Blockchain, Cryptocurrencies & Digital Tokens Demystified

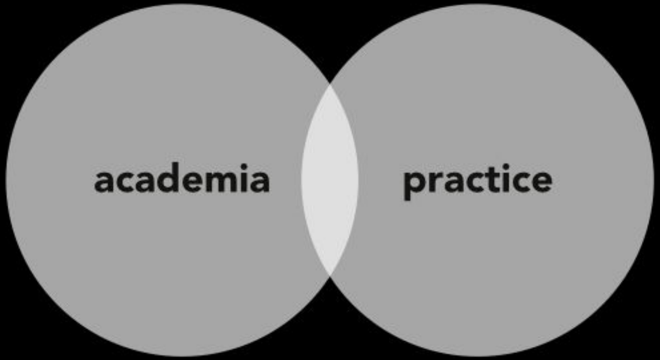
Prof. R.A. Farrokhnia
Columbia Business School
Fall 2023 (EMBA)

Welcome & Agenda

About the Course Faculty

- Prof. R.A. Farrokhnia (***far.oak.nia***)
- Teaching at Columbia **Business & Engineering Schools**
- Recipient of Dean’s Award for Teaching Excellence

business
engineering
storytelling
design



tech + data
start-ups
private equity
venture capital

About the Course Faculty

- Prof. R.A. Farrokhnia (***far.oak.nia***)
- Teaching at Columbia **Business & Engineering Schools**
- Recipient of Dean's Award for Teaching Excellence
- Executive Director (Dean's Office) of "**Advanced Projects and Applied Research in Fintech**" at Columbia Business School
- Board Member & Senior Lecturer: Columbia Journalism School KB Program
- Building a next-gen **DevLab**

fintech.gsb.columbia.edu

Columbia Business School
AT THE VERY CENTER OF BUSINESS™

Advanced Projects and Applied Research in Fintech

About Projects & Research Courses Fellowship Events Contact

The Future of Financial Services

Advanced Projects and Applied Research in Fintech ("APAR") is a multidisciplinary initiative at the intersection of business and engineering. Its two primary goals are:

1. to research the innovative forms and functions of new enterprise and consumer financial services products, and
2. to explore the development of novel technological solutions and oversee their industry implementation.

Before we begin …

farrokhnia@gsb.columbia.edu

Class Schedule - Nov 4, Nov 18, Dec 2, Dec 9

Class Plan

Nov 4	08:30 am to 6:45 pm (K-440)	Module 1 + 2
Nov 18	08:30 am to 6:45 pm (K-440)	Module 3 + 4
Dec 2	08:30 am to 6:45 pm (K-440)	Midterm Project + 5 & 6 + Guest Speaker
Dec 9	08:30 am to 6:45 pm (K-440)	Module 7 & 8 + Guest Speaker + final presentations

Daily Schedule

8:30-9:45 am	**Lecture**
9:45-10:00 am	*Break*
10:00-11:15 pm	**Lecture**
11:15 am-12:30 pm	Lunch (1h15min) - Kravis 2nd floor (Smith Dining)
12:30-2:00 pm	**Lecture**
2:00-2:15 pm	*Break*
2:15-3:30 pm	**Lecture**
3:30-3:45 pm	*Break*
3:45-5:00 pm	**Lecture**
5:00-5:15 pm	*Break*
5:15-6:45 pm	**Lecture**

Curriculum Roadmap

	Nov 4	Nov 18	Dec 2	Dec 9
Morning	Networks & Protocols	Hashing, Hashing Tables & One- Way Functions & a few more tech	Bitcoin + other forms of crypto payments and store of value mechanisms and media	DeFi & Other Applications (Digital Tokens, CBDC, etc.) + Speaker: Future of Finance + Discussion Forum
	Lunch	Lunch	Lunch	Lunch
Afternoon	Encryption & Cryptography (plus some math!)	**Bring it All Together:** Let's build a blockchain & discuss variety of cases	Ethereum & Other Digital Tokens + Speaker: Regulatory & Legal Considerations in Blockchain & Digital Assets	Governance, Marketplaces, NFTs & More; Final Lecture on How the Future May Play Out + Final Presentations

Administrative Requirements

- Please be **on time and present** for the duration of the class
- Class content is **sequential**. Don't miss class sessions (and watch recordings if you do)
- Lots of technical topics, but I won't use ANY code or much math (only 2-3 parts might be tough - I'll give you the heads-up when we reach these points in our curriculum), so don't worry :-)
- I can explain it to you, but I cannot understand it for you! So be sure to ask questions
- Your breaks are my breaks too! I'll provide ample opportunities for Q&A in class though
- Office hours by appointment (just email me)
- Make sure to read the syllabus
- **CBS code of conduct, incl. during guest speaker presentations**
- Team formations: finalized by Nov 18 no later than 3:30 pm ET (today is even better!)
- Midterm Project
- Final Papers and deliverables: all the details
- Final Papers due on **Monday Dec 18 at 5 pm ET**
- This is a **demanding class,** and we are all in it together. Let's make it the best class we can
- **My promise to you all +** let's have a fun, productive course … worthy of a 5 out of 5

DISCLAIMER

One more thing …
Digital Device Policy Recommendation +
Sharing of Class Slides

Also a reminder of a good practice

The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking

Psychological Science
2014, Vol. 25(6) 1159–1168

Reprints and permissions:
sagepub.com/journalsPermissions.nav
DOI: 10.1177/0956797614524581
pss.sagepub.com
SAGE

Pam A. Mueller[1] and Daniel M. Oppenheimer[2]
[1]Princeton University and [2]University of California, Los Angeles

Abstract
Taking notes on laptops rather than in longhand is increasingly common. Many researchers have suggested that laptop note taking is less effective than longhand note taking for learning. Prior studies have primarily focused on students' capacity for multitasking and distraction when using laptops. The present research suggests that even when laptops are used solely to take notes, they may still be impairing learning because their use results in shallower processing. In three studies, we found that students who took notes on laptops performed worse on conceptual questions than students who took notes longhand. We show that whereas taking more notes can be beneficial, laptop note takers' tendency to transcribe lectures verbatim rather than processing information and reframing it in their own words is detrimental to learning.

Also a reminder of a good practice

The Pe
Advant
Note T

Pam A. Mu
[1]Princeton Unive

Abstract

Taking notes on laptops rather than in longhand is increasingly common. Many researchers have suggested that laptop note taking is less effective than longhand note taking for learning. Prior studies have primarily focused on students' capacity for multitasking and distraction when using laptops. The present research suggests that even when laptops are used solely to take notes, they may still be impairing learning because their use results in shallower processing. In three studies, we found that students who took notes on laptops performed worse on conceptual questions than students who took notes longhand. We show that whereas taking more notes can be beneficial, laptop note takers' tendency to transcribe lectures verbatim rather than processing information and reframing it in their own words is detrimental to learning.

Also a reminder of a good practice

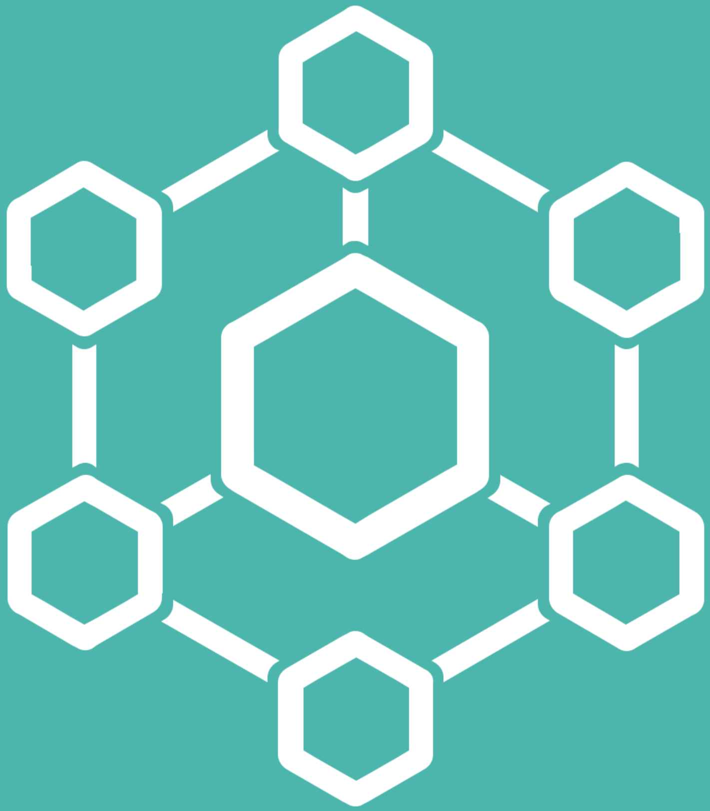
Class is mostly slides for Day 1 and 2 + we'd switch to discussions & whiteboarding (no slides) on subsequent days

All done? Then let’s go … but first, a little fun!

https://www.facebook.com/CryptoCurrencyTM/videos/bitcoin-commercial-what-is-it-and-how-does-it-work/288600998335966/

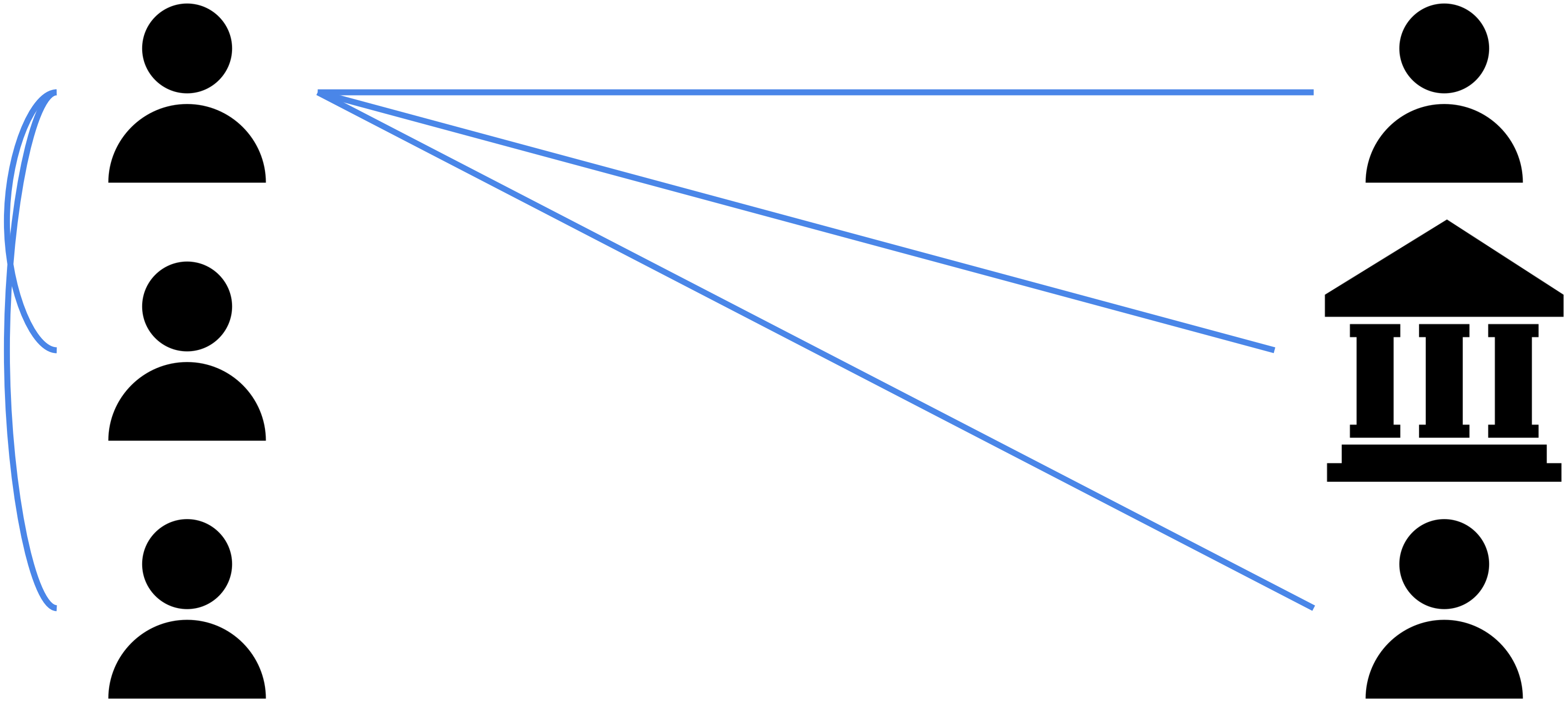
I. A Series of Tubes

How does the internet work?
Why do we need to protect it?

