Cryptography

Cryptography

- Cryptography is art and science of obscuring (and protecting!) information.

Cryptography

- Cryptography is art and science of obscuring (and protecting!) information.

- We ordinarily do this to provide a basic level of security against an adversary who might do bad things with the information, had they been able to see it "in the clear."

Ciphers

- Ciphers are *algorithms* used to obscure (encipher) or reveal (decipher) information.

Ciphers

- Ciphers are *algorithms* used to obscure (encipher) or reveal (decipher) information.

- A wide variety of different types of ciphers exist, with varying levels of inherent security potential.

Substitution Cipher

- Imagine having possession of this device.

Image source: eBay

Substitution Cipher

- Imagine having possession of this device.

- 3 = L
- 4 = M
- 5 = K
- 6 = W
- 7 = N
- 8 = O

Image source: eBay

Substitution Cipher

- What's the problem with this cipher? Put another way, what is the "attack vector"?

Image source: eBay

Substitution Cipher

- What's the problem with this cipher? Put another way, what is the "attack vector"?

- If the adversary is also a member of Little Orphan Annie's Secret Society, they know how to crack the code.

Image source: eBay

Substitution Cipher

- What's the problem with this cipher? Put another way, what is the "attack vector"?

- If the adversary is also a member of Little Orphan Annie's Secret Society, they know how to crack the code.

- We might think of this decoder pin as a "key".

Image source: eBay

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	…	24	25	26

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	…	24	25	26

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
3	4	5	6	7	8	9	10	11	12	13	…	26	27	28

+2

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
3	4	5	6	7	8	9	10	11	12	13	…	26	1	2

+2

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
21	22	23	24	25	26	1	2	3	4	5	…	18	19	20

+20

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	…	24	25	26

+26

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Substitution Cipher

- Alternatively, we could use the ordinal positions of letters in a cipher, perhaps.

A	B	C	D	E	F	G	H	I	J	K	…	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	…	24	25	26

+0

- Now, if we simply rotate the starting point (or even if we don't!), we have the basis of a cipher.

Caesar Cipher

- This *rotational cipher* is a rather famous one called Caesar cipher, attributed to Julius Caesar.

Caesar Cipher

- This *rotational cipher* is a rather famous one called Caesar cipher, attributed to Julius Caesar.

- In ancient times, it was apparently very challenging to crack. Nowadays, it's quite easy.

Caesar Cipher

- This *rotational cipher* is a rather famous one called Caesar cipher, attributed to Julius Caesar.
- In ancient times, it was apparently very challenging to crack. Nowadays, it's quite easy.
- Limited number of rotational "keys". Only 26 ways to lay the alphabet out.

Vigenere Cipher

- The Vigenere cipher is an extended idea to the Caesar cipher, but instead of using a single key, it uses multiple keys, by selecting a keyword.

Vigenere Cipher

- The Vigenere cipher is an extended idea to the Caesar cipher, but instead of using a single key, it uses multiple keys, by selecting a keyword.

- Each new letter of the message we want to encrypt (aka the *plaintext*) is enciphered using a different letter of the keyword.

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext					
ordinal position					
keyword					
keyword ordinal position					
sum					
sum, wrapping around					
ciphertext					

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**				
ordinal position	8				
keyword					
keyword ordinal position					
sum					
sum, wrapping around					
ciphertext					

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**				
ordinal position	8				
keyword	**L**				
keyword ordinal position	12				
sum					
sum, wrapping around					
ciphertext					

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**				
ordinal position	8				
keyword	**L**				
keyword ordinal position	12				
sum	**20**				
sum, wrapping around	20				
ciphertext	**T**				

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**	**E**			
ordinal position	8	5			
keyword	**L**	**A**			
keyword ordinal position	12	1			
sum	**20**	**6**			
sum, wrapping around	20	6			
ciphertext	**T**	**F**			

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**	**E**	**L**		
ordinal position	8	5	12		
keyword	**L**	**A**	**W**		
keyword ordinal position	12	1	23		
sum	**20**	**6**	**35**		
sum, wrapping around	20	6	9		
ciphertext	**T**	**F**	**I**		

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**	**E**	**L**	**L**	
ordinal position	8	5	12	12	
keyword	**L**	**A**	**W**	**L**	
keyword ordinal position	12	1	23	12	
sum	**20**	**6**	**35**	**24**	
sum, wrapping around	20	6	9	24	
ciphertext	**T**	**F**	**I**	**X**	

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.

plaintext	**H**	**E**	**L**	**L**	**O**
ordinal position	8	5	12	12	15
keyword	**L**	**A**	**W**	**L**	**A**
keyword ordinal position	12	1	23	12	1
sum	**20**	**6**	**35**	**24**	**16**
sum, wrapping around	20	6	9	24	16
ciphertext	**T**	**F**	**I**	**X**	**P**

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.
- So, **HELLO** becomes **TFIXP**.

