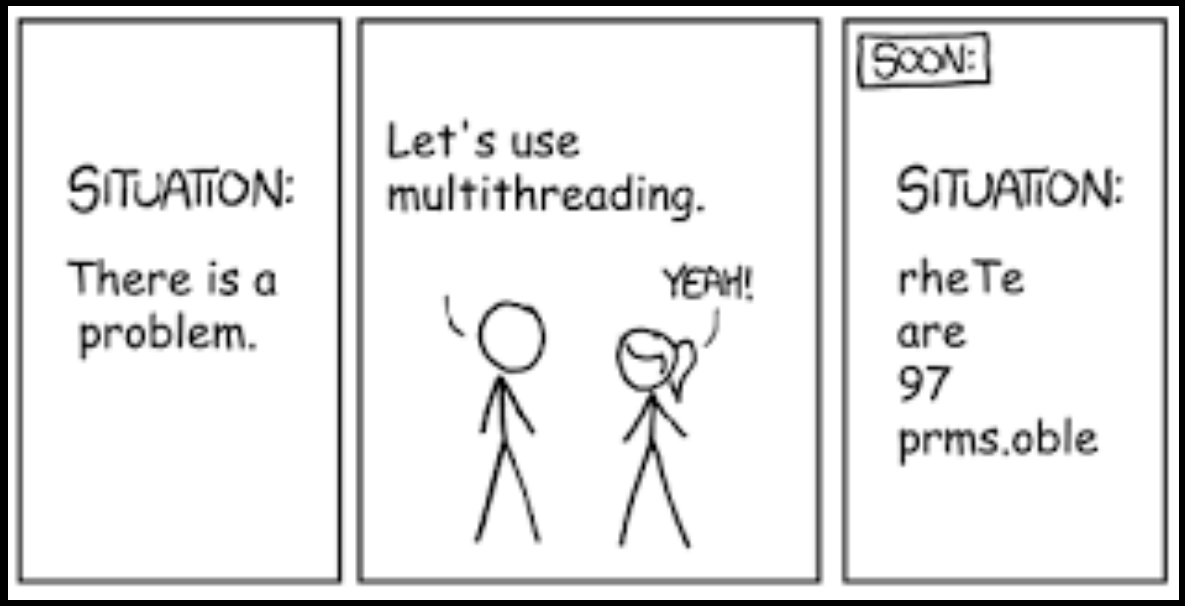
Computer Science: The Good Parts

A Practical Journey for Early-Career Developers

Part 4



Quick Tips

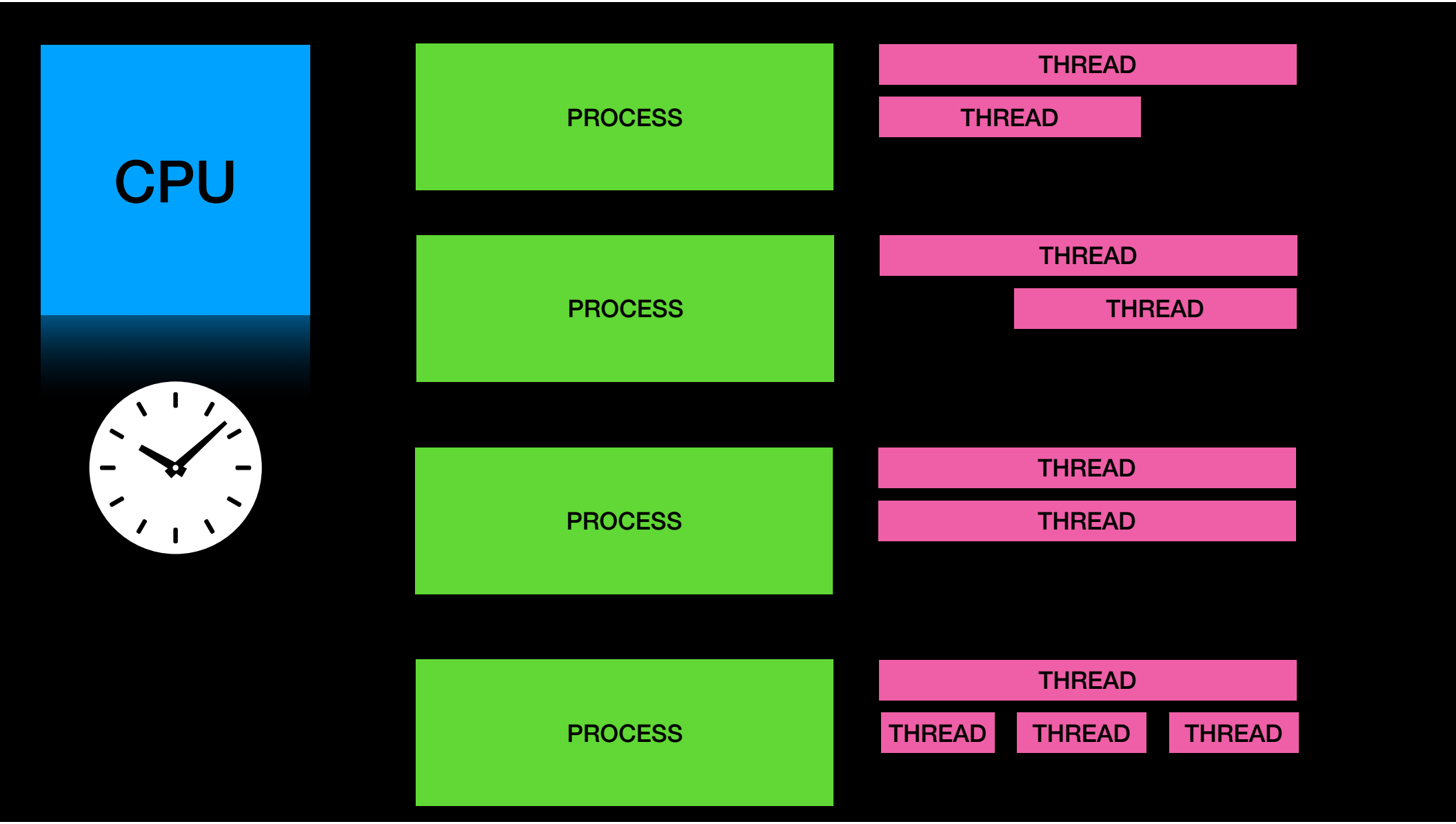
Hex/binary literal notation

Hex/binary conversion

Hex file viewer

Computer Architecture

Threading Model


CPU
PROCESS
PROCESS
PROCESS
PROCESS
THREAD
THREAD
THREAD
THREAD
THREAD
THREAD
THREAD
THREAD
THREAD
THREAD

Threading Model

Every thread has:

