Cryptography

Terminology[1]

Suppose that someone wants to send a message to a receiver, and wants to be sure that no-one else can read the message. However, there is the possibility that someone else opens the letter or hears the electronic communication.

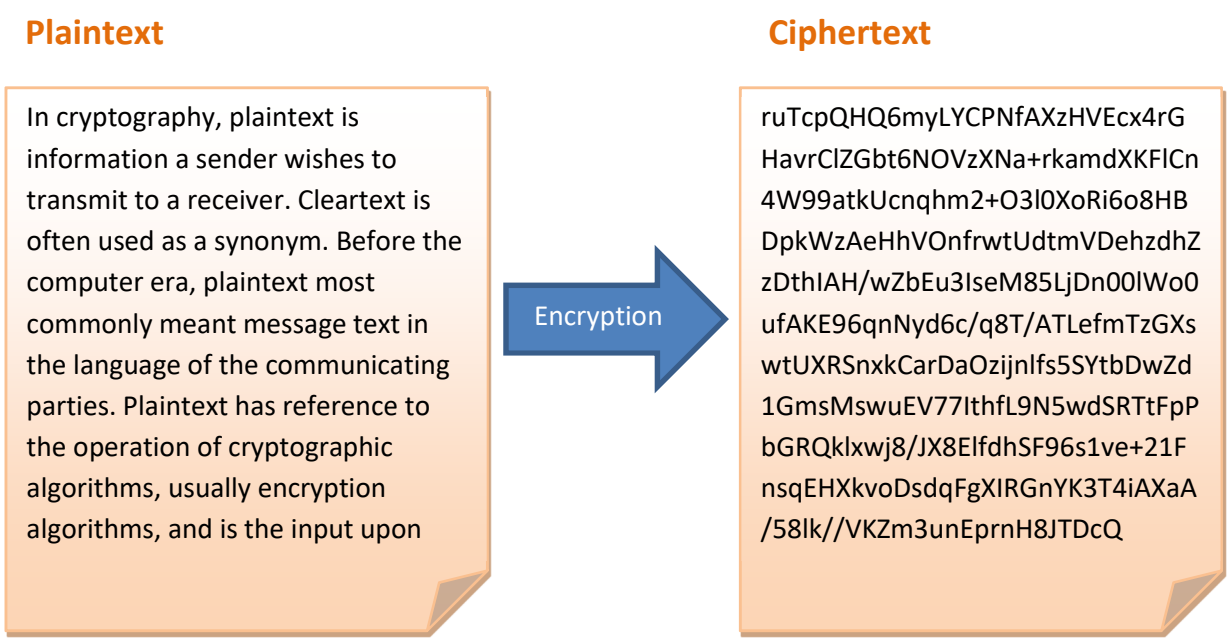
In cryptographic terminology, the message is called **plaintext** or **cleartext**. Encoding the contents of the message in such a way that hides its contents from outsiders is called **encryption**. The encrypted message is called the **ciphertext**. The process of retrieving the plaintext from the ciphertext is called **decryption**. Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key.

Cryptography is the art or science of keeping messages secret. **Cryptanalysis** is the art of **breaking** ciphers, i.e. retrieving the plaintext without knowing the proper key. People who do cryptography are **cryptographers**, and practitioners of cryptanalysis are **cryptanalysts**.

Use of the Cryptography

1. Encryption

In a simplest form, encryption means the conversion of data in some unreadable form. This helps in protecting the privacy while sending the data from sender to receiver.