Vigenere Cipher

- Let's encrypt the message **HELLO** using the keyword **LAW**.
- So, **HELLO** becomes **TFIXP**.
- Unlike Caesar, which is limited to 26 keys, Vigenere cipher has 26^n keys, where n is the length of the keyword chosen.

Substitution Cipher

- Let's assume that your adversary *isn't* a member of Little Orphan Annie's Secret Society.

Image source: eBay

Substitution Cipher

- Let's assume that your adversary *isn't* a member of Little Orphan Annie's Secret Society.

- How might they nevertheless crack a code enciphered with the pin?

Image source: eBay

Frequency Analysis

A	B	C	D	E	F	G	H	I	J	K	L	M
8.1%	1.5%	2.8%	4.3%	12.7%	2.2%	2.0%	6.1%	7.0%	0.2%	0.8%	4.0%	2.4%

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6.7%	7.5%	1.9%	0.1%	6.0%	6.3%	9.1%	2.8%	1.0%	2.4%	0.2%	2.0%	0.1%

Frequency Analysis

A	B	C	D	E	F	G	H	I	J	K	L	M
8.1%	1.5%	2.8%	4.3%	12.7%	2.2%	2.0%	6.1%	7.0%	0.2%	0.8%	4.0%	2.4%

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6.7%	7.5%	1.9%	0.1%	6.0%	6.3%	9.1%	2.8%	1.0%	2.4%	0.2%	2.0%	0.1%

- It's probably tedious for a human to analyze, but a computer can do it very quickly!

Ciphers

- There are other ciphers that substitute pairs or triples of characters at a time.

Ciphers

- There are other ciphers that substitute pairs or triples of characters at a time.

- There are also *transposition* ciphers, which algorithmically rearrange the letters in a message.

Ciphers

- There are other ciphers that substitute pairs or triples of characters at a time.
- There are also *transposition* ciphers, which algorithmically rearrange the letters in a message.
- The problem is that all of these classic ciphers are easily cracked, and often suffer from a problem of how to distribute the key.

RECOVER PASSWORD

Enter your email and we'll email you a link to change your password.

Enter Email

SEND PASSWORD LINK SIGN IN REGISTER

Image source: fender.com

Hashes

- A major distinction between ciphers and hashes are that ciphers are (generally) reversible, while hashes are (generally) not.

Hashes

- A major distinction between ciphers and hashes are that ciphers are (generally) reversible, while hashes are (generally) not.

- To *hash* some data, we run it through a *hash function*, which mathematically manipulates it in some way, and it outputs a value (sometimes a number, sometimes a string).

Hashes

- Passwords on sites that you likely use every day on the internet are hashed when stored in the site's database.

Hashes

- Passwords on sites that you likely use every day on the internet are hashed when stored in the site's database.

- This is why those services can't just tell you what your password is – they don't know it either (hopefully!)

Hash Function

- A good hash function should:

Hash Function

- A good hash function should:
  - Use only the data being hashed

Hash Function

- A good hash function should:
  - Use only the data being hashed
  - Use all of the data being hashed

Hash Function

- A good hash function should:
  - Use only the data being hashed
  - Use all of the data being hashed
  - Be deterministic

Hash Function

- A good hash function should:
  - Use only the data being hashed
  - Use all of the data being hashed
  - Be deterministic
  - Uniformly distribute data

Hash Function

- A good hash function should:
  - Use only the data being hashed
  - Use all of the data being hashed
  - Be deterministic
  - Uniformly distribute data
  - Generate very different hash codes for very similar data

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58
- Note that this isn't reversible. There are lots of other words that would hash to 58 using this (bad) function:

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58
- Note that this isn't reversible. There are lots of other words that would hash to 58 using this (bad) function:
  - **ARTS, RATS**

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58
- Note that this isn't reversible. There are lots of other words that would hash to 58 using this (bad) function:
  - **ARTS, RATS**
  - **SWAP, PAWS, WASP**

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58
- Note that this isn't reversible. There are lots of other words that would hash to 58 using this (bad) function:
  - **ARTS, RATS**
  - **SWAP, PAWS, WASP**
  - **MULL**

Hash Function

- Hash function (bad one!):
  - Add up the ordinal positions of all the letters in the hashed string.