ID

Process ID

Private Memory ("thread-local")

Call Stack

Threading Model

Code "runs" on a thread.

By default, all code runs on the main thread.

You can start other threads in order to run other code "concurrently" with the main thread.

```ruby
def sum_series
  total = 0
  upper_limit = 25
  1.upto(upper_limit) do |n|
    total += n
  end
  puts "Sum total: #{total}"
end

```

FRAME

caller

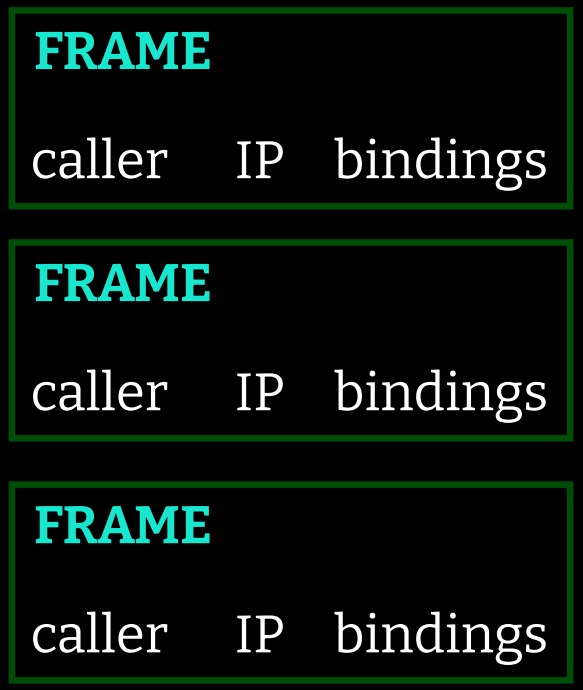
IP

bindings

```ruby
def sum_series
  upper_limit = 25
  total = calc_sum(upper_limit)
  puts "Sum total: #{total}"
end

def calc_sum(zebra)
  total = 0
  1.upto(zebra) do |n|
    total += n
  end
  return total
end

sum_series()
```



Recursion, Part 3

Let's re-examine the Series example.

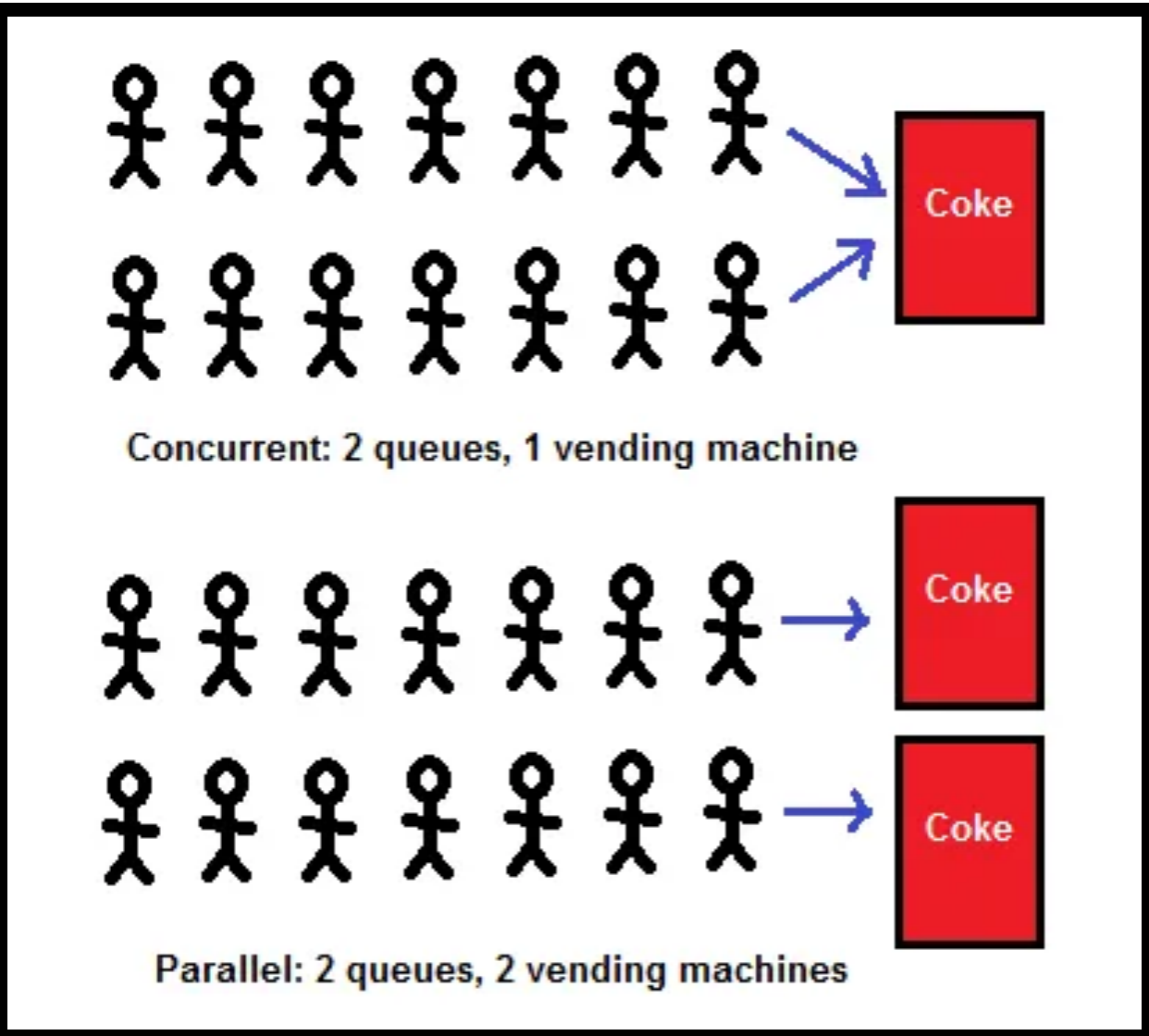
How high can we calculate?