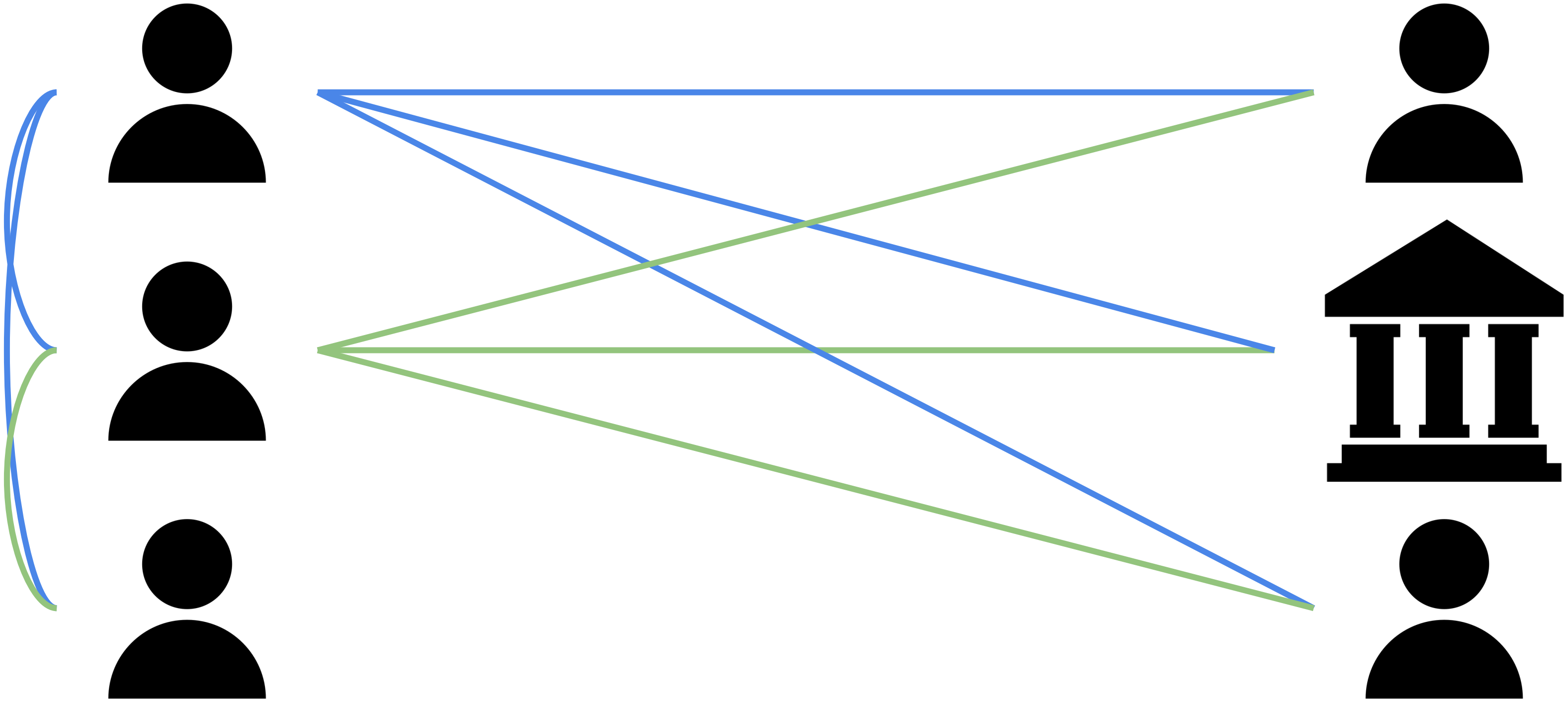


An ideal network

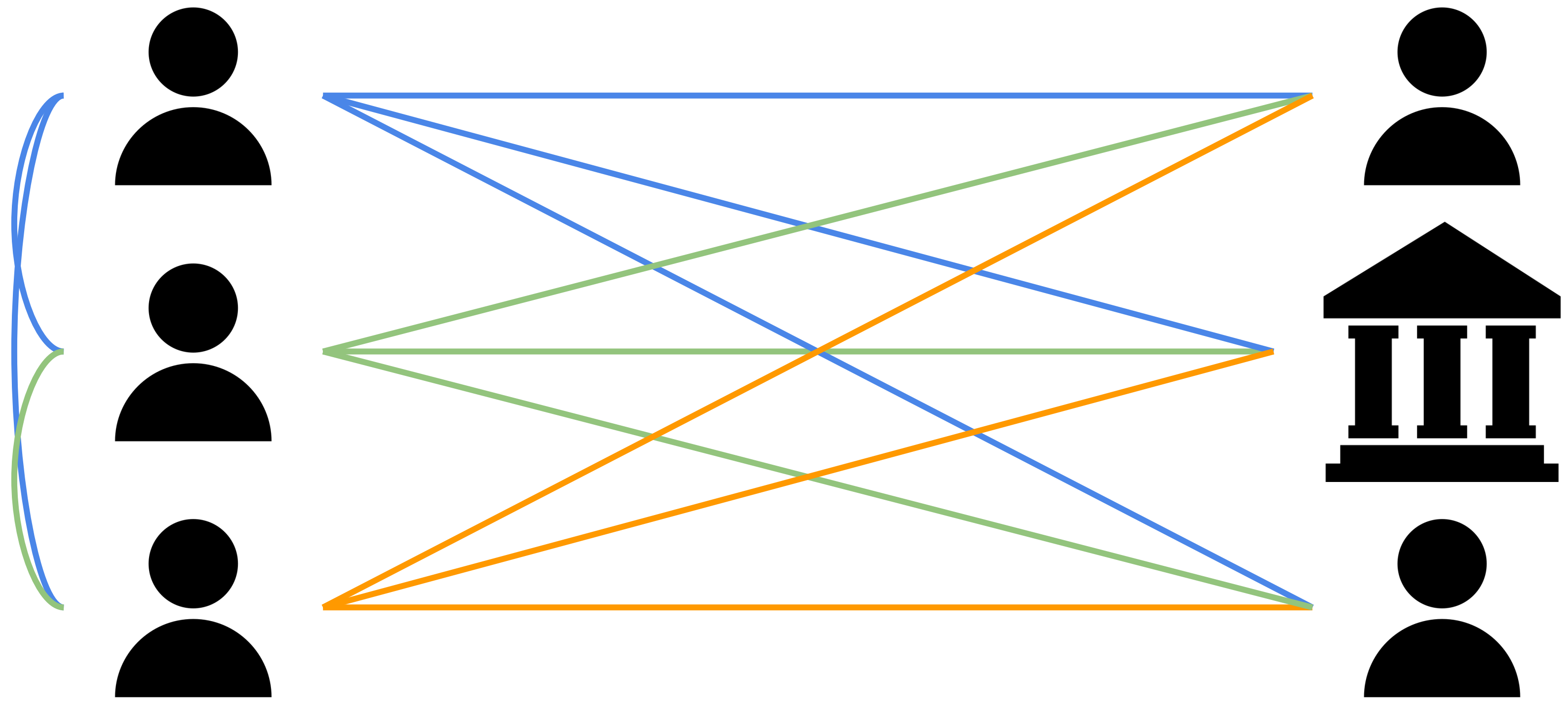
An ideal network



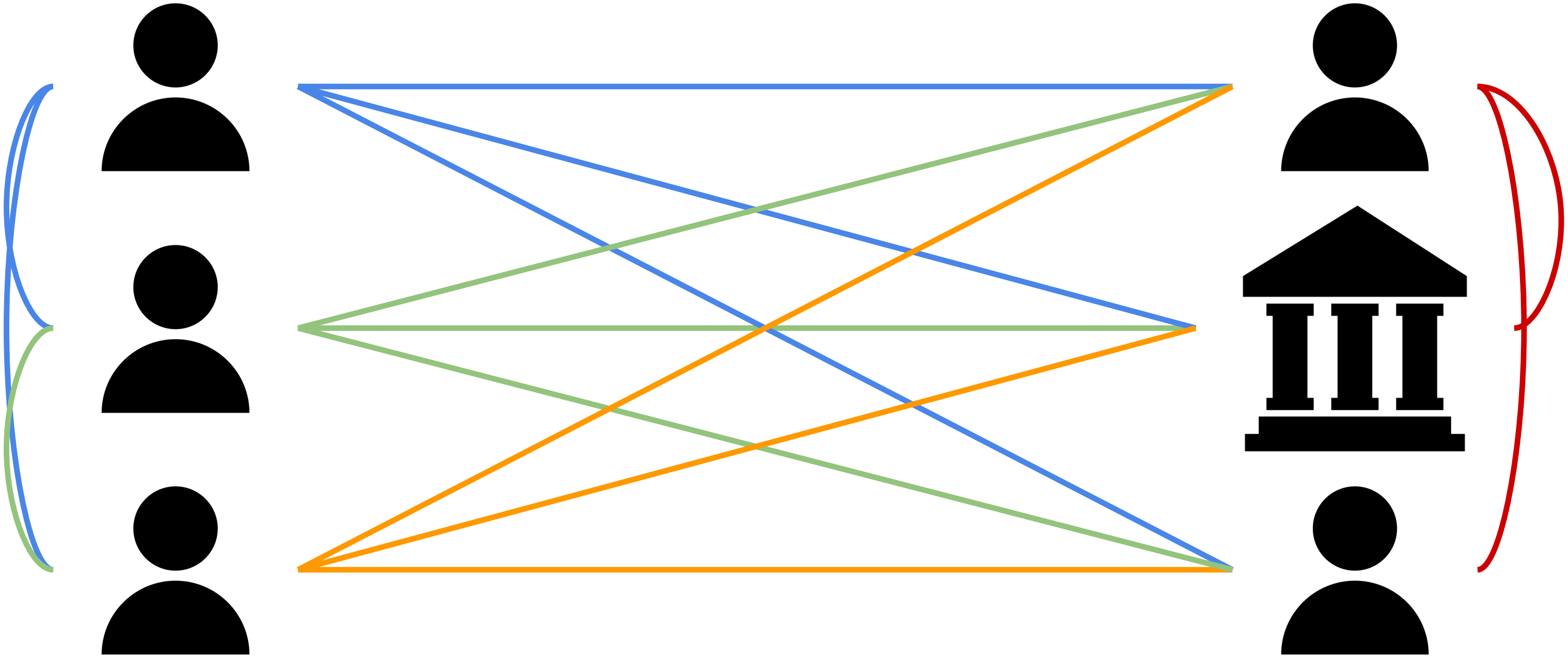
An ideal network



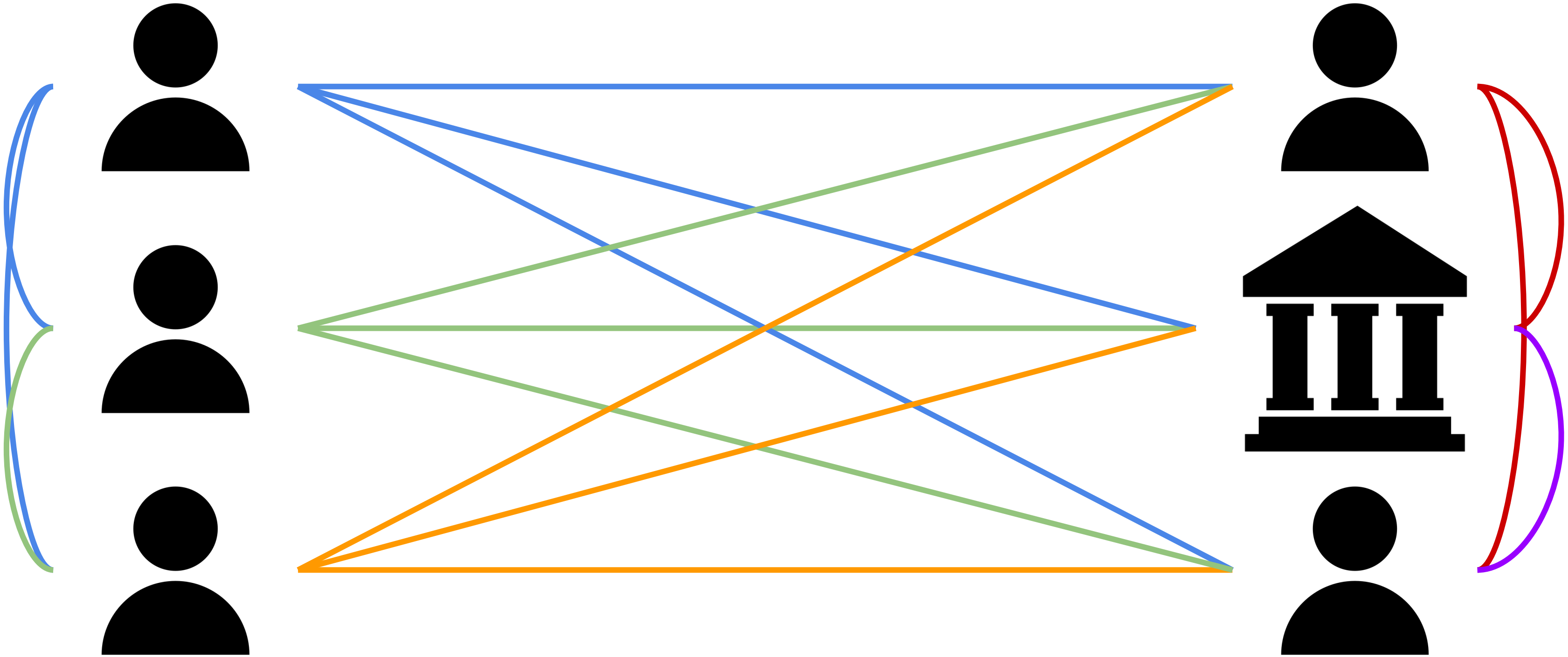