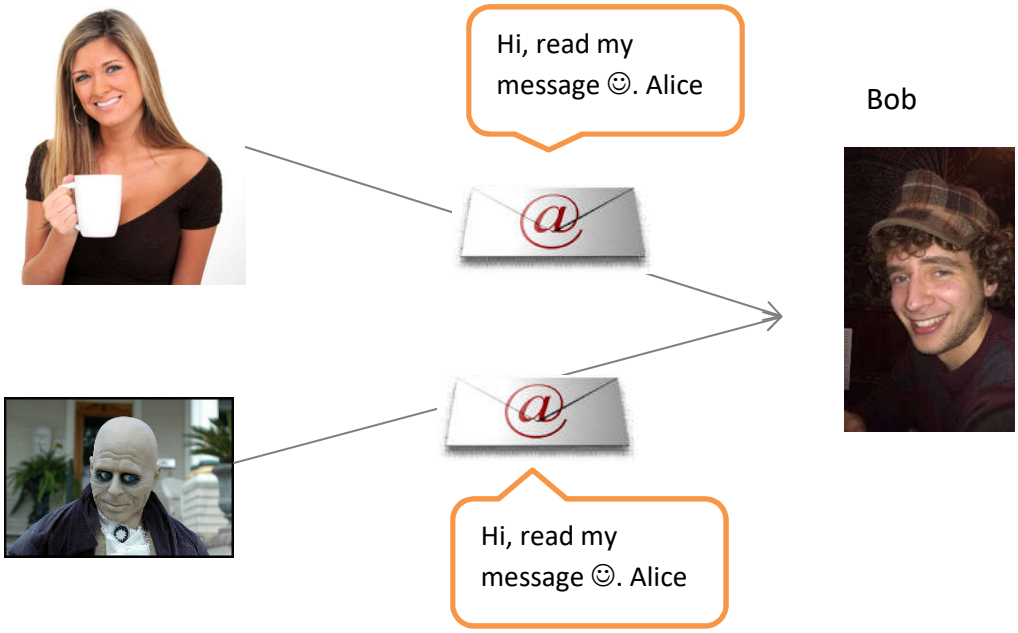
On the receiver side, the data can be decrypted and can be brought back to its original form. The reverse of encryption is called **decryption**. The concept of encryption and decryption requires some extra information for encrypting and decrypting the data. This information is known as **key**. There

[1] http://minelinks.com/supercode/index.html

may be cases when same key can be used for both encryption and decryption while in certain cases, encryption and decryption may require different keys.

2. Authentication

This is another important principle of cryptography. Authentication ensures that the message was originated from the originator claimed in the message. Now, one may think how to make it possible? Suppose, Alice sends a message to Bob and now Bob wants proof that the message has been indeed sent by Alice.



This can be made possible if Alice performs some action on message that Bob knows only Alice can do. Well, this forms the basic fundamental of **authentication**.

3. Integrity

Now, one problem that a communication system can face is the loss of integrity of messages being sent from sender to receiver. This means that cryptography should ensure that the messages that are received by the receiver are not altered anywhere on the communication path. This can be achieved by using the concept of **cryptographic hash**.

4. Non Repudiation

What happens if Alice sends a message to Bob but denies that she has actually sent the message? Cases like these may happen and cryptography should prevent the originator or sender to act this way. One popular way to achieve this is through the use of **digital signatures**. These may serve also as the means of authentication, but if the use of digital signatures is enforced, then the non-repudiation is accomplished as well.

Types of Cryptography

There are three types of cryptography techniques:

- **Secret key** Cryptography
- **Public key** cryptography
- **Hash** Functions

Secret Key Cryptography

This type of cryptography technique uses just a **single key**. The sender applies a key to encrypt a message while the receiver applies the same key to decrypt the message. Since only single key is used so we say that this is a **symmetric encryption**.

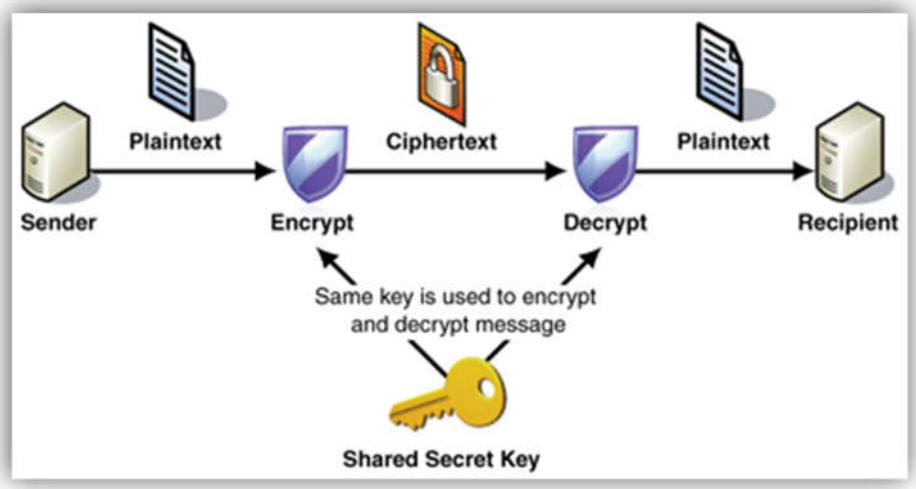


Figure 1: Symmetric Encryption (source: http://msdn.microsoft.com)