Code Example

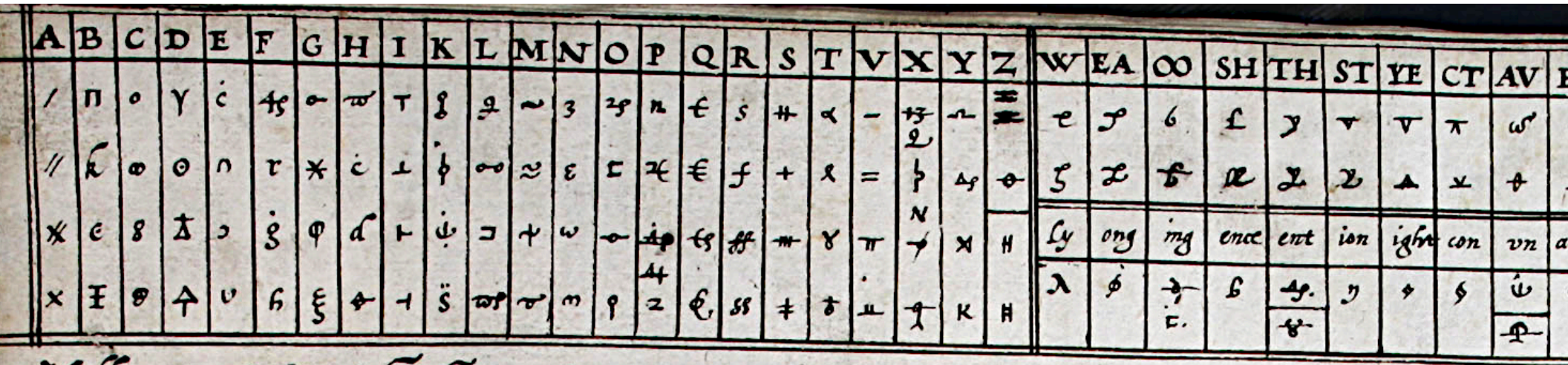
```
def create_user_account(uname, pwd)

  if !User.find_by(username: uname)
    User.create(username: uname, password: pwd)
  else
    raise "You already created your account."
  end

end
```


Coke
Concurrent: 2 queues, 1 vending machine
Coke
Coke
Parallel: 2 queues, 2 vending machines

