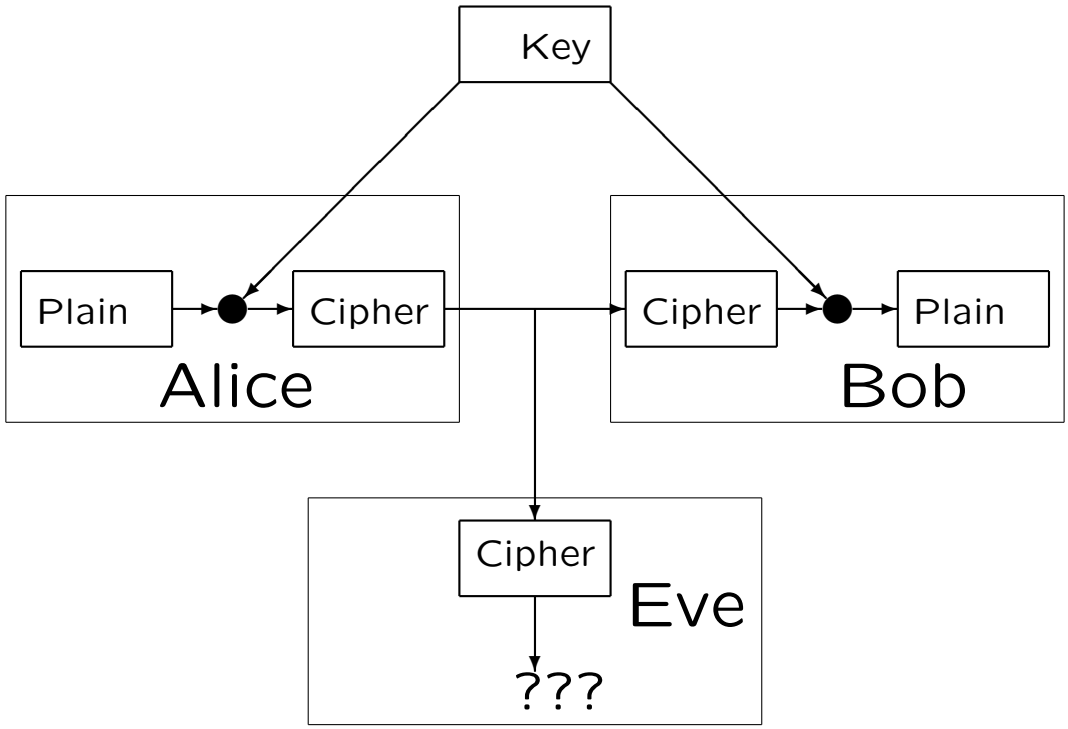
Cryptography and cryptanalysis

Traditional cryptography works like this. Alice wants to communicate with Bob in such a way that an eavesdropper cannot understand the communication.



Alice and Bob must share the key without Eve's knowledge.

A stream cipher

Plaintext	**Hi!**
ASCII code	100100011010010100001
Random key	110111001110110110001
Ciphertext	010011010100100010000
Random key	110111001110110110001
ASCII code	100100011010010100001
Plaintext	**Hi!**

The key is a binary string which must be as long as the message.

Shannon's Theorem: If the key is random then the cipher is secure against any statistical attack.

But how do Alice and Bob share the key securely, if they can't share the message securely?

Vigenère cipher

Vigenère's idea in the 16th century was to use a keyword repeated as often as necessay. The keyword is easy to remember. Letters of the key tell us how many places in the alphabet to shift the plaintext in each position.

e	n	e	m	y	p	a	t	r	o	l	s
F	O	X	E	S	F	O	X	E	S	F	O
J	B	B	Q	Q	U	O	Q	V	G	Q	G

We can make this more secure by various methods:

- use several keywords of different lengths successively;
- use a random Latin square instead of just shifting.

A mechanised version of this was used in the German Fish cipher in World War II, and subsequently in shift register ciphers.

Public-key cryptography

The most significant recent development in cryptography, invented by Diffie and Hellman in the early 1970s, is *public-key cryptography*. Each user of the system chooses a secret key and calculates from it a public key which is available to all users. If Alice wants to send Bob a message, she uses his public key to encrypt.

The security of the system depends on the following assumptions:

- Encrypting is easy.
- Decrypting is easy if you know the secret key, and is hard otherwise.
- Calculating the public key from the secret key is easy, but going in the reverse direction is hard.

Public-key cryptography

Here ‘hard’ means that the computation is not difficult in principle, it just takes so long that by the time you get the answer, it is no longer of any value.

Of course, as computers get faster, the grey area between ‘easy’ and ‘hard’ shifts, and it is necessary to keep updating the size of the key.

In practice, public-key ciphers are not as convenient or fast as conventional ciphers such as the Advanced Encryption System, so they are mostly used for sending a key which is then used in a conventional cipher.

The RSA cipher

This depends on the fact that testing whether a large integer is prime is 'easy', but factorising a large integer into its prime factors is 'hard'.

Bob chooses two large prime numbers p and q, and an integer e such that e and $(p-1)(q-1)$ have no prime factor. This guarantees that the function

$$T_e : x \mapsto x^e \bmod pq$$

is one-to-one and has an inverse T_d.

Bob's public key is the pair (N, e), where $N = pq$. To send a message to Bob, translate it in a number in the range $[0, \ldots, N-1]$ and encrypt it with the function T_e. Then Bob can decrypt it with the function T_d (which he knows).

It can be shown that finding the inverse function T_d knowing only N and e is as hard as factorising N.

Quantum theory

Quantum theory will have two significant impacts on cryptography. First, if a *quantum computer* can be built, then one thing it will be able to do is to factorise large integers very quickly, and so the RSA cipher will no longer be secure.

On the other hand, quantum theory raises the possibility of a completely secure cipher. This depends on the principle that any observer influences the system they are observing in unpredictable ways, and so the presence of an eavesdropper can be detected.

Quantum cryptography

It works roughly like this.

Alice sends a large number of random bits to Bob over a quantum channel (encoded as polarised photons). Then Alice and Bob sacrifice a fixed number of these (say 100); communicating over an insecure conventional line, Alice tells Bob the bits she sent and Bob compares them with the ones he received. If they agree, then all is well.

If there is an eavesdropper, the chance of 100 bits all being received correctly is the same as a fair coin coming down heads 100 times, which is negligible.

If there was an eavesdropper, Alice and Bob try again later. If not, then they use the remaining random bits as the shared key in a one-time pad. The message can be sent over an insecure channel.