An ideal network



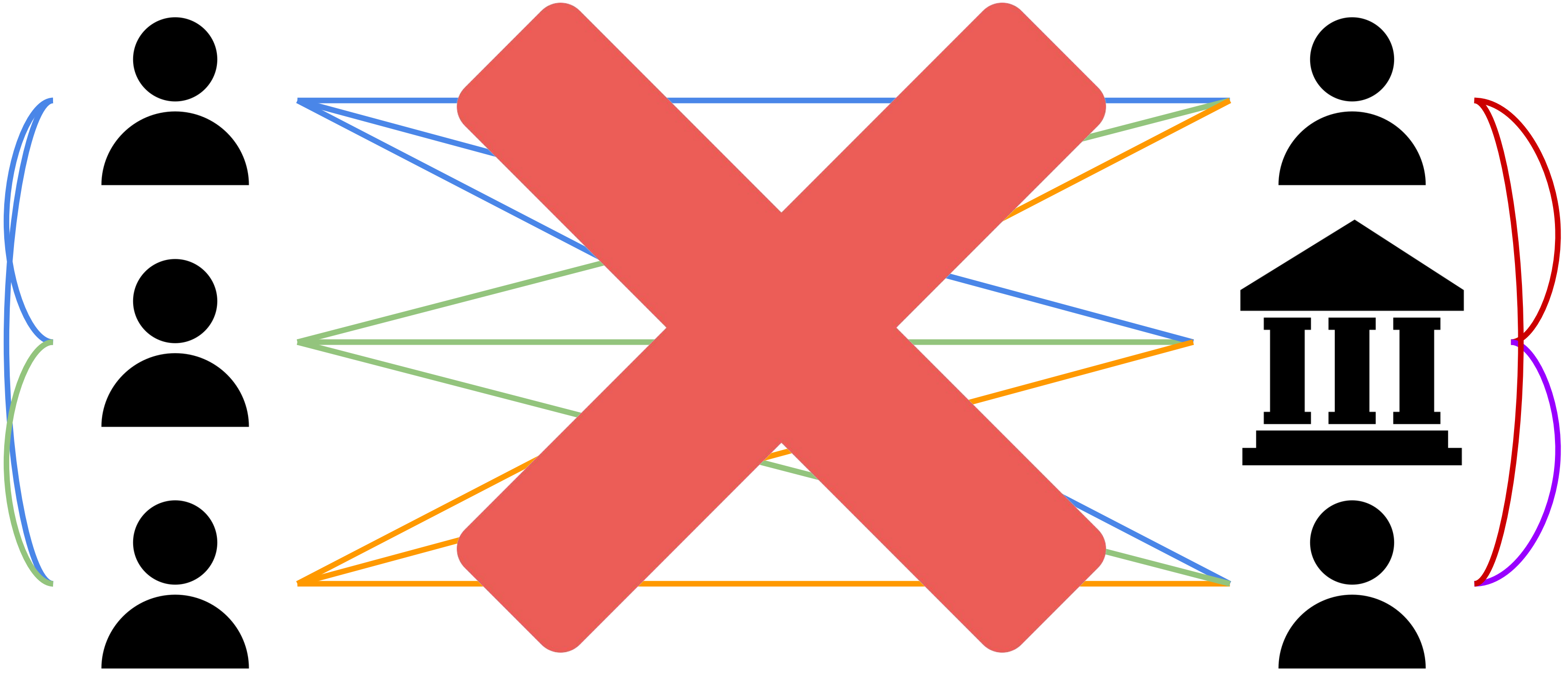
An ideal network



An ideal network

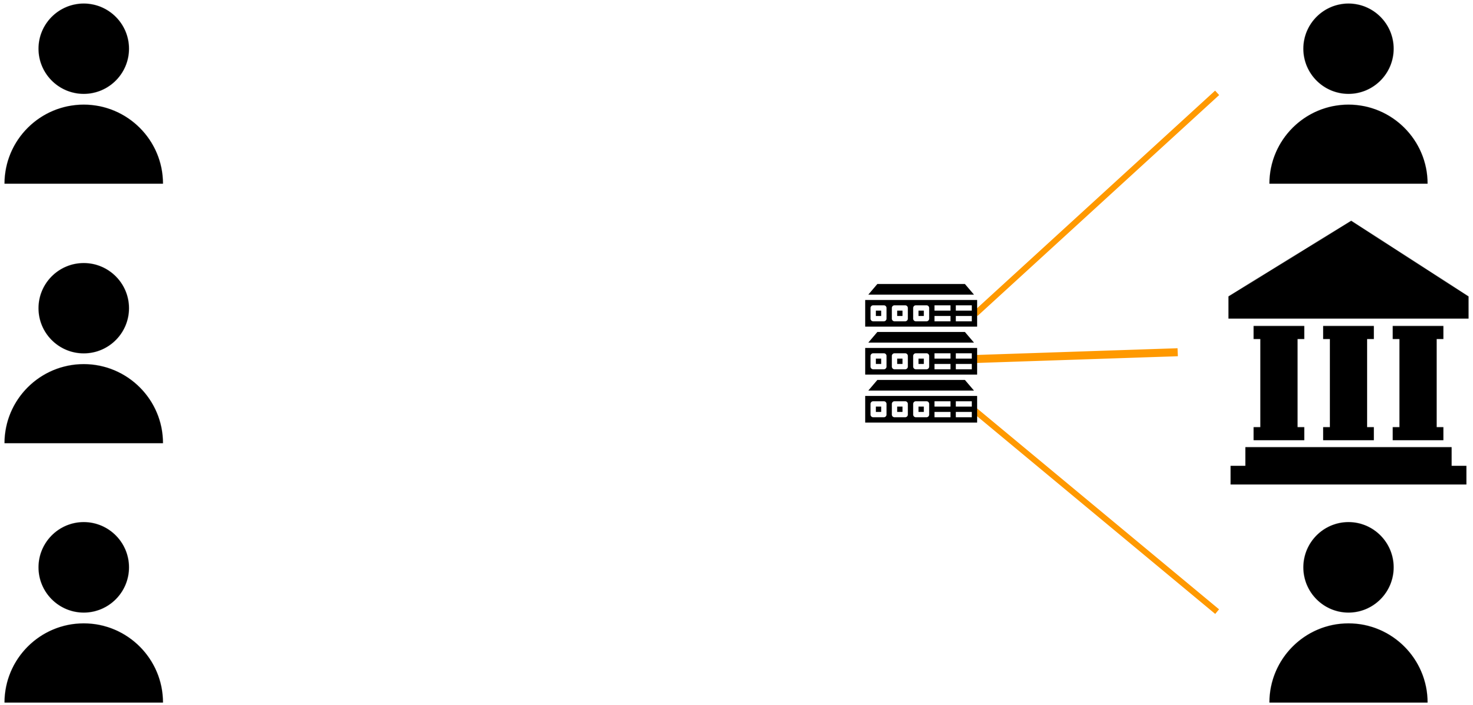


An ideal network

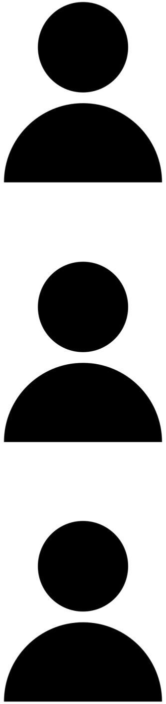


The real world