Modern Cryptography for the absolute beginner

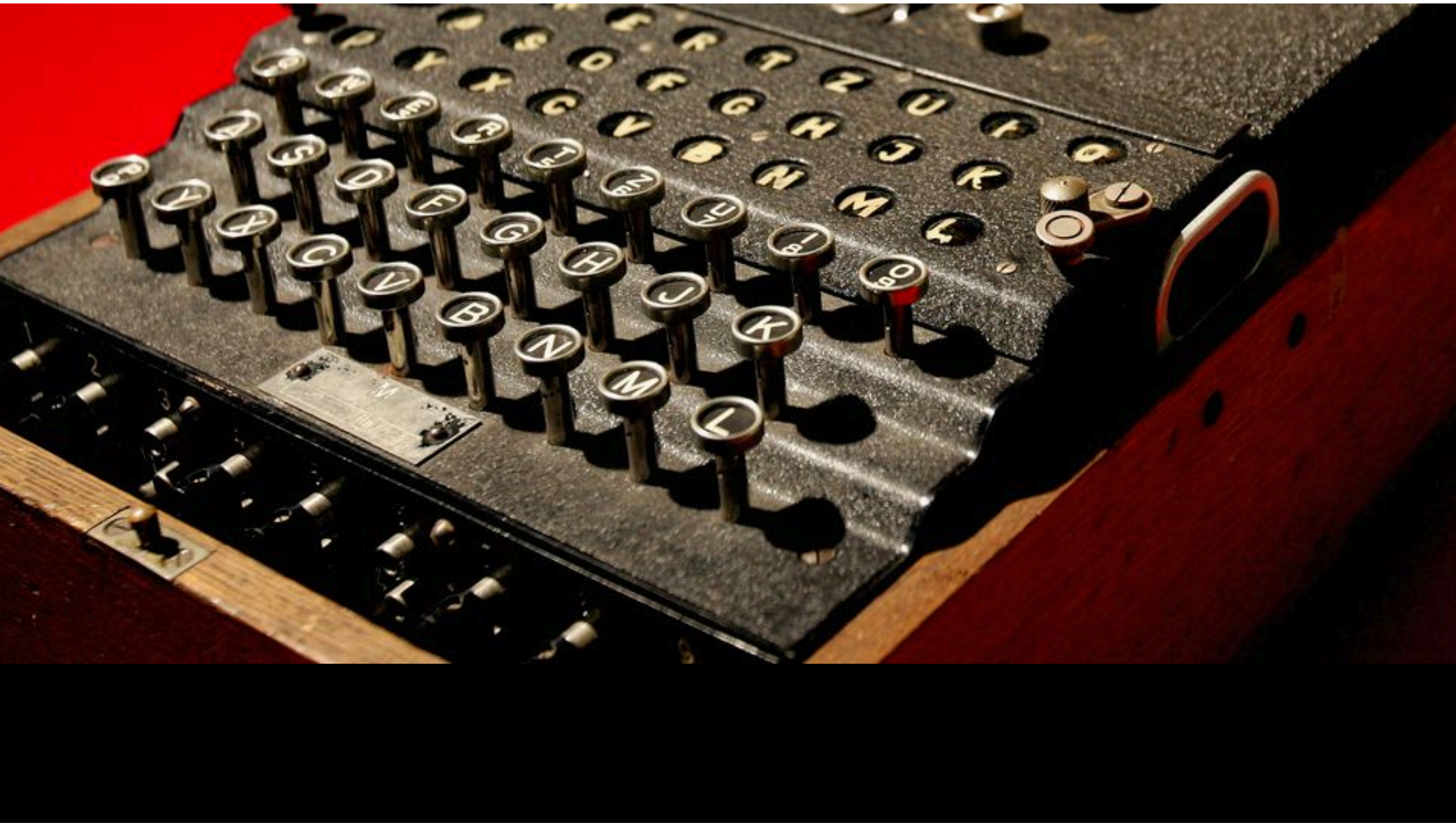


This note. l. shall alwayes double the Caractere. l. preceding the same.

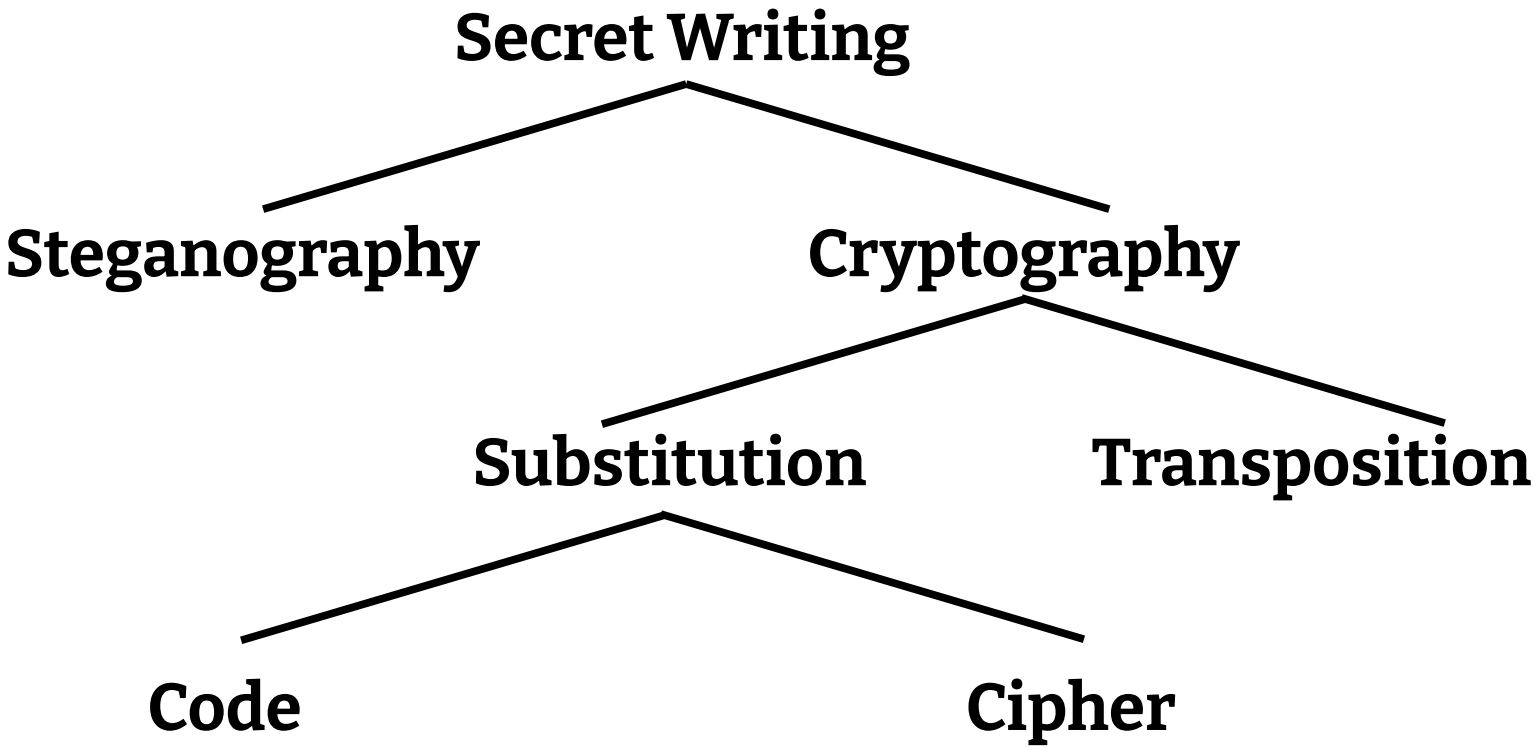
This . ◻ . for the punctuing. | This . V . for Parentheses . | This . J . to of periodes & Sentences

X.	The Pope .	+.	the Earle of Arondel	-o.	the E: of Angous .	-o.	Madame
X.	The king of france	ℓ.	the Earle of Oxforde	wc.	the E: of Atholl .	ℓ.	Maiestie

/.	waye
π.	receaue




Bank Name
1234 5678 9876 5432
1234
MONTH/YEAR
VALID THRU
12/99
CARDHOLDER



From "The Code Book," by Simon Singh. Doubleday, 1999

Quick Example: Base64

Hello there, how are you?

Quick Example: Base64

Hello there, how are you?

Quick Example: Base64

Hello there, how are you?

72 101 108 108 111 32 116 104 101 114 101 44 32
104 111 119 32 97 114 101 32 121 111 117 63

Quick Example: Base64

Hello there, how are you?