- **STAR** → 58
- Note that this isn't reversible. There are lots of other words that would hash to 58 using this (bad) function:
  - **ARTS, RATS**
  - **SWAP, PAWS, WASP**
  - **MULL**
  - **BBBBBBBBBBBBBBBBBBBBBBBBBBBBB**

Hash Function

- So then, how do you check if the user gave the right password if all we store is the hash (assuming the hash function is good and more complicated than this example)

Hash Function

- So then, how do you check if the user gave the right password if all we store is the hash (assuming the hash function is good and more complicated than this example)

- By running the actual password through the hash function, if we get a match, odds are they entered the right password.

Modern Cryptography

- A significant fraction of the basis of modern cryptography relies on variations on this theme of hashing.

Modern Cryptography

- A significant fraction of the basis of modern cryptography relies on variations on this theme of hashing.

- Typically the algorithms that are used do not work on a character-by-character basis.

Modern Cryptography

- A significant fraction of the basis of modern cryptography relies on variations on this theme of hashing.
- Typically the algorithms that are used do not work on a character-by-character basis.
- Most work by taking data of some arbitrary size, and mapping it to a string of bits that is always exactly the same size.

Cryptographic Hash Functions

- A *cryptographic* hash function should:
  - Be extremely difficult (to the point of infeasibility) to reverse

Cryptographic Hash Functions

- A *cryptographic* hash function should:
  - Be extremely difficult (to the point of infeasibility) to reverse
  - Be deterministic

Cryptographic Hash Functions

- A *cryptographic* hash function should:
  - Be extremely difficult (to the point of infeasibility) to reverse
  - Be deterministic
  - Generate very different hash codes for very similar data

Cryptographic Hash Functions

- A *cryptographic* hash function should:
  - Be extremely difficult (to the point of infeasibility) to reverse
  - Be deterministic
  - Generate very different hash codes for very similar data
  - Never allow two different sets of data to hash to the same value

Cryptographic Hash Functions

- A *cryptographic* hash function should:
  - Be extremely difficult (to the point of infeasibility) to reverse
  - Be deterministic
  - Generate very different hash codes for very similar data
  - Never allow two different sets of data to hash to the same value

- The output of a cryptographic hash function is normally referred to as the digest.

SHA-1

- SHA-1 is a famous cryptographic hash function first developed by the NSA in the mid-1990s.

SHA-1

- SHA-1 is a famous cryptographic hash function first developed by the NSA in the mid-1990s.

- It works by mapping messages of arbitrary size into a "bit string" of 160 bits. This means that there are 2^{160} different SHA-1 digests, or a bit over 10^{48}.

SHA-1

- SHA-1 is a famous cryptographic hash function first developed by the NSA in the mid-1990s.
- It works by mapping messages of arbitrary size into a "bit string" of 160 bits. This means that there are 2^{160} different SHA-1 digests, or a bit over 10^{48}.
- It is such an important algorithm that federal regulations require its use.

SHAttered

https://shattered.io/

SHAttered

"It is now practically possible to craft two colliding PDF files and obtain a SHA-1 digital signature on the first PDF file which can also be abused as a valid signature on the second PDF file. For example, by crafting the two colliding PDF files as two rental agreements with different rent, it is possible to trick someone to create a valid signature for a high-rent contract by having him or her sign a low-rent contract."

SHAttered

"It is now practically possible to craft two colliding PDF files and obtain a SHA-1 digital signature on the first PDF file which can also be abused as a valid signature on the second PDF file. For example, by crafting the two colliding PDF files as two rental agreements with different rent, it is possible to trick someone to create a valid signature for a high-rent contract by having him or her sign a low-rent contract."

Modern Cryptography

- Fortunately, many other (often more secure!) cryptographic standards are in use by other organizations.

Modern Cryptography

- Fortunately, many other (often more secure!) cryptographic standards are in use by other organizations.

- SHA-2, SHA-3
- *MD5*, MD6

Cryptography

- What sorts of things do we do every day on the internet that rely on cryptography?

Cryptography

- What sorts of things do we do every day on the internet that rely on cryptography?

- Email

Cryptography

- What sorts of things do we do every day on the internet that rely on cryptography?

- Email
- Secure web browsing

Cryptography

- What sorts of things do we do every day on the internet that rely on cryptography?

- Email
- Secure web browsing
- VPN

Cryptography

- What sorts of things do we do every day on the internet that rely on cryptography?