The real world: Routers & Switches

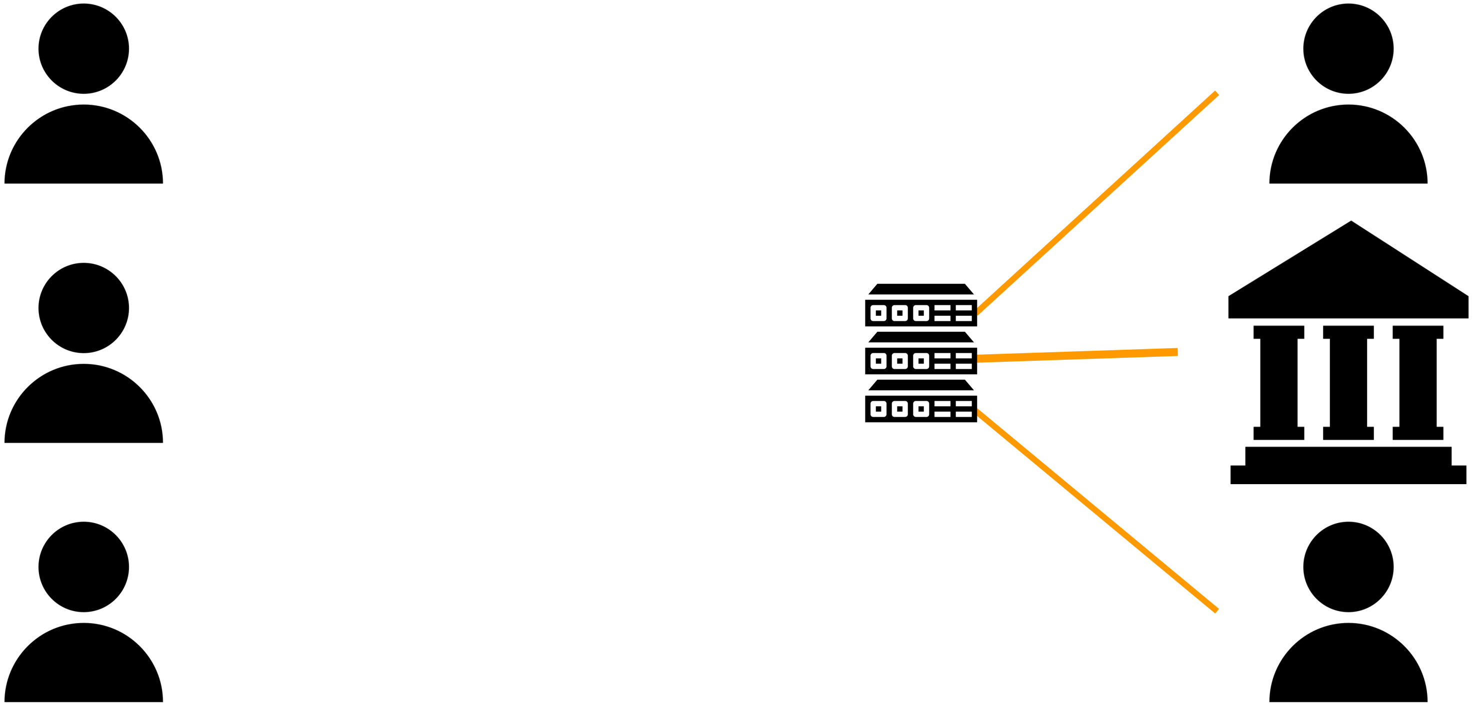


The real wo

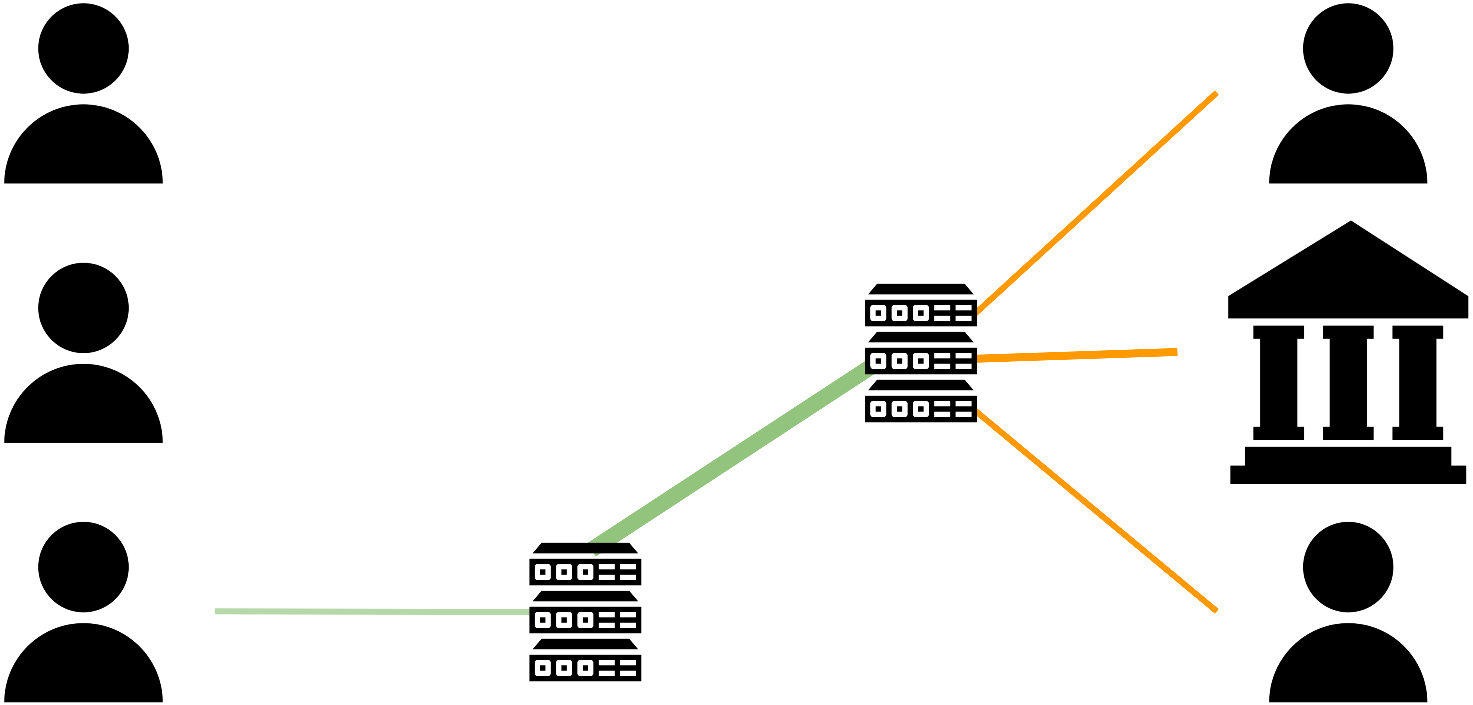


The real world … as it was!

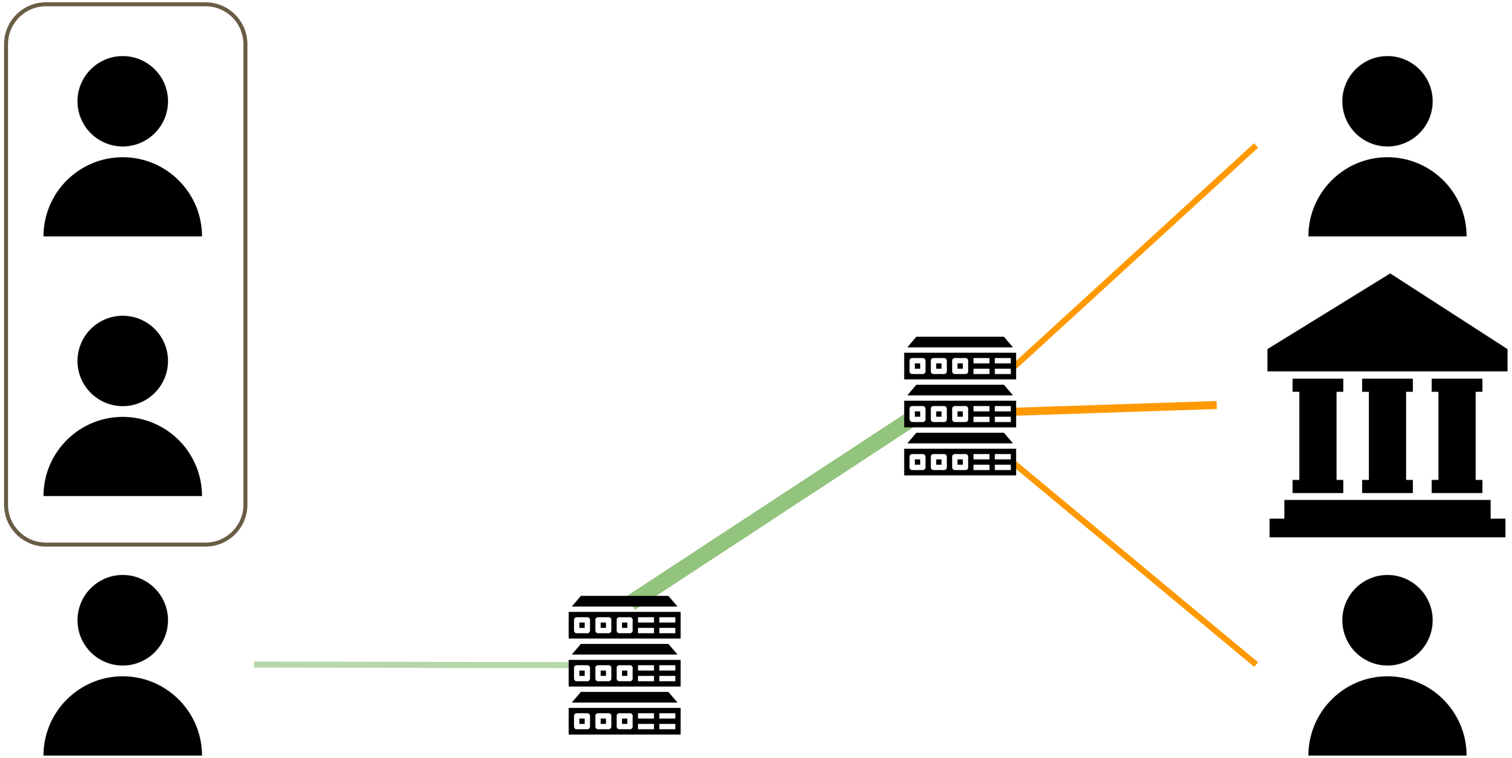
The real world … with PROTOCOLS!