72 101 108 108 111 32 116 104 101 114 101 44 32
104 111 119 32 97 114 101 32 121 111 117 63

01001000 01100101 01101100 01101100 01101111 00100000 01110100 01101000 01100101 01110010 01100101 00101100 00100000 01101000 01101111 01110111 00100000 01100001 01110010 01100101 00100000 01111001 01101111 01110101 00111111

Quick Example: Base64

Hello there, how are you?

72 101 108 108 111 32 116 104 101 114 101 44 32
104 111 119 32 97 114 101 32 121 111 117 63

01001000 01100101 01101100 01101100 01101111 00100000 01110100
01101000 01100101 01110010 01100101 00101100 00100000 01101000
01101111 01110111 00100000 01100001 01110010 01100101 00100000
01111001 01101111 01110101 00111111

SGVsbG8gdGhlcmUsIGhvdyBhcmUgeW91Pw==

Cryptography

There are two primary use cases for digital cryptography:

- Verification
- Secrecy

Cryptography

There are two uses cases for verification:

- Message tampering
- Authorship

Use Case: Message Tampering

How can we verify that a message was transmitted perfectly without any accidental changes?

Use Case: Parity Bits

Use Case: Parity Bits

Content:

C A T

Use Case: Parity Bits

Content:

01000011 01000001 01010100

Use Case: Parity Bits

Content:

01000011 01000001 01010100

Even Parity:

11000011 01000001 11010100

Use Case: Parity Bits

Content:

01000011 01000001 01010100

Odd Parity:

01000011 11000001 01010100

Check Digits

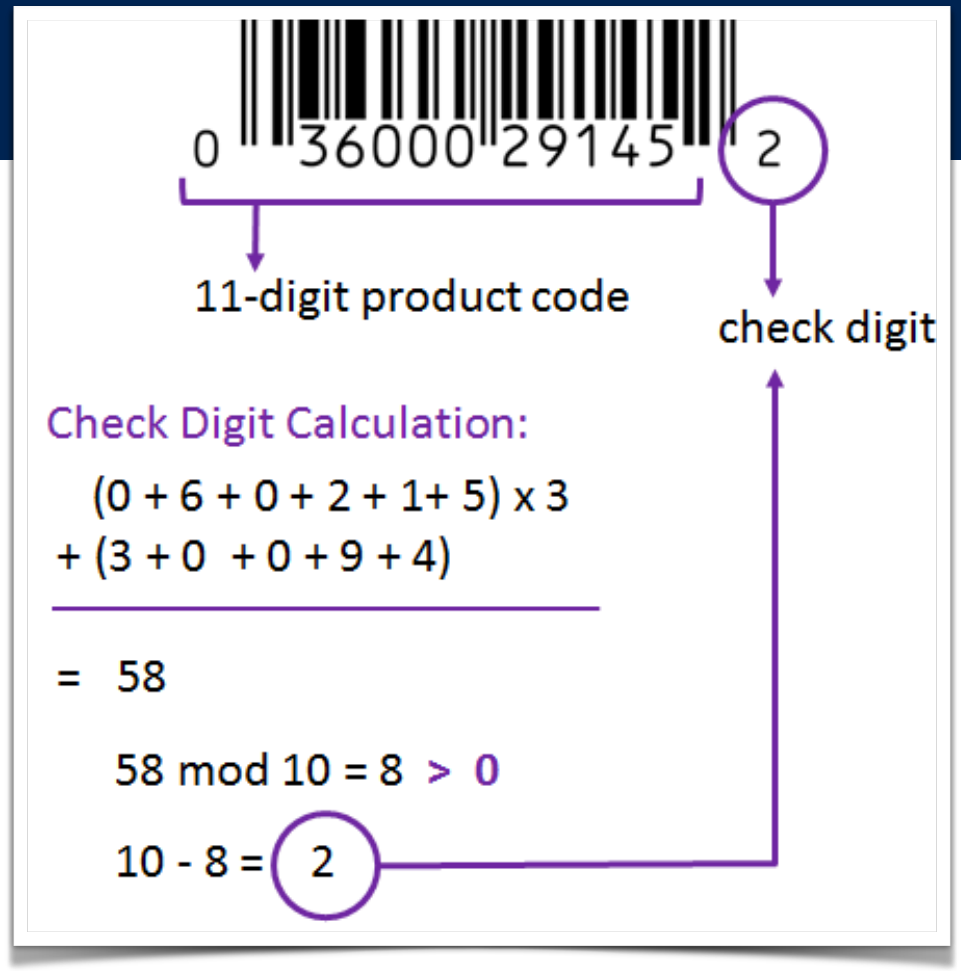
Check Digits

1. Sum the digits in the odd-numbered positions, then multiply by 3.

2. Add the digits in the even-numbered positions to the previous result.

3. Divide by 10, and keep the remainder.

4. If the remainder is not 0, subtract the remainder from 10.



Data Verification

These are all synonyms!

Checksum

Hash

Digest

Fingerprint

MD5

SHA-1

SHA-256

bcrypt

Use Case: Password Security

Password: **swordfish**

bcrypt hash: **$20A6@3AC194F02...**

Reversing this process is impossible.

Symmetric Encryption

Symmetric Encryption

Symmetric Encryption

"Advance by 1"

Symmetric Encryption

"Advance by 1"

Symmetric Encryption

Fortunately, this is reversible

Symmetric Encryption

Fortunately, this is reversible

Symmetric Encryption

But how do we transmit the key?

Asymmetric Encryption

Public Key Cryptography

Each key transforms data.

They are called a *pair* because they mathematically exactly reverse the effect of the other key.

