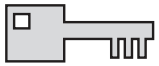
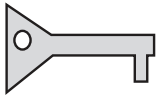
Asymmetrical Encryption Schemes

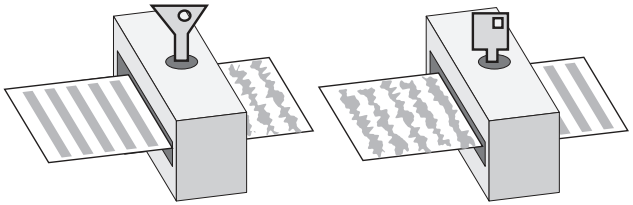
What can we do with a special lock with two specially matched keys?



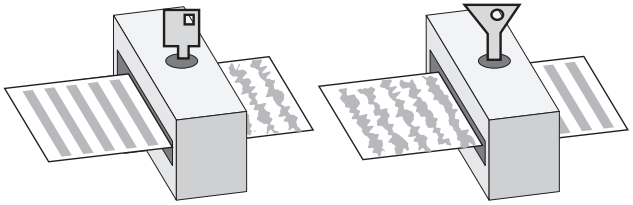
The lock works like this: Either key can lock it, but then only the *other* key will be able unlock it!



That is, there are two specially matched keys for an asymmetrical encryption scheme. Either key can encrypt a message, but then the *other* key is needed to decrypt it.



You can encrypt with first key and decrypt with the second...
... or the other way round



(And though the keys are specially matched, they really are different: there shouldn't be any way to guess what one key is if you know the other.)

In the entire history of encryption, until 1975, no one even conceived that something like this might be possible, much less useful. Yet today, such schemes are the foundation of the internet economy, and are used in all secure electronic transactions!

Even more amazing, the mathematics underlying these schemes had been known for 300 years!

Let's worry about *how* this scheme is possible later: for now, just imagine that *somehow* this can be implemented. **What can we do with such a device?**

Alice Wants To Send Bob A Secret Message

Alice wants to send Bob a secret message across the internet. They've never met, so don't share a key for a symmetrical encryption scheme. But using an asymmetrical scheme, this is a piece of cake.

Bob chooses a pair of keys; one he'll keep for his private key, and he'll send the other, his public key, to Alice. Alice will use his public key to encode her message to Bob.

Bob's public key Bob's private key

Only Bob has his private key, so only Bob can decode the message.

The beauty of this is that Bob doesn't care who in the world sees his public key: it's only good for encoding messages if you don't have the private key— and that's safely in his pocket!

Problem #1: Asymmetrical schemes tend to be much slower than symmetrical ones.

That's easy enough to solve: asymmetrical schemes are very good for short messages— like sending the key for a more efficient symmetrical scheme!

So for longer messages, Alice chooses a key for a symmetrical message, locks it with Bob's public key and sends it to Bob. Bob unlocks it with his private key, and presto! They both have a copy of a key for a symmetrical encryption scheme!

Problem #2: Alice wants to send her credit card number to Bob's Internet Bonanza (.com) She recieves a public key. Hmm: How does she know it's really Bob's public key, and not, say Charlie's? We have to assume Charlie can hijack anything along the route!

If someone handed you an envelope with an unknown address on it, would you put your checking account number in it and mail it off?

The solution was the invention of "Certificates", vouched for by trusted agencies; two companies VeriSign and RSA Security provide most commercial certificates today; PGP (Pretty Good Privacy) provides an open-source, crypto-activist, non-commercial alternative source of certificates.

Here's how it works:

Bob sends his public key to VeriSign, with other information (such as his name and an email address); for a fee, Verisign looks into matters and then certifies the key by "signing" it.

But this raises another problem!

Problem #3: How can VeriSign "sign" something securely? How can *anyone* sign something on the internet? Why should you believe a message like:

"This is my genuine signature! Really it is"

Asymmetrical schemes provide the key!

To provide an authentic signature, VeriSign encrypts this message with its private key. This encrypted message serves as a signature for VeriSign: If Alice, or anyone else, wants to check that VeriSign's signature is valid, she just uses VeriSign's public key to decrypt the signature! (She can only decrypt the signature if the signature is valid!)

Problem #4) But Alice needs to trust VeriSign's public key to decrypt VeriSign's signature, to vouch for Bob's public key! This looks like the same problem all over again! Who vouches for VeriSign?