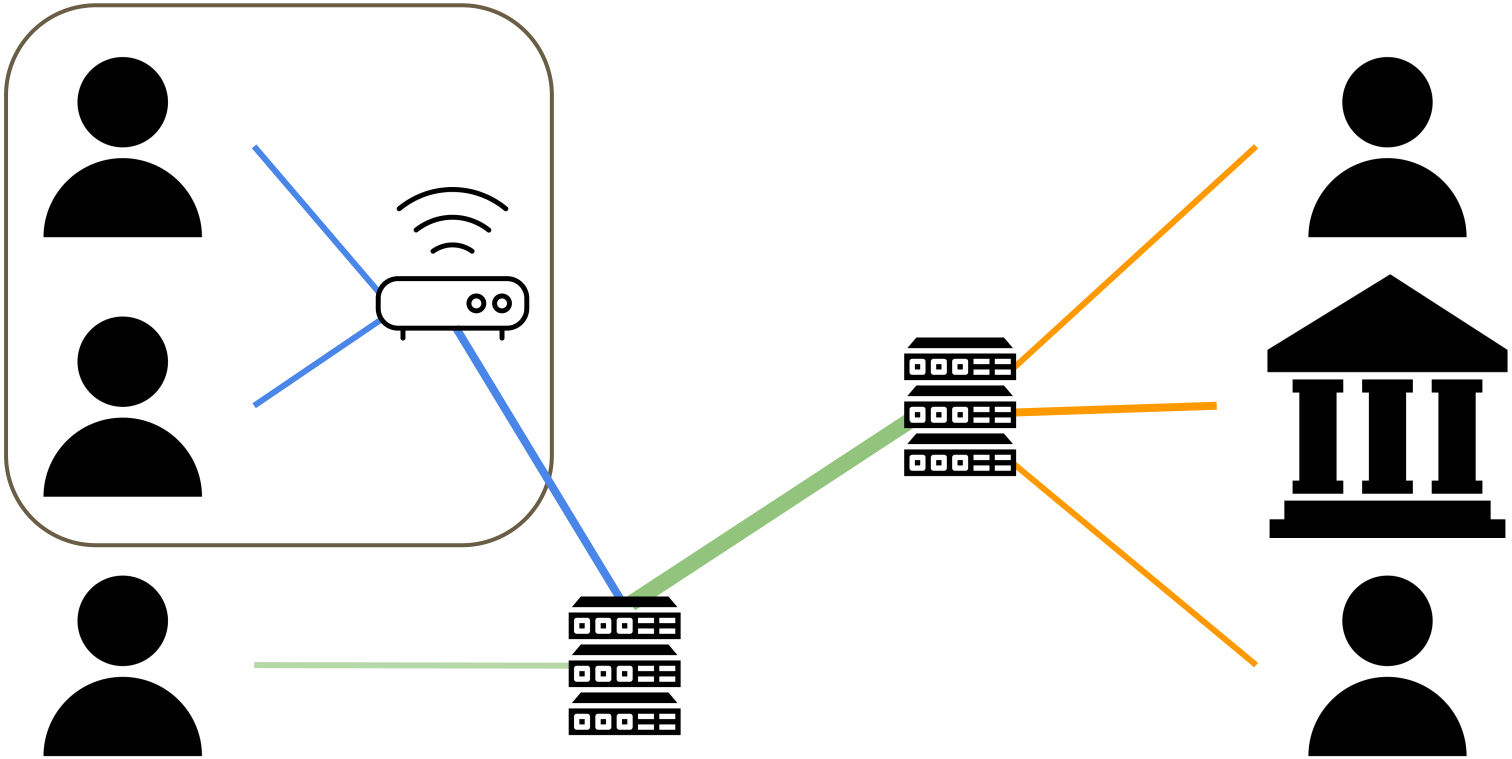
The real world



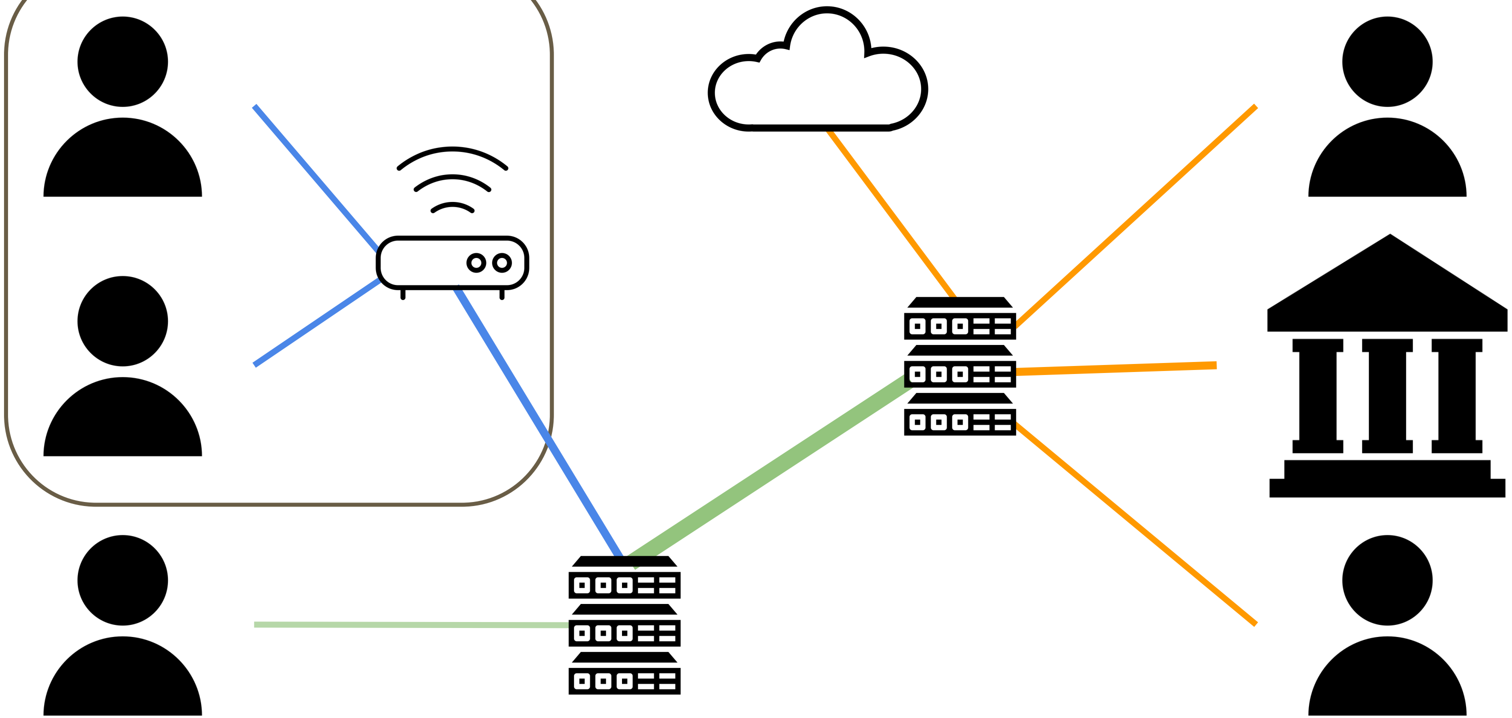
The real world



The real world

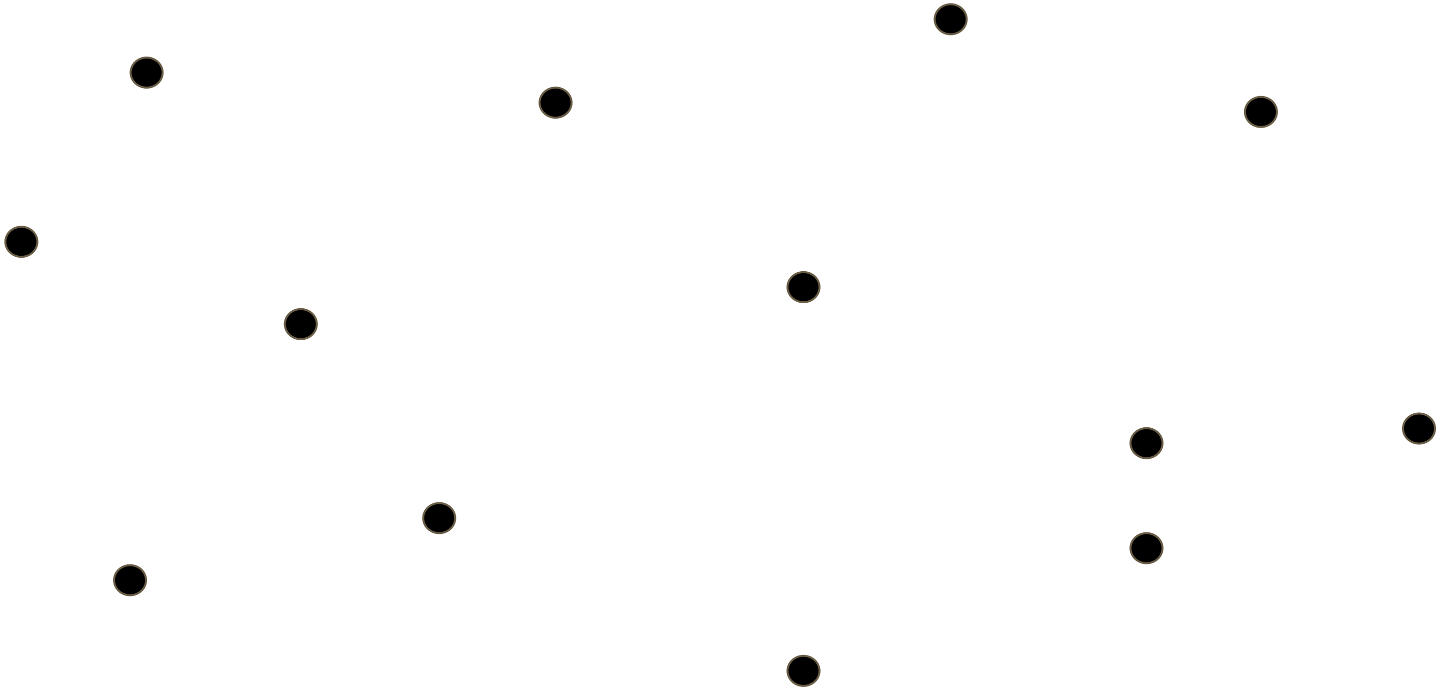


The real world

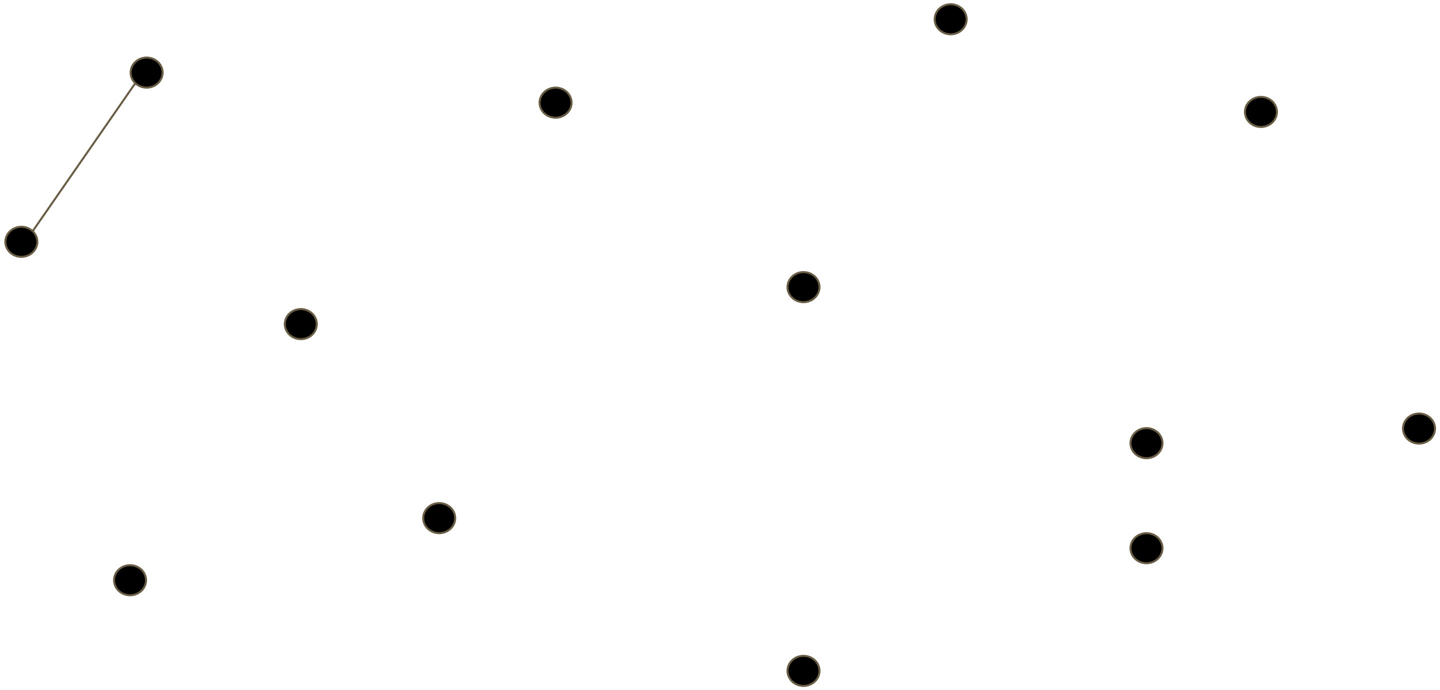


A few words on networks … in the context of order, complexity, and resiliency

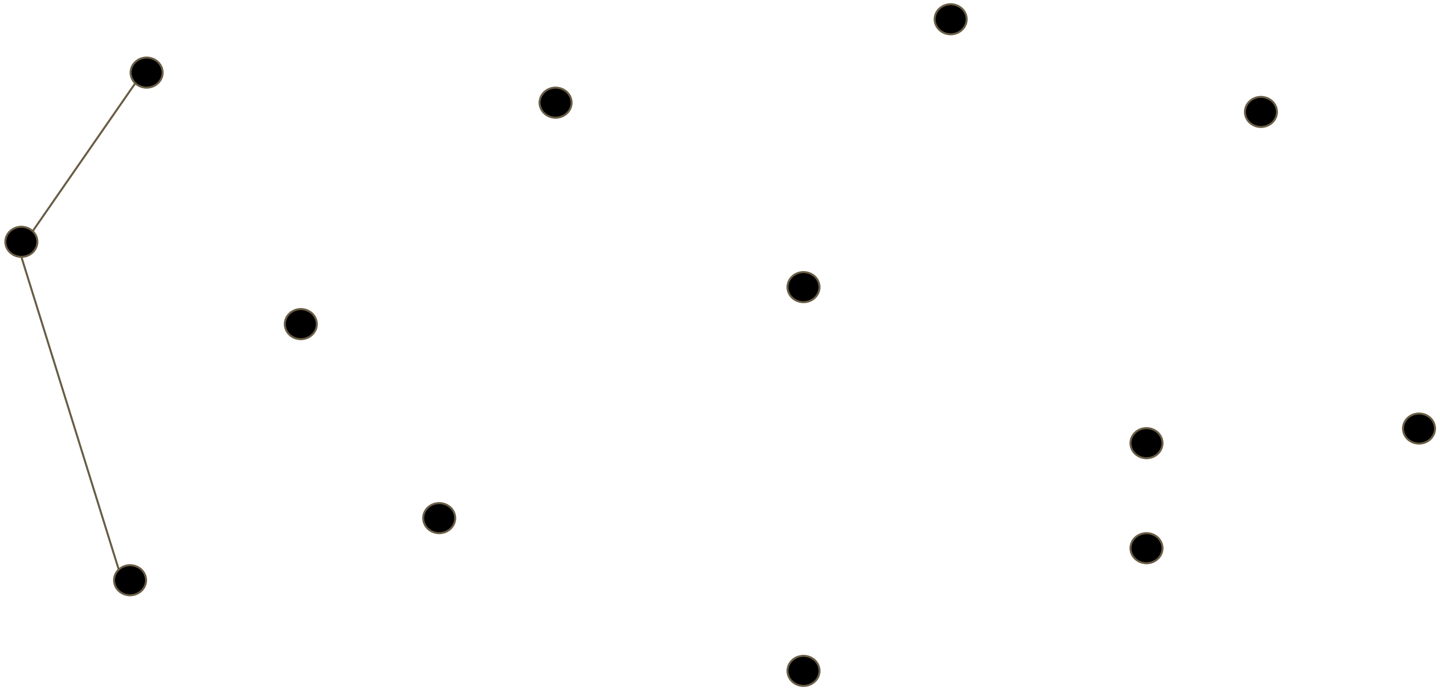
Networks: a collection of connected nodes



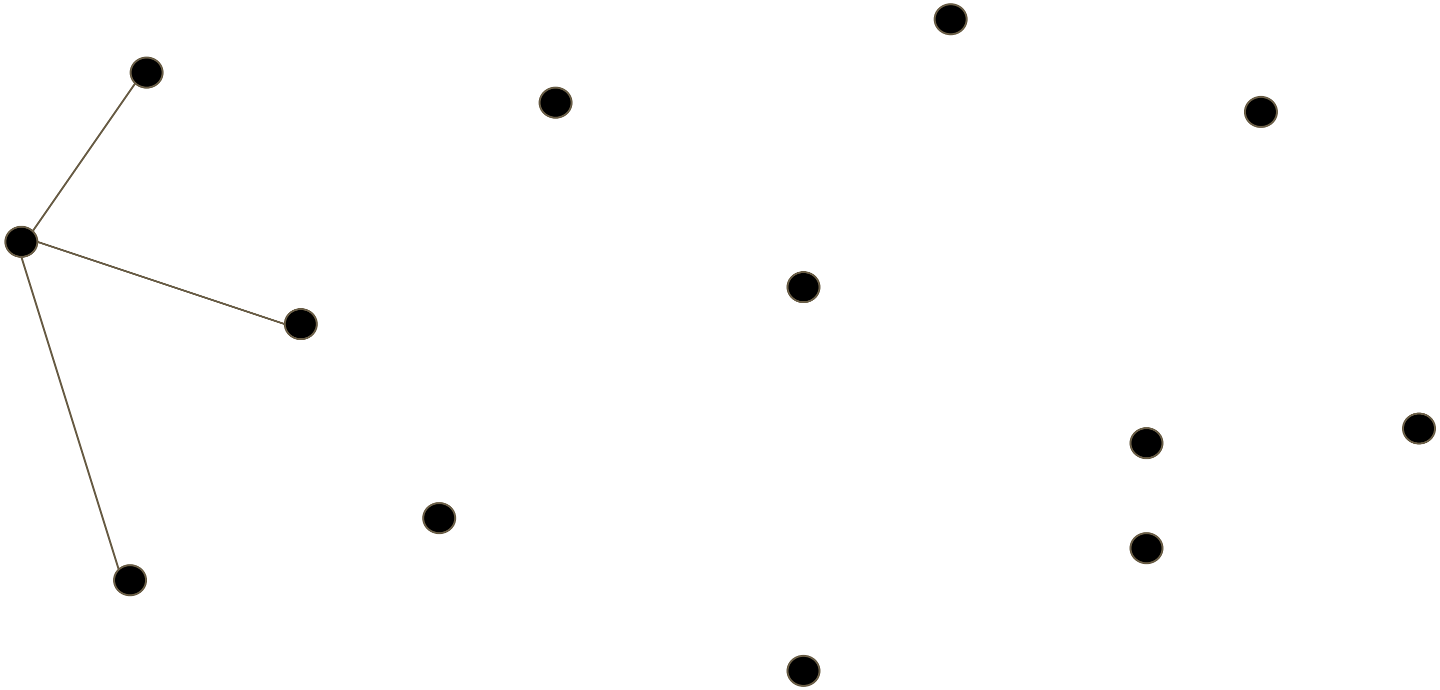