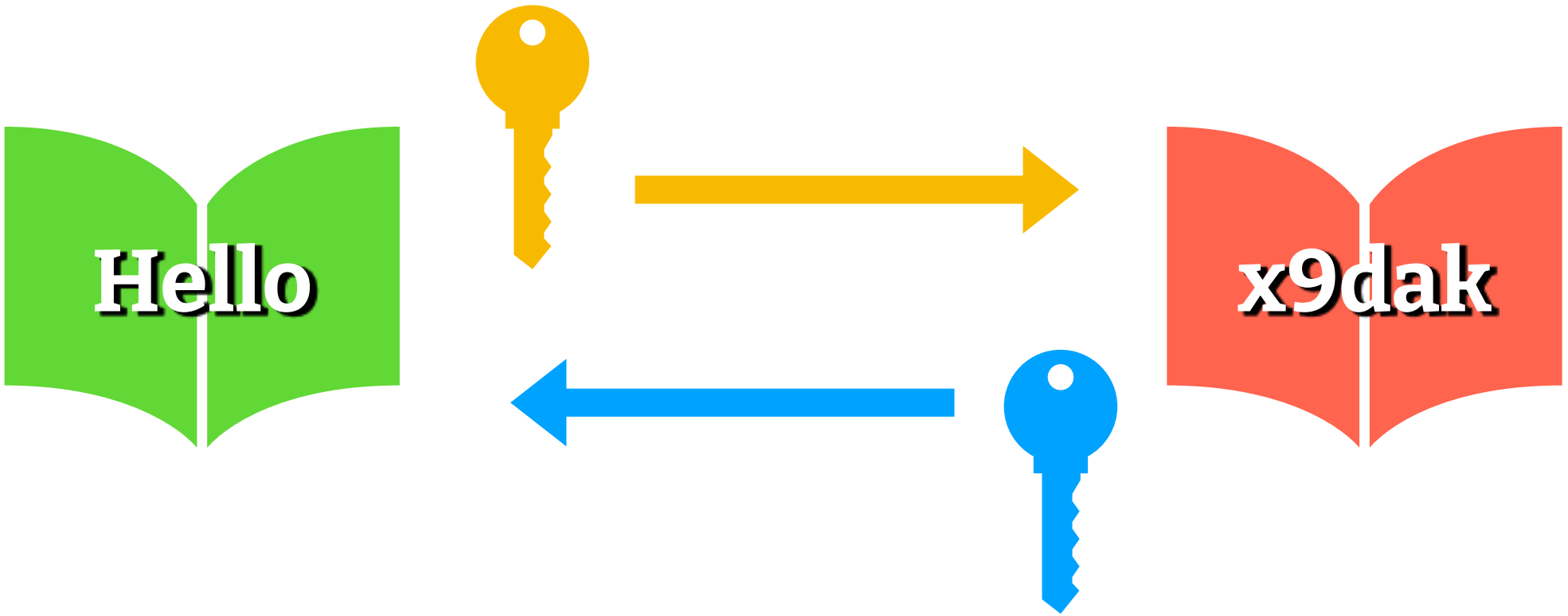
Public Key Cryptography

One key is arbitrarily selected to be the public key, and the other will be the private key.

Public Key Cryptography

Public Key Cryptography

Public Key Cryptography



Use Case: Secret Message

Use Case: Secret Message

Mr. A wants to send a secret message to Mr. B.

Mr. B,
Meet me at noon for lunch.
Your friend, Mr. A

Use Case: Secret Message

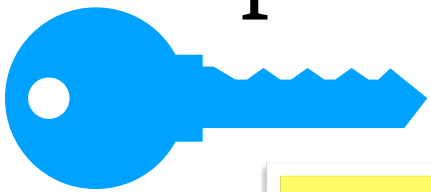
STEP 1:
Mr. A encrypts the message with Mr. B's public key.

Mr. B,
Meet me at noon for lunch.
Your friend, Mr. A

Li%8aja^@*9cmakA
P91&*C9Naxw8723h
Yolq&6209CKn02K

Use Case: Secret Message

STEP 2:
Mr. B decrypts the message with Mr. B's private key.



Mr. B,
Meet me at noon for lunch.
Your friend, Mr. A

Mr. B,
Meet me at noon for lunch.
Your friend, Mr. A

Use Case: Authenticity

A1. Mr. A calculates the content's digest.

A2. Mr. A encrypts the digest with his private key. This is the "wax seal".

B1. Mr. B decrypts the digest value using Mr. A's public key.

B2. Mr. B independently calculates the digest of the received content.

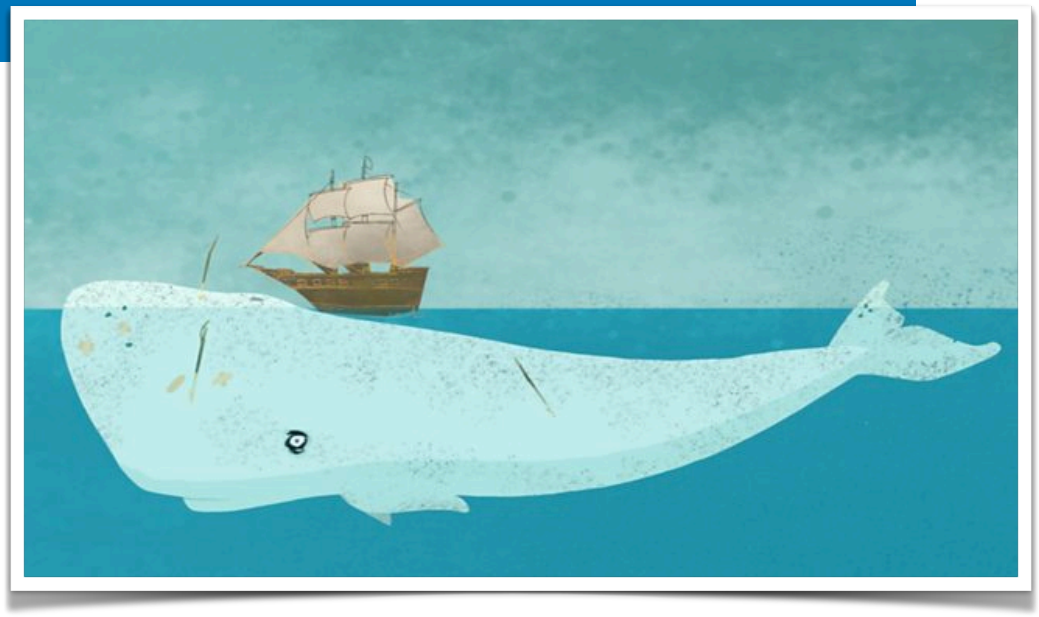
B3. Mr. B expects the digests to match!