- Email
- Secure web browsing
- VPN
- Document storage
- …

Public-Key Cryptography

- Let's take a trip down memory lane…

Public-Key Cryptography

- Let's take a trip down memory lane…

14 x 8 = 112

Public-Key Cryptography

- Let's take a trip down memory lane…

$$112 \times \tfrac{1}{8} = 14$$

Public-Key Cryptography

- Let's take a trip down memory lane…

112 / 8 = 14

Public-Key Cryptography

- Let's take a trip down memory lane…

- In this sense, we can think of multiplication as a reversible function. If we multiply some number *x* by some other number *y*, we get a result, *z*.

Public-Key Cryptography

- Let's take a trip down memory lane…

- In this sense, we can think of multiplication as a reversible function. If we multiply some number *x* by some other number *y*, we get a result, *z*.

- If we multiply that result, *z* by the reciprocal of *y*, we get back the original *x*. As long as you know what *y* is, it's reversible.

Public-Key Cryptography

- Let's take a trip down memory lane…

$$14 \times 8 = 112$$

$$f(n) = n \times 8$$

Public-Key Cryptography

- Let's take a trip down memory lane…

$$f(14) = 112$$

Public-Key Cryptography

- Let's take a trip down memory lane…

$$f(14) = 112$$

- Here, imagine *14* is the plaintext and *112* is the ciphertext.

Public-Key Cryptography

- Let's take a trip down memory lane…

$$f(14) = 112$$
$$f(n) = n \times 8$$

Public-Key Cryptography

- Let's take a trip down memory lane…

$$f(14) = 112$$

$$f(n) = (n \times 10) - 28$$

Public-Key Cryptography

- Let's take a trip down memory lane…

$$f(14) = 112$$

$$f(n) = (4/7) \times n^2$$

Public-Key Cryptography

- This is the basic idea behind how public-key cryptography works.

Public-Key Cryptography

- This is the basic idea behind how public-key cryptography works.

- Two functions, f(n) and g(n), each of which is a one-way function.

Public-Key Cryptography

- This is the basic idea behind how public-key cryptography works.
- Two functions, f(n) and g(n), each of which is a one-way function.
- One of those functions is *public* and anyone can use it to encrypt information intended for you. The other is *private*, known only to you, and can be used to reverse the encryption of the first.

Asymmetric Encryption

- To generate these keys, start with a really huge, normally prime, randomly-generated number.

Asymmetric Encryption

- To generate these keys, start with a really huge, normally prime, randomly-generated number.

- From there, two complementary one-way functions (quite a bit more complicated than our f(n)) are generated to create a public-private key pair.

Asymmetric Encryption

- To generate these keys, start with a really huge, normally prime, randomly-generated number.
- From there, two complementary one-way functions (quite a bit more complicated than our f(n)) are generated to create a public-private key pair.
- Typically done by a program called RSA.

