)} mod(n).$$

Example

The proof becomes more clear in an example:

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4.

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4. Hence $\phi(10) = 4$.

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4. Hence $\phi(10) = 4$. Take $m = 7$. Then

$\{mk_1, mk_2, mk_3, mk_4\} = \{7{\cdot}1, 7{\cdot}3, 7{\cdot}7, 7{\cdot}9\} =$

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4. Hence $\phi(10) = 4$. Take $m = 7$. Then

$$\{mk_1, mk_2, mk_3, mk_4\} = \{7{\cdot}1, 7{\cdot}3, 7{\cdot}7, 7{\cdot}9\} = \{7, 1, 9, 3\} = \{k_1, k_2, k_3, k_4\}.$$

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4. Hence $\phi(10) = 4$. Take $m = 7$. Then

$$\{mk_1, mk_2, mk_3, mk_4\} = \{7{\cdot}1, 7{\cdot}3, 7{\cdot}7, 7{\cdot}9\} = \{7, 1, 9, 3\} = \{k_1, k_2, k_3, k_4\}.$$

So $m' =' 1/7' = 3$.

Example

The proof becomes more clear in an example:

Let $n = 10$. The numbers less than n that are relatively prime to n are 1,3,7,9= k_1, k_2, k_3, k_4. Hence $\phi(10) = 4$. Take $m = 7$. Then

$\{mk_1, mk_2, mk_3, mk_4\} = \{7{\cdot}1, 7{\cdot}3, 7{\cdot}7, 7{\cdot}9\} = \{7, 1, 9, 3\} = \{k_1, k_2, k_3, k_4\}$.

So $m' =' 1/7' = 3$.

And

$$7^4 = 49^2 = 9^2 = 81 = 1.$$

Cryptography

Now it is time to get back to cryptography. The customer is going to send an encrypted message to the bank. The bank sets up the scheme: They take two large prime numbers p and q, typically between 100 and 1000 digits. Then they form $n = pq$ which is of course easy. The point is that given n, and even knowing that it is the product of only two primes, it is hard to find the two primes – even for a computer.

Cryptography

Now it is time to get back to cryptography. The customer is going to send an encrypted message to the bank. The bank sets up the scheme: They take two large prime numbers p and q, typically between 100 and 1000 digits. Then they form $n = pq$ which is of course easy. The point is that given n, and even knowing that it is the product of only two primes, it is hard to find the two primes – even for a computer.

Then the bank chooses some number e between 1 and $n - 1$, which is relatively prime to $\phi(n)$. Then, by the lemma, there is a number d such that $ed = 1$ modulo $\phi(n)$. Note:

1. Now we are working modulo $\phi(n)$, not modulo n.

Cryptography

Now it is time to get back to cryptography. The customer is going to send an encrypted message to the bank. The bank sets up the scheme: They take two large prime numbers p and q, typically between 100 and 1000 digits. Then they form $n = pq$ which is of course easy. The point is that given n, and even knowing that it is the product of only two primes, it is hard to find the two primes – even for a computer.

Then the bank chooses some number e between 1 and $n-1$, which is relatively prime to $\phi(n)$. Then, by the lemma, there is a number d such that $ed = 1$ modulo $\phi(n)$. Note:

1. Now we are working modulo $\phi(n)$, not modulo n.

2. It is very hard to find $\phi(n)$ if you know n.

Cryptography

Now it is time to get back to cryptography. The customer is going to send an encrypted message to the bank. The bank sets up the scheme: They take two large prime numbers p and q, typically between 100 and 1000 digits. Then they form $n = pq$ which is of course easy. The point is that given n, and even knowing that it is the product of only two primes, it is hard to find the two primes – even for a computer.

Then the bank chooses some number e between 1 and $n - 1$, which is relatively prime to $\phi(n)$. Then, by the lemma, there is a number d such that $ed = 1$ modulo $\phi(n)$. Note:

1. Now we are working modulo $\phi(n)$, not modulo n.

2. It is very hard to find $\phi(n)$ if you know n. In fact, $\phi(n) = (p - 1)(q - 1) = pq - p - q + 1 = n - p - q - 1$. So, if you know $\phi(n)$, then you know $p + q$. But, you already know pq. From these two equations you can solve for p and q. So, finding $\phi(n)$ is as hard as to find p and q.

Now the bank sends e and n to the customer, and it does not matter if other people can see what they are; e and n are *public*. The number e is called the *encryption key*.

Now the bank sends e and n to the customer, and it does not matter if other people can see what they are; e and n are *public*. The number e is called the *encryption key*. On the other hand, the bank keeps d, the *decryption key*, secret.

Now the bank sends e and n to the customer, and it does not matter if other people can see what they are; e and n are *public*. The number e is called the *encryption key*. On the other hand, the bank keeps d, the *decryption key*, secret.

The customer now wants to send a message to the bank. The message is a number m between 1 and n which is relatively prime to n. (Most numbers are). Then the customer computes

$$m^e = a$$

modulo n and gets a number between 1 and $n - 1$. This is the encrypted message, which is sent to the bank.