Wait a Minute!

ssh-keygen -t rsa -b 4096

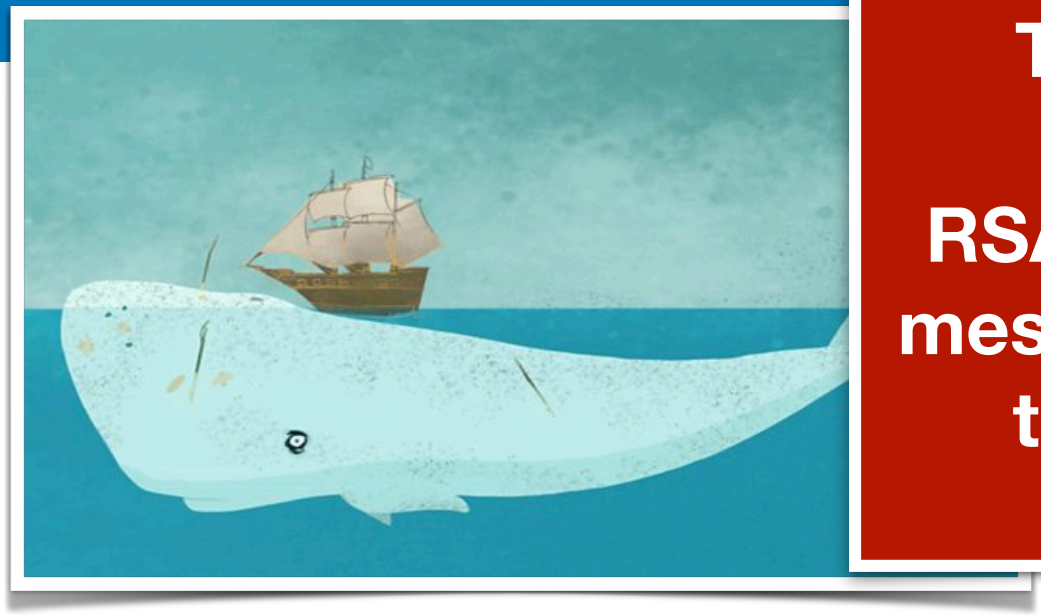
Wait a Minute!

ssh-keygen -t rsa -b 4096



Wait a Minute!

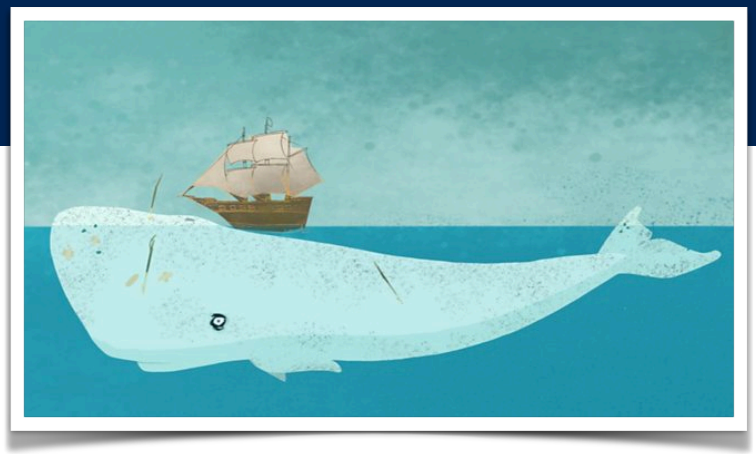
ssh-keygen -t rsa -b 4096



This won't work!

RSA can only encrypt messages shorter than the "key length".

Wait a Minute!



In reality, we use both!

We use asymmetric cryptography to securely transmit a random symmetric key.

What's Next?

Now: PKCS

Soon: Elliptic-curve PKCS

Future: Quantum implications?