Networks: a collection of connected nodes



Networks: a collection of connected nodes

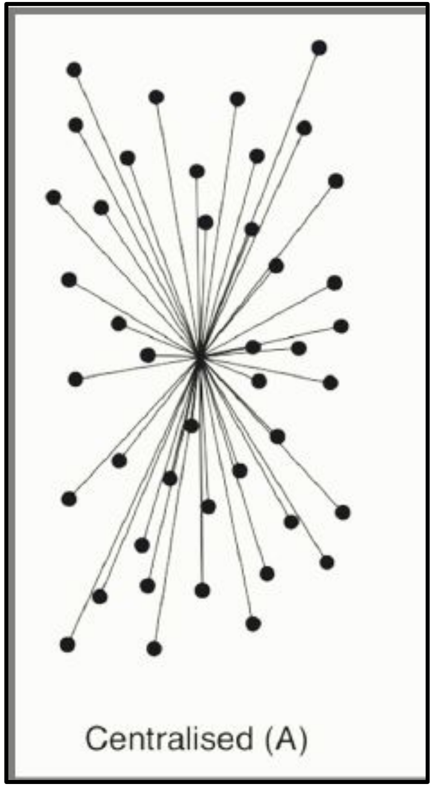


Networks: a collection of connected nodes

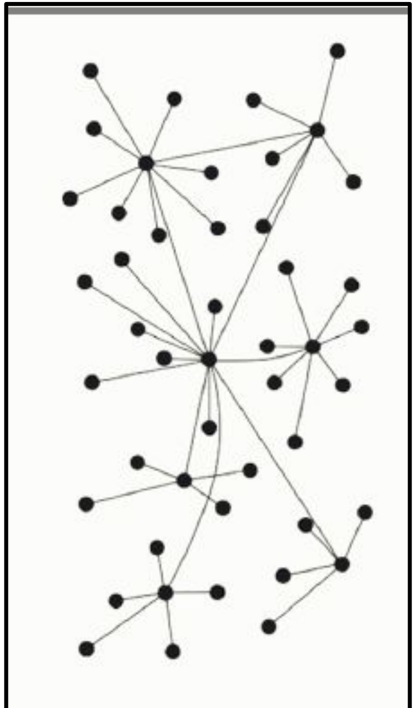


Centralized (vs. Decentralized vs. Distributed)

Centralized (vs. Decentralized vs. Distributed)

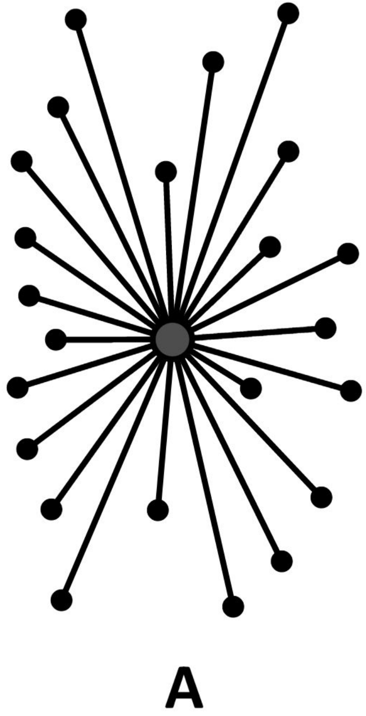


Centralized vs. Decentralized (vs. Distributed)

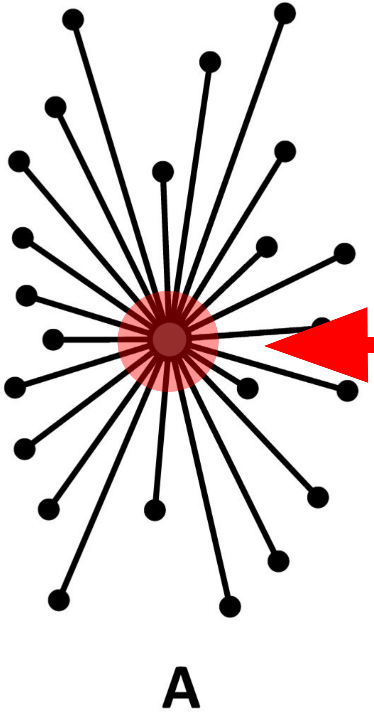


Decentralised (B)