Asymmetric Encryption

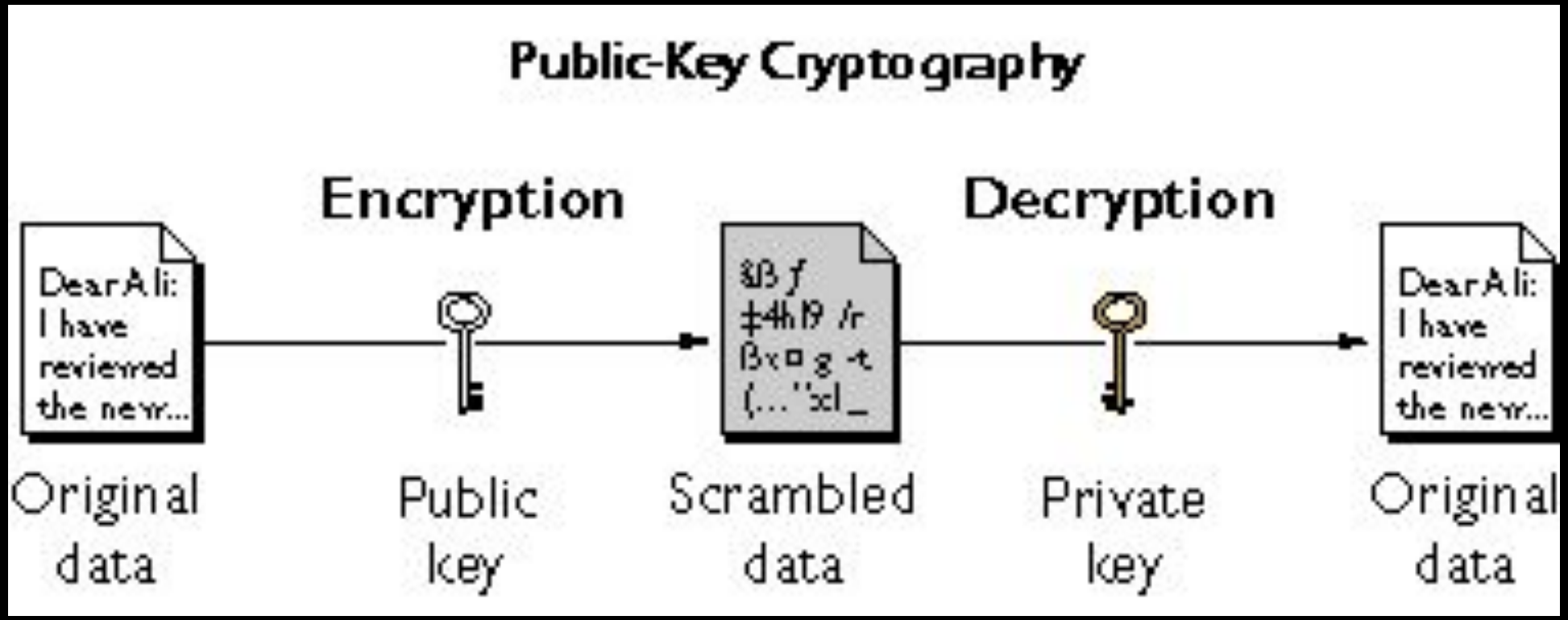


Image source: ibm.com

Asymmetric Encryption

- The encryption step can be done by anyone who has access to the public key.

Asymmetric Encryption

- The encryption step can be done by anyone who has access to the public key.

- The decryption step can (theoretically) be done only by the individual(s) who have the private key.

Digital Signatures

- Digital signatures (not the same as e-signatures!) are almost the inverse of encryption.

Digital Signatures

- Digital signatures (not the same as e-signatures!) are almost the inverse of encryption.

- Using a digital signature, one can verify the authenticity of the sender of a document.

Digital Signatures

- Digital signatures (not the same as e-signatures!) are almost the inverse of encryption.
- Using a digital signature, one can verify the authenticity of the sender of a document.
- Many digital signatures are 256-bits (meaning 2^{256} distinct digital signatures are possible, meaning the likelihood of a "forgery" is infinitesimal).