This is pretty easy: VeriSign's business is founded on trust in its brand. You can't fake a public key from VeriSign, because it is so widely available and easy to check. VeriSign won't scam you because that would destroy its business.

An asymmetrical encryption scheme: RSA

Rivest, Shamir, Adleman, ca. 1975

using Fermat's Little Theorem, ca 1640.

part 1

1) Experiment: Pick any two numbers — oops, make sure they have no common factor (are "relatively prime"; so 8 and 18 won't work: they have a common factor of 2. But 4 and 9 would be fine. 1 doesn't count as a common factor.)

Call one the first one the "base" and the other the "modulus". Take powers of the base mod the modulus. So for example, taking powers of 4 mod 9 we have
4 7(remember, mod 9!) then 1 4 7 1 4 7 1 4 7 1 etc.

Now the next thing is: count the numbers from 1 to the modulus (1 to 9, in this case) that are relatively prime to the modulus itself (From 1 to 9, there are six numbers relatively prime to 9,: 1, 2, 4, 5, 7, 8) This is called, mysteriously, the *totient* of 9, and is denoted ϕ.

Fermat's Little Theorem: For any base b and any modulus m (such that b and m have no common factors) b raised to the ϕth power is 1 mod m!

Check it out! 4 raised to the 6th power is 1 mod 9!

Try a few other examples!

2) The totient of a prime number p is p-1: by definition a prime number has no factors other than 1 and itself, and and so no numbers up from 1 to p-1 have a common factor with p.

Harder: What is the totient of a number is of the form pq where p and q are different prime numbers? Let's try an example:

Of the numbers less than 15, which are relatively prime to 15?

```
 1  2  3  4  5          1  2  3
 6  7  8  9 10          4  5  6
11 12 13 14 15          7  8  9
                       10 11 12
                       13 14 15
```

Can you come with the right formula?
What is the totient of 3•5 What is the totient of 3•5; How about 5•7
What's the formula? Why does it work?

3) Accepting Fermat's Little Theorem, $a^{\phi} = 1 \bmod m$. What is $a^{\phi+1} \bmod m$?
What is $a^{2\phi+1} \bmod m$? $a^{2\phi+1} \bmod m$? $a^{n} \bmod m$ where $n = 1 \bmod \phi$?

An asymmetrical encryption scheme: RSA

Rivest, Shamir, Adleman, ca. 1975

using Fermat's Little Theorem, ca 1640.

part 1

1) Experiment: Pick any two numbers — oops, make sure they have no common factor (are "relatively prime"; so 8 and 18 won't work: they have a common factor of 2. But 4 and 9 would be fine. 1 doesn't count as a common factor.)

Call one the first one the "base" and the other the "modulus". Take powers of the base mod the modulus. So for example, taking powers of 4 mod 9 we have
4 7(remember, mod 9!) then 1 4 7 1 4 7 1 4 7 1 etc.

Now the next thing is: count the numbers from 1 to the modulus (1 to 9, in this case) that are relatively prime to the modulus itself (From 1 to 9, there are six numbers relatively prime to 9,: 1, 2, 4, 5, 7, 8) This is called, mysteriously, the *totient* of 9, and is denoted ϕ.

Fermat's Little Theorem: For any base b and any modulus m (such that b and m have no common factors) b raised to the ϕth power is 1 mod m!

Check it out! 4 raised to the 6th power is 1 mod 9!

Try a few other examples!

2) The totient of a prime number p is p-1: by definition a prime number has no factors other than 1 and itself, and and so no numbers up from 1 to p-1 have a common factor with p.

Harder: What is the totient of a number is of the form pq where p and q are different prime numbers? Let's try an example:

Of the numbers less than 15, which are relatively prime to 15?

```
 1  2  3  4  5          1  2  3
 6  7  8  9 10          4  5  6
11 12 13 14 15          7  8  9
                       10 11 12
                       13 14 15
```

Can you come with the right formula?
What is the totient of 3•5 What is the totient of 3•5; How about 5•7
What's the formula? Why does it work?

3) Accepting Fermat's Little Theorem, $a^{\phi} = 1 \bmod m$. What is $a^{\phi+1} \bmod m$?
What is $a^{2\phi+1} \bmod m$? $a^{2\phi+1} \bmod m$? $a^{n} \bmod m$ where $n = 1 \bmod \phi$?