Public Key Cryptography

In this method, each party has a private key and a public key. The private is secret and is not revealed while the public key is shared with all those whom you want to communicate with. If Alice wants to send a message to bob, then Alice will encrypt it with Bob's public key and Bob can decrypt the message with its private key.

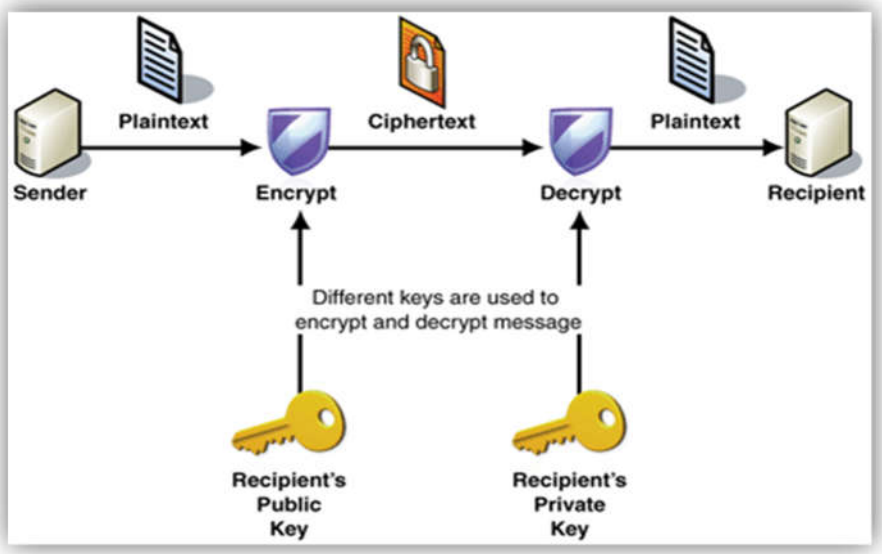


Figure 2: Asymmetric Encryption (source: http://msdn.microsoft.com)

Since a pair of keys is applied here so this technique is also known as asymmetric encryption.

Hash Functions

This technique does not involve any key. Rather it uses a fixed length hash value that is computed on the basis of the plain text message. Hash functions are used to check the integrity of the message to ensure that the message has not be altered, compromised or affected by virus. **This is ONE WAY transformation (!).**

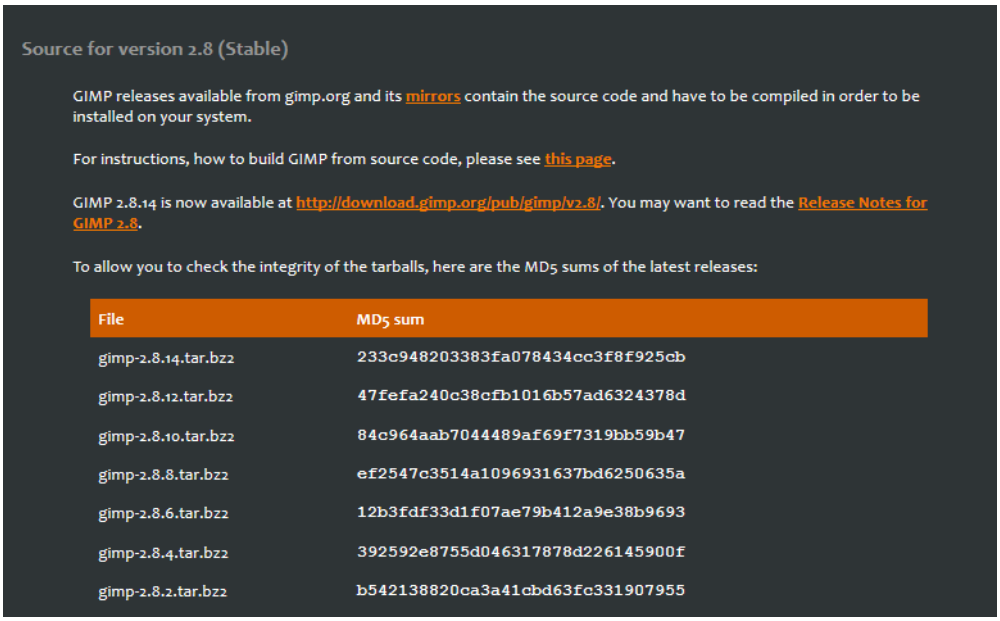
Example:

In this method, each party has a private key and a public key. The private is secret and is not revealed while the public key is shared with all those whom you want to communicate with. If Alice wants to send a message to bob, then Alice will encrypt it with Bob's public key and Bob can decrypt the message with its private key.	SHA1 Hash Algorithm →	**SHA1 Hash** ac1ad2fcbbb8e59080450d6b078f1be43f507b34

If the message is altered a bit only, the hash changes noticeably.

On this method, each party has a private key and a public key. The private is secret and is not revealed while the public key is shared with all those whom you want to communicate with. If Alice wants to send a message to bob, then Alice will encrypt it with Bob's public key and Bob can decrypt the message with its private key.	SHA1 Hash Algorithm →	**SHA1 Hash** 34d33b4f722e3a7a29205f8d9edf8be419d31280

Another example is the hash available for downloaded files (e.g. setup files, ISO images of drives etc.), where the hash can ensure the user files were downloaded correctly.



Source for version 2.8 (Stable)

GIMP releases available from gimp.org and its mirrors contain the source code and have to be compiled in order to be installed on your system.

For instructions, how to build GIMP from source code, please see this page.

GIMP 2.8.14 is now available at http://download.gimp.org/pub/gimp/v2.8/. You may want to read the Release Notes for GIMP 2.8.

To allow you to check the integrity of the tarballs, here are the MD5 sums of the latest releases:

File	MD5 sum
gimp-2.8.14.tar.bz2	233c948203383fa078434cc3f8f925cb
gimp-2.8.12.tar.bz2	47fefa240c38cfb1016b57ad6324378d
gimp-2.8.10.tar.bz2	84c964aab7044489af69f7319bb59b47
gimp-2.8.8.tar.bz2	ef2547c3514a1096931637bd6250635a
gimp-2.8.6.tar.bz2	12b3fdf33d1f07ae79b412a9e38b9693
gimp-2.8.4.tar.bz2	392592e8755d046317878d226145900f
gimp-2.8.2.tar.bz2	b542138820ca3a41cbd63fc331907955

Figure 3: GIMP download page - there are MD5 hash results for files containing the source codes (source: http://www.gimp.org/downloads/)

Common Cryptographic Algorithms

- **Private key:** DES, AES (this one is used to encrypt WPA2 communication over Wi-Fi), Twofish
- **Public key:** RSA, Diffie-Helman
- **Hash:** MD5 (not secure anymore), SHA-1, SHA-256, Whirpool

Examples of Simple Symmetric Algorithms

Caesar Cipher

It is a primitive encryption/decryption done by shift of symbols in the alphabet. For instance, if shift is 3, then

SECRETMESSAGE

is encrypted as

VHFUHWPHVVDJH

Prone to brute force attack or frequency analysis.

Figure 4: Ceasar Cipher - logo of the Android application (source: https://play.google.com/store/apps/details?id=com.coconuts.caesarcipher)

Vigenère Cipher

The encryption is based on a key – a word or a sentence – and a table of shifted alphabets:

Process:

Write the key (e.g. BALL) under the letters of the plain text:

```
HI THERE, THIS IS ALICE

BA LLBAL  LBAL LB ALLBA
```

Then, find in the table intersections for the pairs of plaintext symbol (*row*) and the related key symbol (*column*):

H, B → I

I, A → I

T, L → E

H, L → S

etc.

The cipher text is `II ESFRP, EIID TT AWTDE.` The decryption is trivial.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vernam Cipher

Also known as **one-time pad**. It cannot be cracked, if it is used properly.

Plaintext is paired with a random secret key. The key must meet these requirements:

- truly random
- never re-used (not even a part)
- as long as the message

The cipher text is calculation is based on modulo arithmetic.