Centralized vs. Decentralized Networks

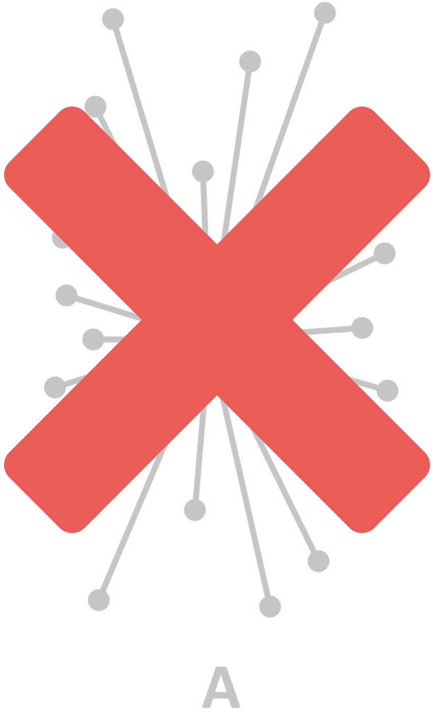


Centralized vs. Decentralized Networks

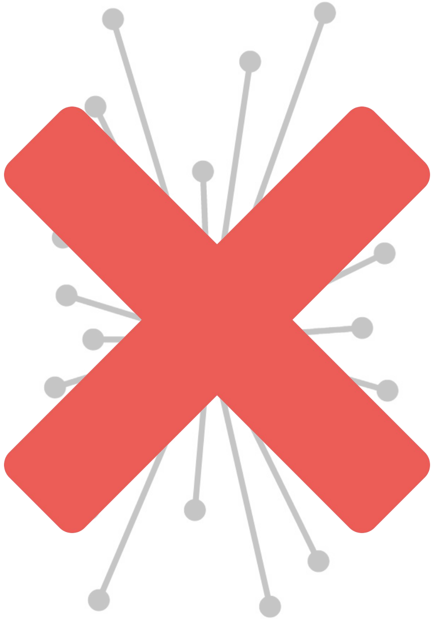


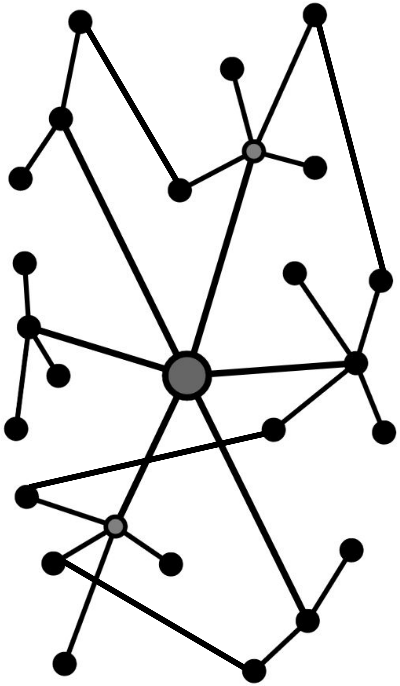
If this node is compromised, the <u>whole</u> network goes down!

Centralized vs. Decentralized Networks

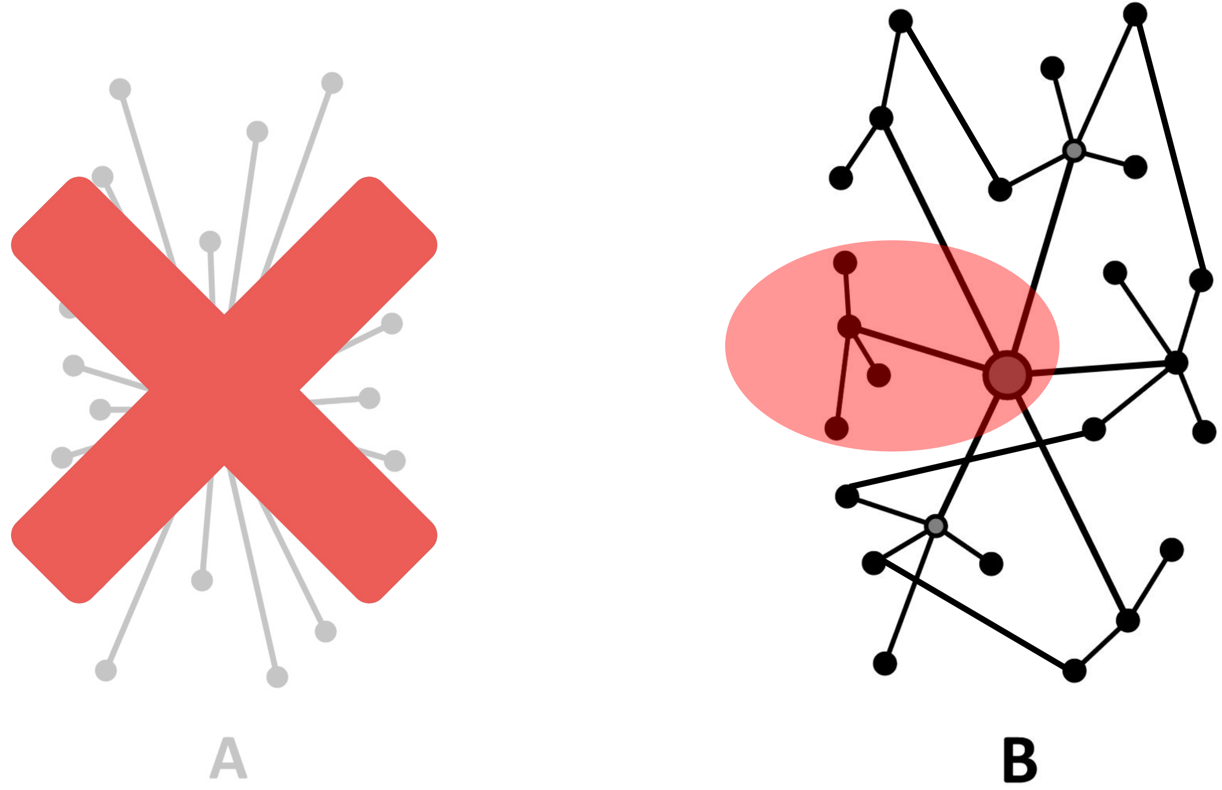


Decentralized Networks


A

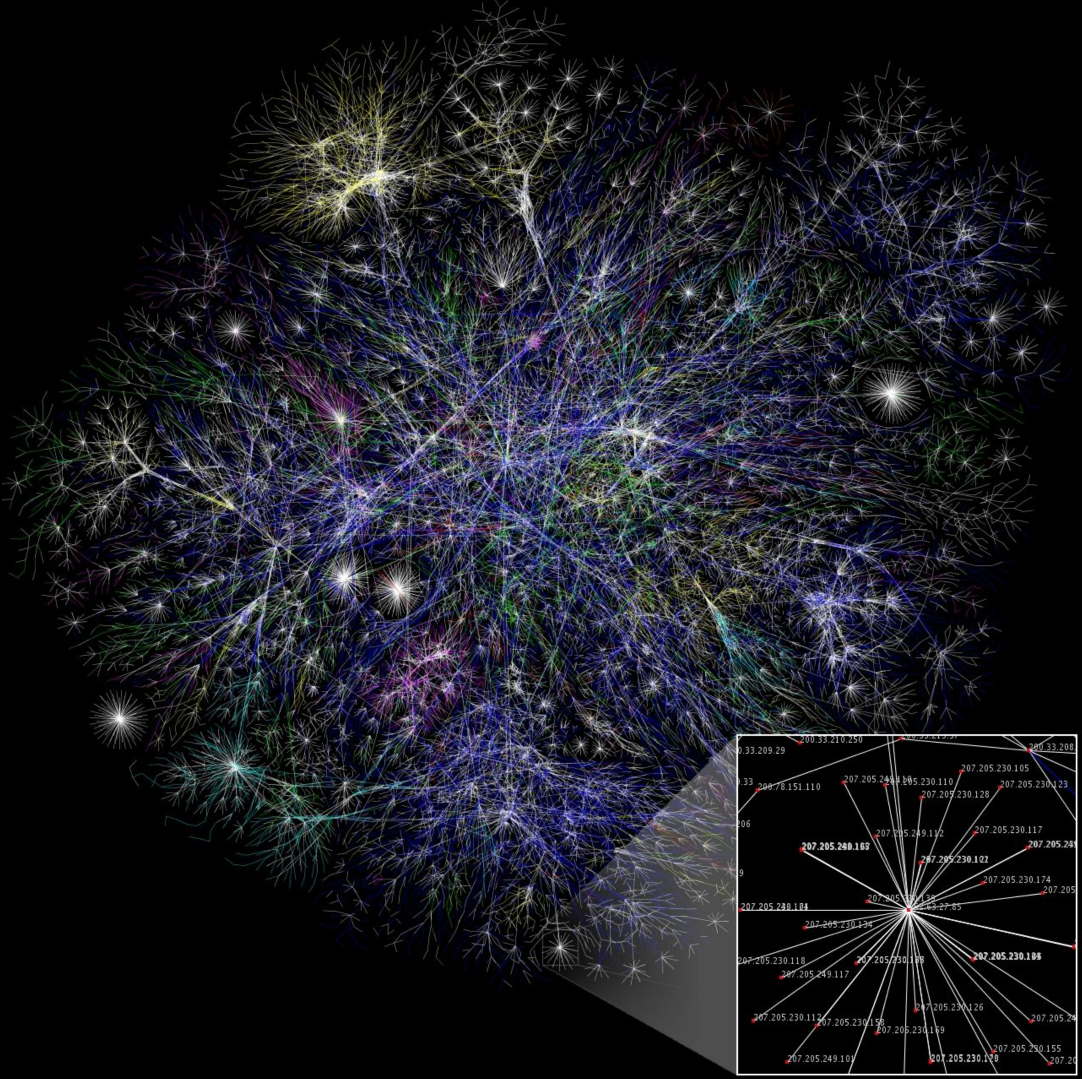

B

Decentralized Networks



Real Decentralized Technologies

Internet



200.33.210.250
0.33.209.29
208.78.151.110
207.205.230.105
207.205.230.110
207.205.230.123
207.205.230.128
207.205.249.112
207.205.230.117
207.205.230.174
207.205.230.134
207.205.230.118
207.205.249.117
207.205.230.112
207.205.230.126
207.205.230.169
207.205.230.155
207.205.249.101

Real Decentralized Technologies

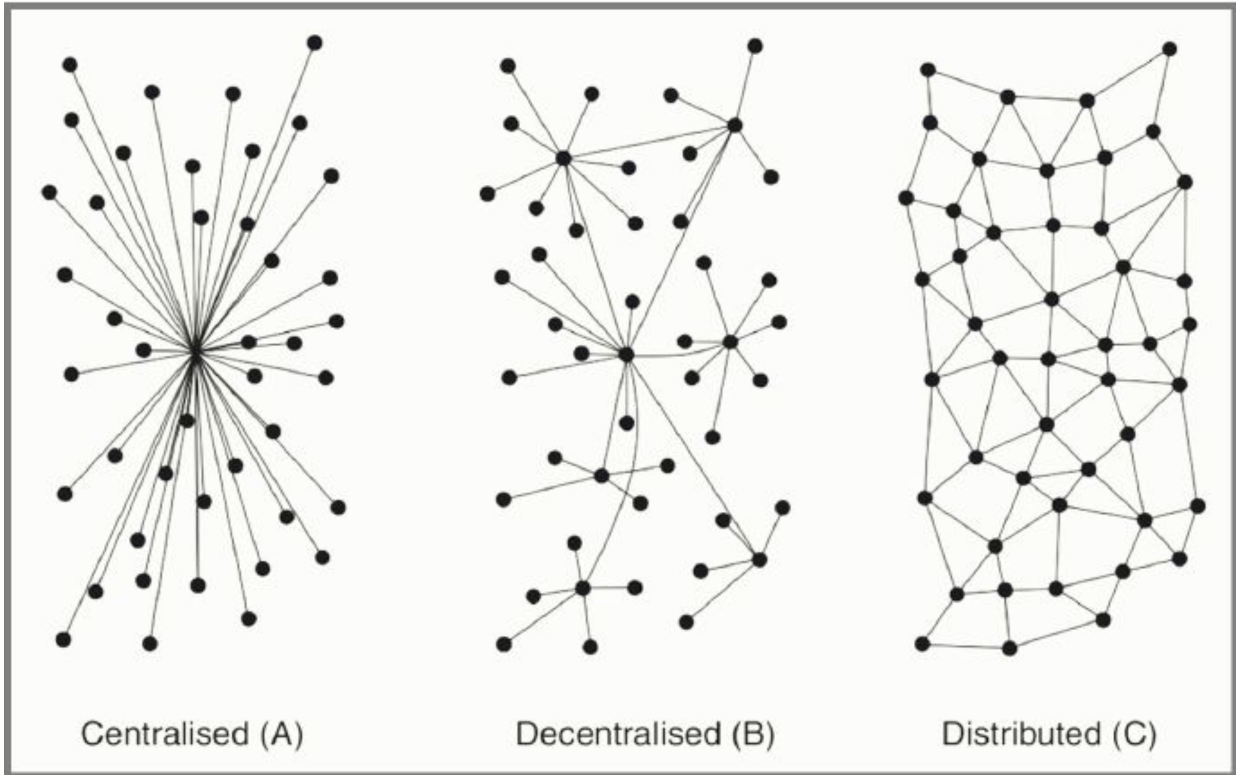
Internet

Real Decentralized Technologies

Internet

Bitcoin

Centralized vs. Decentralized vs. Distributed



A few words on how internet works as a network ...