Digital Signatures

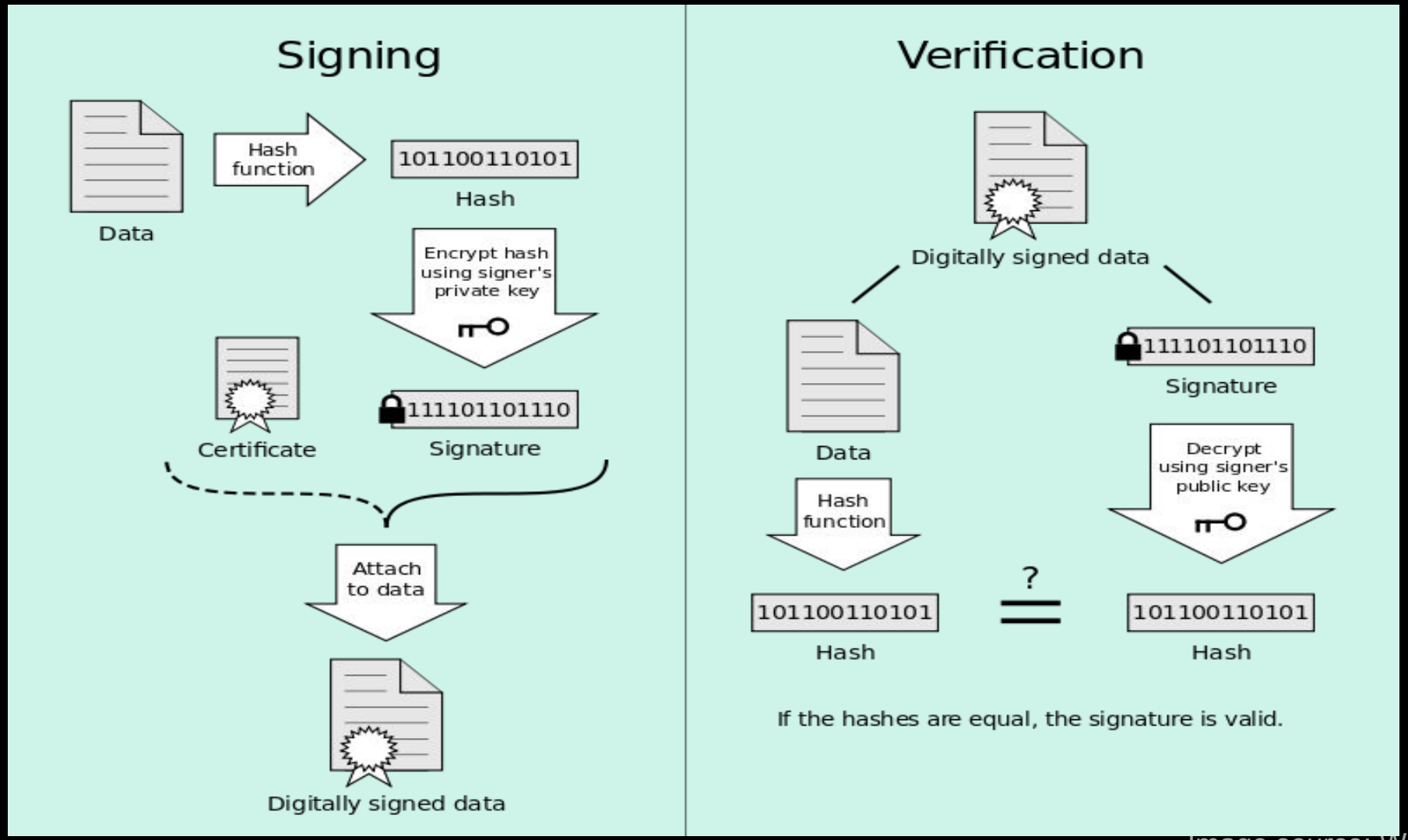


Image source: WikiMedia

Blockchain

- Digital signatures and their ease of verification provide the basis for the very interesting topic of the *blockchain.*

Blockchain

- Digital signatures and their ease of verification provide the basis for the very interesting topic of the *blockchain.*

- The use of blockchain known by most people is in the cryptocurrency domain – Bitcoin and the like, but it has utility far beyond that.

Blockchain

- Digital signatures and their ease of verification provide the basis for the very interesting topic of the *blockchain.*

- The use of blockchain known by most people is in the cryptocurrency domain – Bitcoin and the like, but it has utility far beyond that.

- 3Blue1Brown: https://www.youtube.com/watch?v=bBC-nXj3Ng4

Blockchain

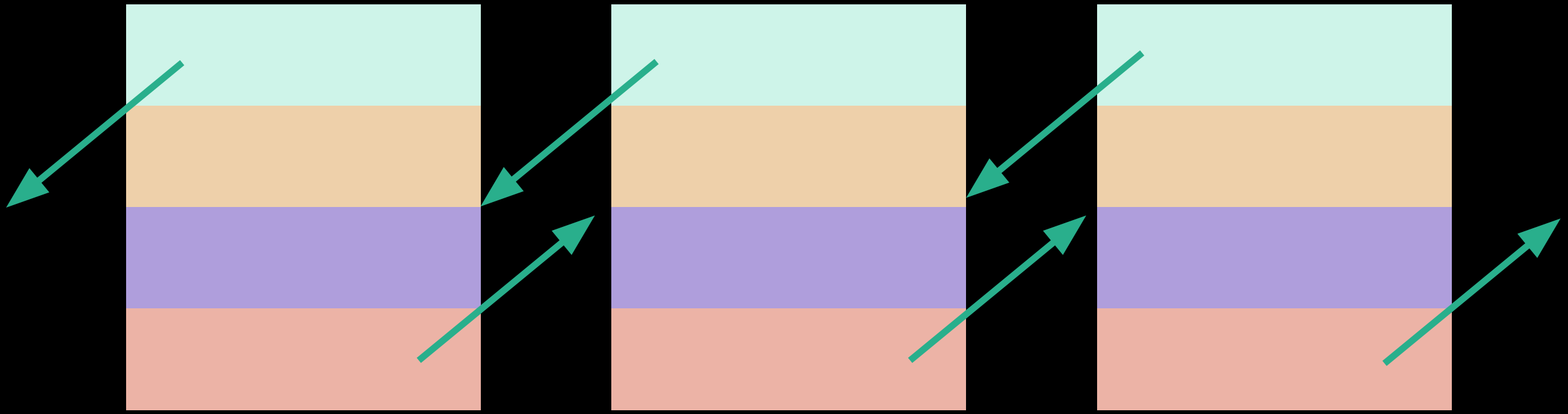
Image source: bitcoin.it

Blockchain

- It's easiest to think of the blockchain as a linked list.