Example

Plain text: `HI THERE, THIS IS ALICE`

Key: `WJ SKKVE  DVYK DW PWCVR`

`Alphabet:`

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Calculation of the ciphertext:

H (7th from A → 7) + W(22) = 29 ➔ 29 mod 26 (number of letters in the English alphabet) = 3 → D

I (8) + J (9) = 17 ➔ 17 mod 26 = 17 → R

T (19) + S (18) = 37 ➔ 37 mod 26 = 11 → L

H (7) + K (10) = 17 ➔ 17 mod 26 = 17 → R

Decryption is similar – key is subtracted from the ciphertext, and then it is modified by modulo.

D (3) – W(22) = -19 ➔ -19 mod 26 = 7 → H

R (17) – J (9) = 8 ➔ 8 mod 26 = 8 → I

Digital Signatures

Some public-key algorithms can be used to generate **digital signatures**. A digital signature is a small amount of data that was created using some secret key, and there is a public key that can be used to verify that the signature was really generated using the corresponding private key. The algorithm used to generate the signature must be such that without knowing the secret key it is not possible to create a signature that would verify as valid.

Figure 5: Example of the e-signature in a PDF file

Digital signatures are used to verify that a message really comes from the claimed sender (assuming only the sender knows the secret key corresponding to his/her public key). They can also be used to **timestamp** documents: a trusted party **signs** the document and its timestamp with his/her secret key, thus testifying that the document existed at the stated time.

Digital signatures can also be used to testify (or **certify**) that a public key belongs to a particular person. This is done by signing the combination of the key and the information about its owner by a trusted key. The digital signature by a third party (owner of the trusted key), the public key and information about the owner of the public key are often called **certificates**.

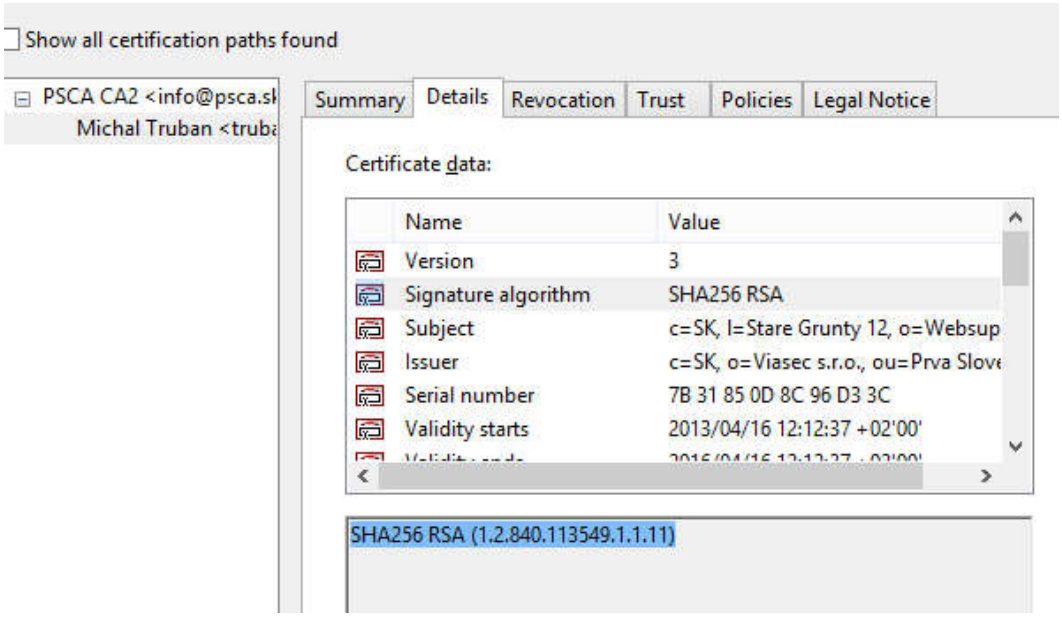


Figure 6: Example of the certificate for an e-signature

The reason for trusting that third party key may again be signed by another trusted key. Eventually some key must be a **root** of the trust hierarchy (that is, it is not trusted because it was signed by

somebody, but because you believe a priori that the key can be trusted). In a **centralized key infrastructure** there are very few roots in the trust network (e.g., trusted government agencies; such roots are also called **certification authorities**).