Now the bank sends e and n to the customer, and it does not matter if other people can see what they are; e and n are *public*. The number e is called the *encryption key*. On the other hand, the bank keeps d, the *decryption key*, secret.

The customer now wants to send a message to the bank. The message is a number m between 1 and n which is relatively prime to n. (Most numbers are). Then the customer computes

$$m^e = a$$

modulo n and gets a number between 1 and $n - 1$. This is the encrypted message, which is sent to the bank.

How does the bank decrypt it to find m?

Using the Euler-Fermat theorem it's easy! The bank computes

$$a^d$$

where d is the decryption key, which satisfies $ed = k\phi(n) + 1$.

Using the Euler-Fermat theorem it's easy! The bank computes

$$a^d$$

where d is the decryption key, which satisfies $ed = k\phi(n) + 1$.

Then

$$a^d = m^{ed} = m^{k\phi(n)+1}.$$

By the Euler-Lagrange theorem, $m^{\phi(n)} = 1$ modulo n. Hence

$$a^d = m$$

modulo n and we have decrypted the message.

Using the Euler-Fermat theorem it's easy! The bank computes

$$a^d$$

where d is the decryption key, which satisfies $ed = k\phi(n) + 1$.

Then

$$a^d = m^{ed} = m^{k\phi(n)+1}.$$

By the Euler-Lagrange theorem, $m^{\phi(n)} = 1$ modulo n. Hence

$$a^d = m$$

modulo n and we have decrypted the message.

Notice that to find d from e (which is public), you must know $\phi(n)$, which is hard even if n is known (but easy if you know p and q). Therefore, the method is considered safe; to break it is as hard as factoring n, which is believed to be hard.

This method of encryption is also called asymmetric, because of the different roles played by the sender and the receiver. The drawback of the method is that it is quite messy to send long messages. Recall that the message m was a number less than n and n has less than a thousand digits. Written with zeroes and ones instead it has perhaps 3000 digits, so you send at most a message of 3kB.

This method of encryption is also called asymmetric, because of the different roles played by the sender and the receiver. The drawback of the method is that it is quite messy to send long messages. Recall that the message m was a number less than n and n has less than a thousand digits. Written with zeroes and ones instead it has perhaps 3000 digits, so you send at most a message of 3kB.

What one does in practice, is to use the method to agree on a key for a different, symmetric, encryption method – that is more efficient.

This method of encryption is also called asymmetric, because of the different roles played by the sender and the receiver. The drawback of the method is that it is quite messy to send long messages. Recall that the message m was a number less than n and n has less than a thousand digits. Written with zeroes and ones instead it has perhaps 3000 digits, so you send at most a message of 3kB.

What one does in practice, is to use the method to agree on a key for a different, symmetric, encryption method – that is more efficient.

The method described was developed roughly between 1970 and 2000. Now there are many more methods that are more efficient. You can send somewhat longer message and the security is higher.

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Such a Riemann surface can be found as the set of solutions of a third degree equation like

$$x^3 + ay^2 = b,$$

where x, y are complex numbers. (You have to add a point at 'infinity').

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Such a Riemann surface can be found as the set of solutions of a third degree equation like

$$x^3 + ay^2 = b,$$

where x, y are complex numbers. (You have to add a point at 'infinity').

Since the equation is of degree 3, one can prove that the set of solutions form a torus.

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Such a Riemann surface can be found as the set of solutions of a third degree equation like

$$x^3 + ay^2 = b,$$

where x, y are complex numbers. (You have to add a point at 'infinity').

Since the equation is of degree 3, one can prove that the set of solutions form a torus. Since a torus is a product of two circles, it has a group structure; you can add points on the torus and get a new point on the torus!

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Such a Riemann surface can be found as the set of solutions of a third degree equation like

$$x^3 + ay^2 = b,$$

where x, y are complex numbers. (You have to add a point at 'infinity').

Since the equation is of degree 3, one can prove that the set of solutions form a torus. Since a torus is a product of two circles, it has a group structure; you can add points on the torus and get a new point on the torus! One can construct other encryption methods using this group instead of the group of integers modulo n.

Elliptic curves

Instead of modular arithmetic, these methods use other algebraic structures. A popular one is based on *elliptic curves*. An elliptic curve is a Riemann surface, that has the form of a torus (a product of two circles).

Such a Riemann surface can be found as the set of solutions of a third degree equation like

$$x^3 + ay^2 = b,$$

where x, y are complex numbers. (You have to add a point at 'infinity').

Since the equation is of degree 3, one can prove that the set of solutions form a torus. Since a torus is a product of two circles, it has a group structure; you can add points on the torus and get a new point on the torus! One can construct other encryption methods using this group instead of the group of integers modulo n. There are many different elliptic curves to choose from!

Epilogue: The NSA

This, and other developments has made the US National Security Agency the biggest employer of mathematics PhD's in the US.

Epilogue: The NSA

This, and other developments has made the US National Security Agency the biggest employer of mathematics PhD's in the US. A few years ago it was disclosed that the NSA had tried to get an edge in codebreaking by making actors use one particular elliptic curve.