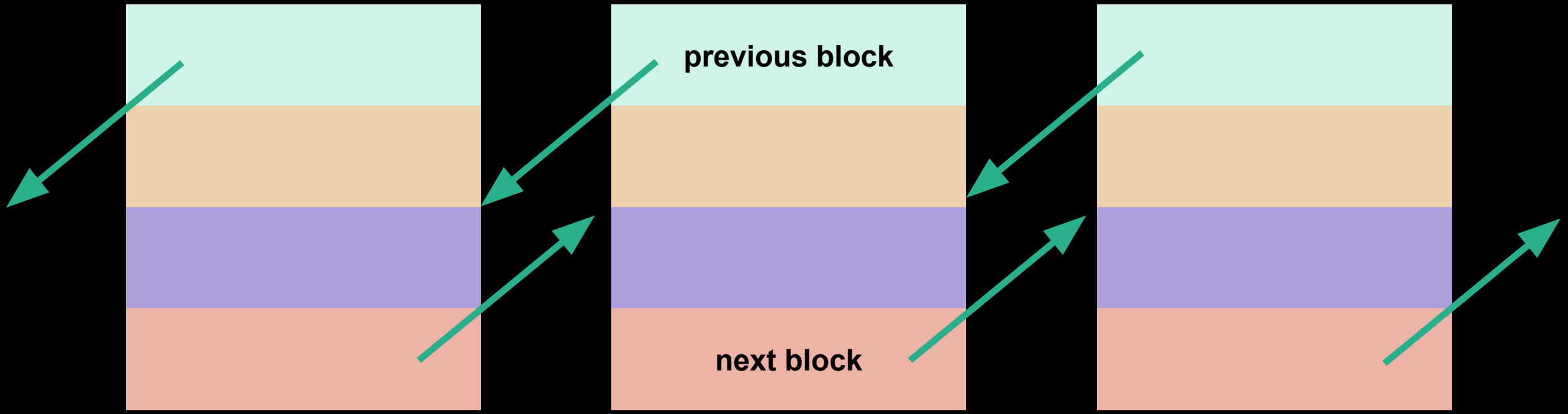
Blockchain

- It's easiest to think of the blockchain as a linked list.



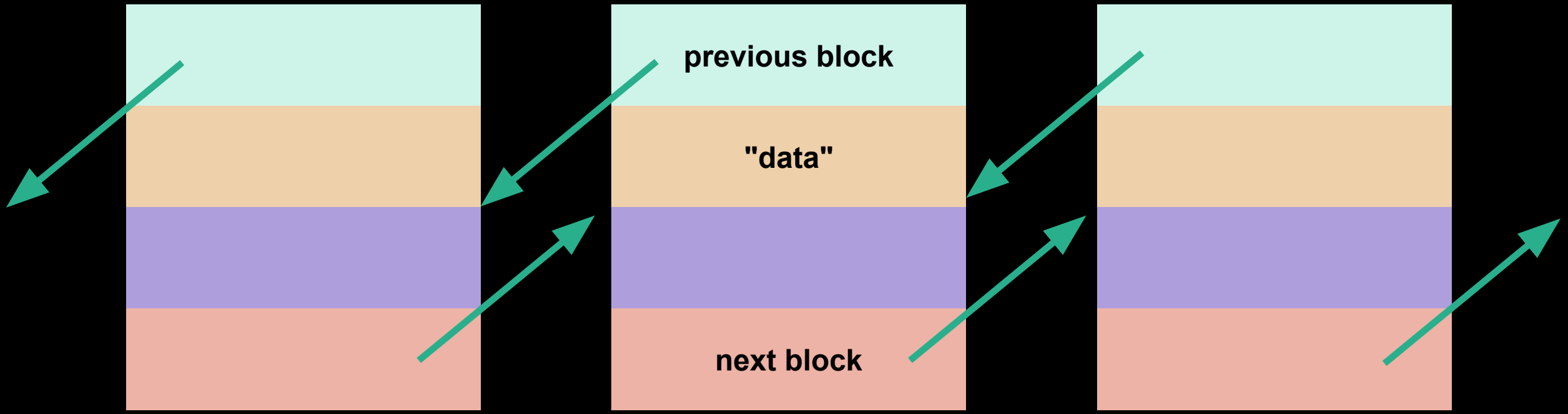
Blockchain

- It's easiest to think of the blockchain as a linked list.