Cryptocurrencies and Block-chains

Cryptocurrency is an encrypted decentralized digital currency transferred between peers and confirmed in a public ledger (*účtovná kniha*) via a process known as **mining** (*this is the essential purpose of the mining, not the gain of new coins – that is just motivation for miners*).

The ledger in case of the cryptocurrencies is a **blockchain**. It stores all confirmed transactions between accounts attached to digital wallets.

Any transfer between the accounts is called a **transaction**. When a transaction is made, wallets use an encrypted electronic signature (an encrypted piece of data called a cryptographic signature) to provide a mathematical proof that the transaction is coming from the owner of the wallet. The confirmation process takes a bit of time (ten minutes for bitcoin) while "miners" mine (ie. confirm transactions and add them to the public ledger).

Mining is the process of confirming transactions and adding them to a public ledger. In order to add a transaction to the ledger, the "miner" must solve an increasingly-complex computational problem (sort of like a mathematical puzzle). Mining is open source, so anyone can confirm the transaction. The first "miner" to solve the puzzle adds a "block" of transactions to the ledger. The way in which transactions, blocks, and the public blockchain ledger work together ensures that no one individual can easily add or change a block at will. Once a block is added to the ledger, all correlating transactions are permanent and a small transaction fee is added to the miner's wallet (along with newly created coins). The mining process is what gives value to the coins and is known as a **proof-of-work system (POW)**.

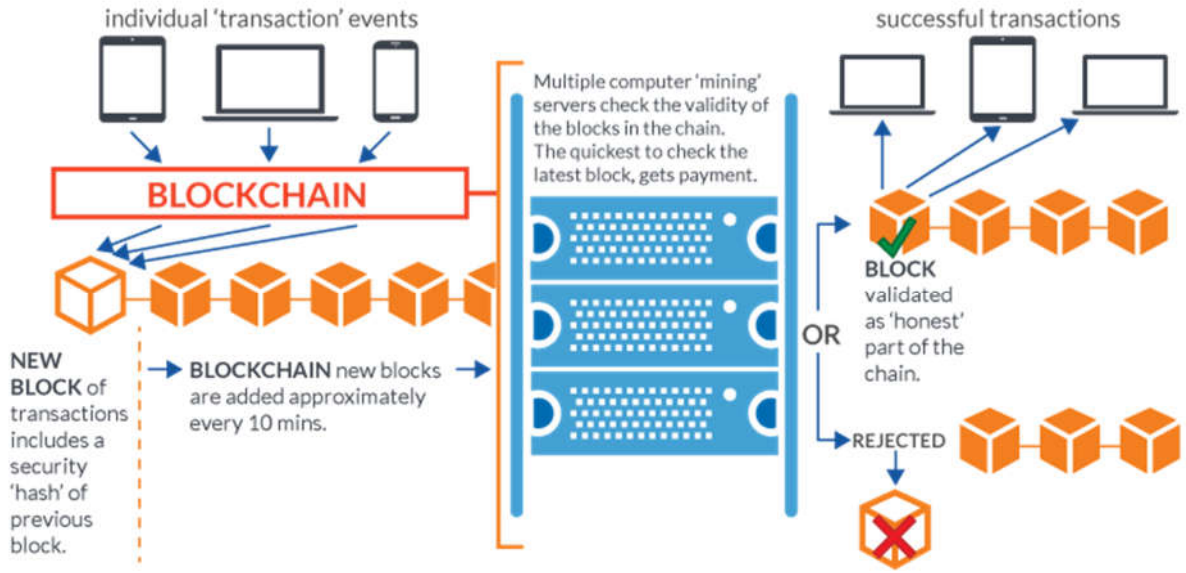


Figure 7: How blockchain works (https://blockchain.open.ac.uk/blckchn-data/uploads/2016/04/blockchain-overview-1-1.png)

This system is, however, very demanding of energy. Currently (as of 2017) a single Bitcoin transaction requires 163 kWh, which makes it very expensive; in the future it can be much worse.

There is another possibility – **proof-of-stake**. In this case the creator of the next block (which contains new transactions) is chosen in pseudo-random order with the chances weighed by his/her wealth (so called **stake**). The higher stake the greater chance of making a block and thus getting the mining fee.