Bletchley Park Monument

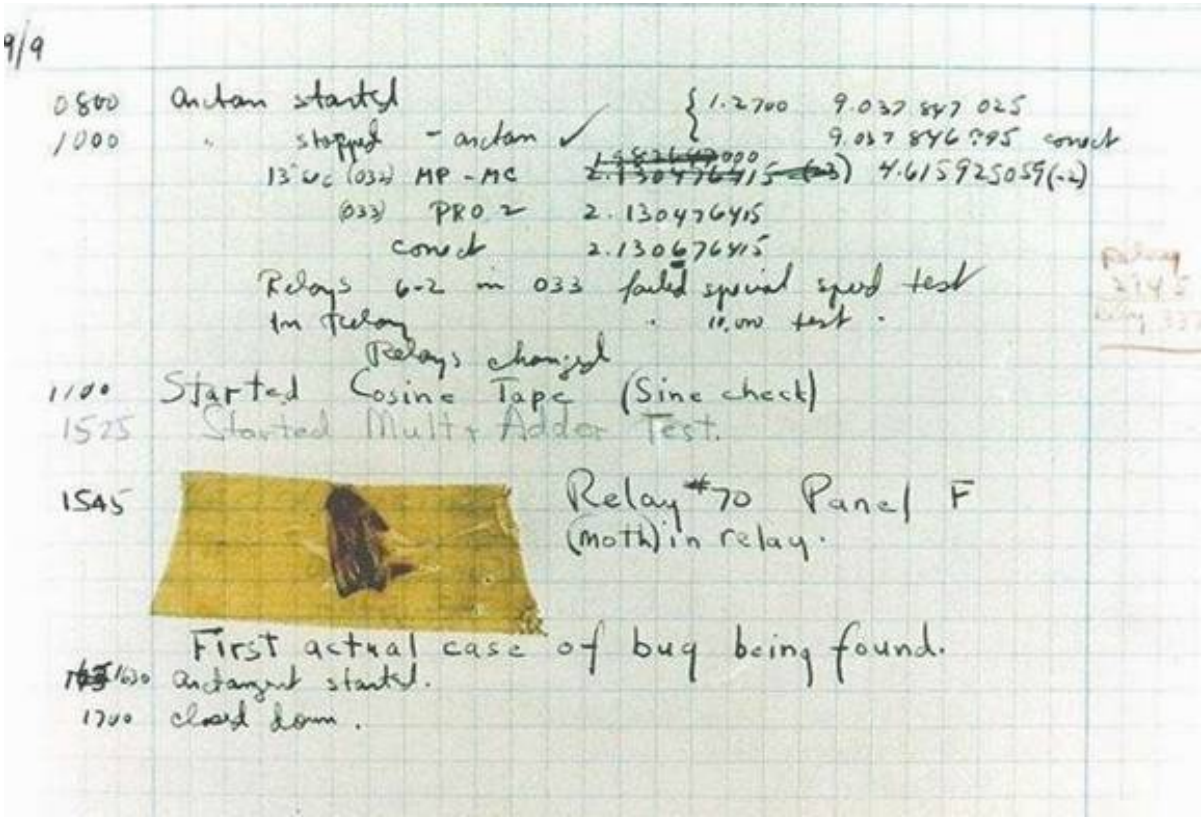
The Best Parts

Alan Turing

It's a way of thinking

Grace Hopper

Grace Hopper



Grace Hopper

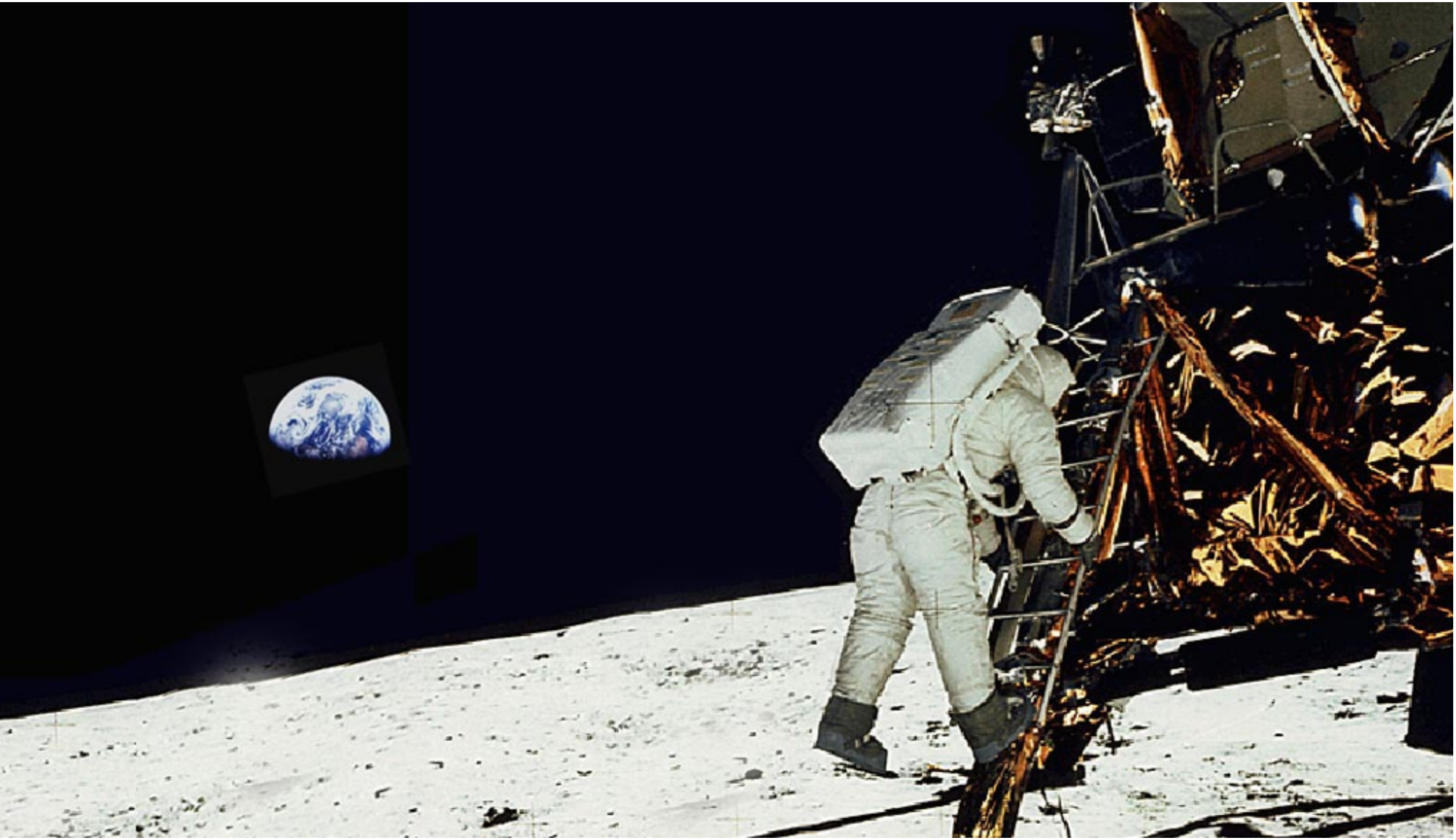
Humans are allergic to change. They love to say, "We've always done it this way." I try to fight that.

That's why I have a clock on my wall that runs counter-clockwise.

Grace Hopper

**A ship in port is safe;
but that is not what
ships are built for.**

**Sail out to sea and do
new things.**



Margaret Hamilton

Source code of the Apollo Guidance Computer

Computer Science: The Good Parts

THANK YOU!

jeff@purpleworkshops.com

other workshops in 2025:

React for Python/Ruby Developers

Rails for Python/.NET Developers