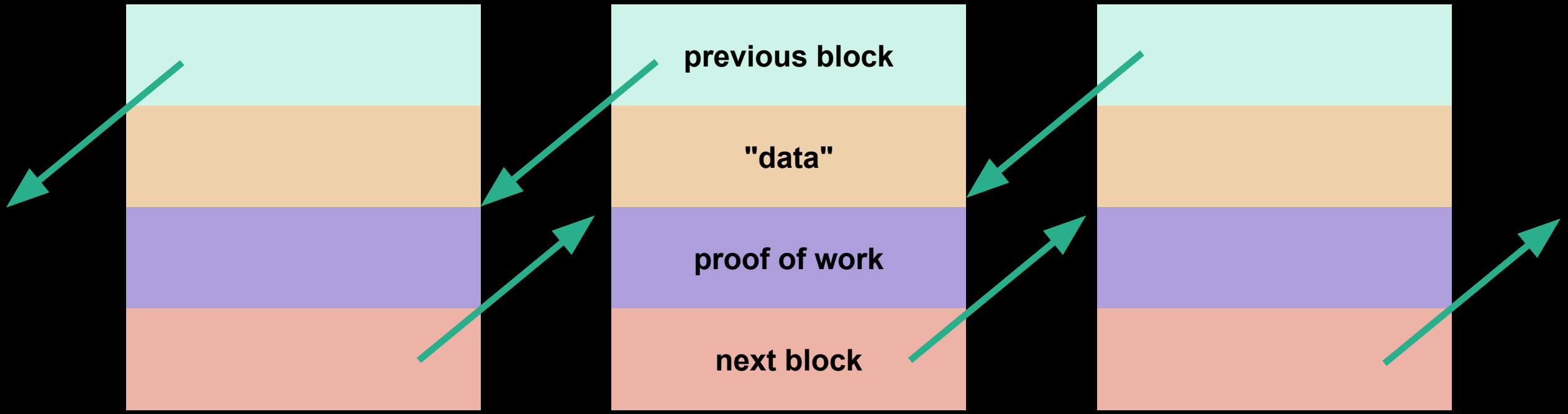
Blockchain

- It's easiest to think of the blockchain as a linked list.



Blockchain

- It's easiest to think of the blockchain as a linked list.



Blockchain

- It's easiest to think of the blockchain as a linked list.

- What's the data? If we're talking about a cryptocurrency, it's a ledger of transactions, each of which is digitally signed by the person who made that transaction.

Blockchain

- It's easiest to think of the blockchain as a linked list.
- What's the data? If we're talking about a cryptocurrency, it's a ledger of transactions, each of which is digitally signed by the person who made that transaction.
- That ledger is also, in the case of a cryptocurrency, *decentralized*, so any time the data is recorded, everyone must record that transaction on their own copy of the ledger, in that block.

Blockchain

- But how do you know the block(chain) is legitimate, if everyone has their own copy and could hypothetically modify it?

Blockchain

- But how do you know the block(chain) is legitimate, if everyone has their own copy and could hypothetically modify it?

- The way many cryptocurrencies do it is to assume the blockchain with the most computational work put into it is the "true" chain.

Blockchain

- But how do you know the block(chain) is legitimate, if everyone has their own copy and could hypothetically modify it?

- The way many cryptocurrencies do it is to assume the blockchain with the most computational work put into it is the "true" chain.

- This leads to the concept of proof of work.

Blockchain

- Recall how hashing works.

Blockchain

- Recall how hashing works.

- We can hash the block, over and over, coupled with some random number, until we find a highly unusual pattern in the first *n* (say, 30 to 40) out of 256 bits.

Blockchain

- Recall how hashing works.
- We can hash the block, over and over, coupled with some random number, until we find a highly unusual pattern in the first *n* (say, 30 to 40) out of 256 bits.
- The smallest change in any of the transactions would produce a totally different hash, making that block unverified (and everything after it potentially fraudulent too.)

Blockchain

- We can very easily verify the correctness of someone's proof of work.

Blockchain

- We can very easily verify the correctness of someone's proof of work.

- The longer a chain gets (in the case of chain conflicts), the more and more likely it is that chain consists only of verified, legitimate transactions.

Blockchain

- We can very easily verify the correctness of someone's proof of work.
- The longer a chain gets (in the case of chain conflicts), the more and more likely it is that chain consists only of verified, legitimate transactions.
- What's a transaction?

Blockchain

- What if the data, instead of being a list of transactions, was something else instead?
  - This is what underpins another blockchain-based technology, Ethereum.

Cryptography