A few words on Internet

- Billions of connected (computing) hosts/end-systems - mobile devices now outnumber others by a large margin

A few words on Internet

- Billions of connected (computing) hosts/end-systems (mobile devices now outnumber others by a large margin)
  - laptops
  - smartphones, tablets
  - TVs
  - Gaming consoles
  - Webcams
  - Automobiles,
  - Environmental sensing devices,
  - Picture frames
  - Home electrical
  - Security systems
  - And more …

A few words on Internet

- Billions of connected (computing) hosts/end-systems - mobile devices now outnumber others by a large margin
  - laptops, smartphones, tablets, TVs, gaming consoles, Webcams, automobiles, environmental sensing devices, picture frames, and home electrical, security systems, …
- Other constituents of the network (mobile, enterprise, home, ISPs, etc.):
  - Servers
  - Routers
  - Link-layer Switches
  - Modems
  - Base Stations
  - Cell Towers
  - And more …

A few words on Internet

- Billions of connected (computing) hosts/end-systems + other constituents (mobile devices now outnumber others by a large margin)
- These devices and hosts/end-systems run network apps
- They are all connected via communication links (fiber, copper, radio, satellite, etc.) and packet switches with various transmission rates (i.e. bandwidth)
- Packet Switches such as routers and switches send around and forward data packets (i.e. chunks of data) throughout the network
- In essence, you have decentralized network of networks (e.g. ISPs) + **protocols** + internet standards

A few words on Protocols

- TCP/IP
- SMTP
- IMAP
- POP
- FTP
- HTTP
- HTTPS/TLS
- UDP
- WLAN
- DNS .... and many more!

A few words on Protocols

Internet protocol suite

Application layer

BGP · DHCP · DNS · FTP · HTTP · IMAP · LDAP · MGCP · NNTP · NTP · POP · ONC/RPC · RTP · RTSP · RIP · SIP · SMTP · SNMP · SSH · Telnet · TLS/SSL · XMPP · *more...*

Transport layer

TCP · UDP · DCCP · SCTP · RSVP · *more...*

Internet layer

IP (IPv4 · IPv6) · ICMP · ICMPv6 · ECN · IGMP · IPsec · *more...*

Link layer

ARP · NDP · OSPF · Tunnels (L2TP) · PPP · MAC (Ethernet · DSL · ISDN · FDDI) · *more...*

V · T · E

“The Unsung Heros”

"The Unsung Heros"

The New York Times

Daniel Kaminsky, Internet Security Savior, Dies at 42

If you are reading this obituary online, you owe your digital safety to him.

218

Daniel Kaminsky, at his Brooklyn office in 2010, was widely hailed after finding a serious flaw in the internet's basic plumbing. Chester Higgins Jr./The New York Times

All Communication needs protocols!

All Communication needs protocols!

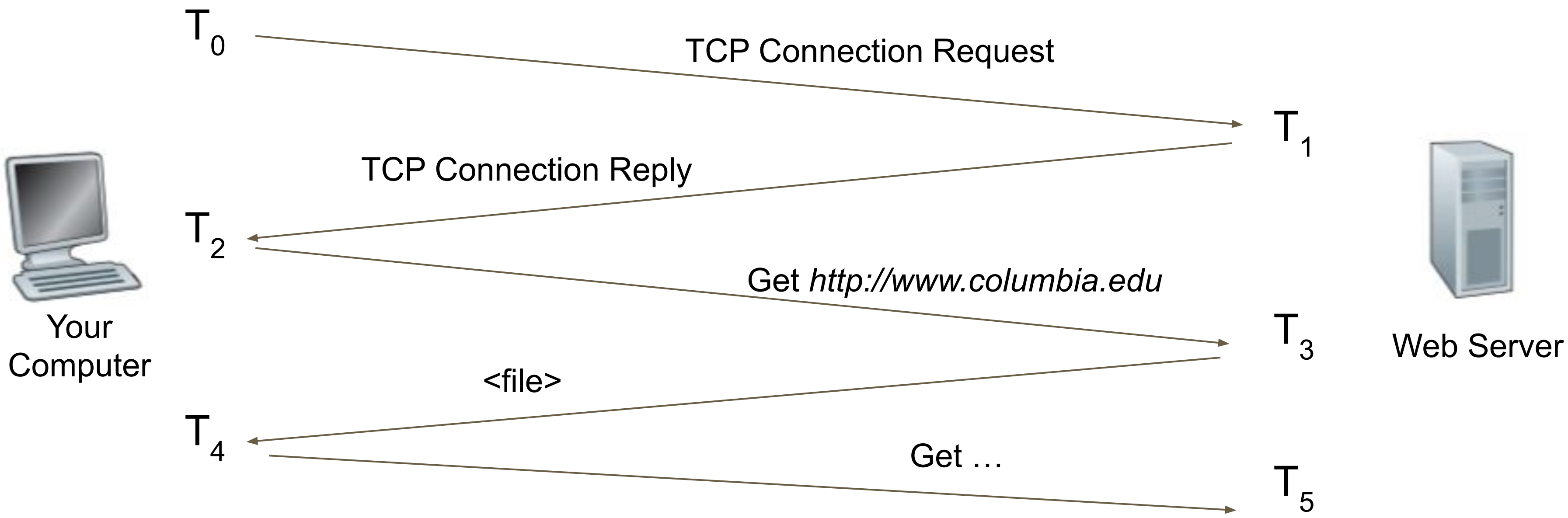
All Communication needs protocols!

All Communication needs protocols!

A few (more) words on Protocols

- Protocols are standardized methods that facilitate communication between and across different “things,” creating a common framework
- In short, Protocols define how **data should be “packetized,” addressed, transmitted, routed, and received** → examples to follow
- Let’s use the example of exchanging messages: first with humans (asking for time, exchanging business cards, mailing a letter), then machines - all communications are in essence governed by protocols
- Protocols help manage complexity across various building blocks of the internet (hosts, routers, switches, applications, hardware, software, etc.) … BUT … how do we **organize** them and the structure of our network?
- [by the way, it was mostly a volunteer effort, with no possibility for monetization by the makers]

Sample Computer Network Protocol (signals & msgs)

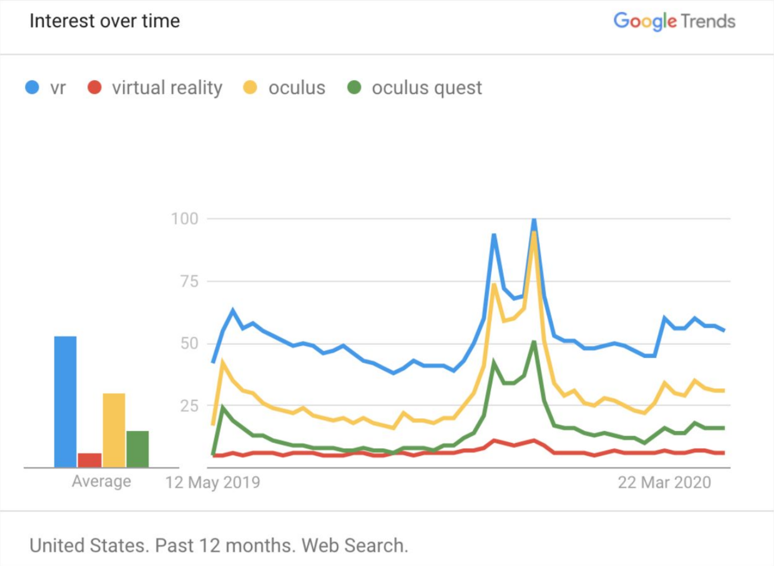


case comparison: how do you ask questions in class?

How a web page is rendered (put simplistically)

Benedict Evans   Newsletter   Essays   Presentations   Contact

Meanwhile, it's instructive that now that we're all locked up at home, video calls have become a huge consumer phenomenon, but VR has been not. This should have been a VR moment, and it isn't.

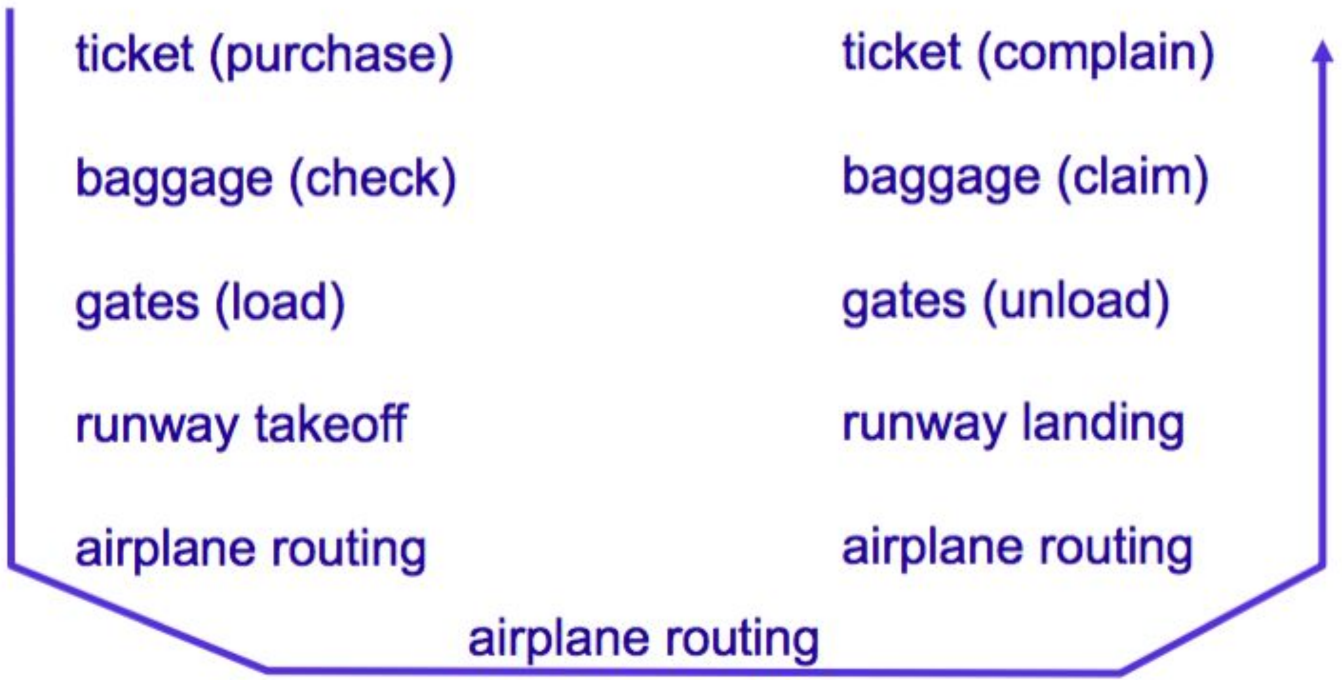


Master Definition of a Protocol

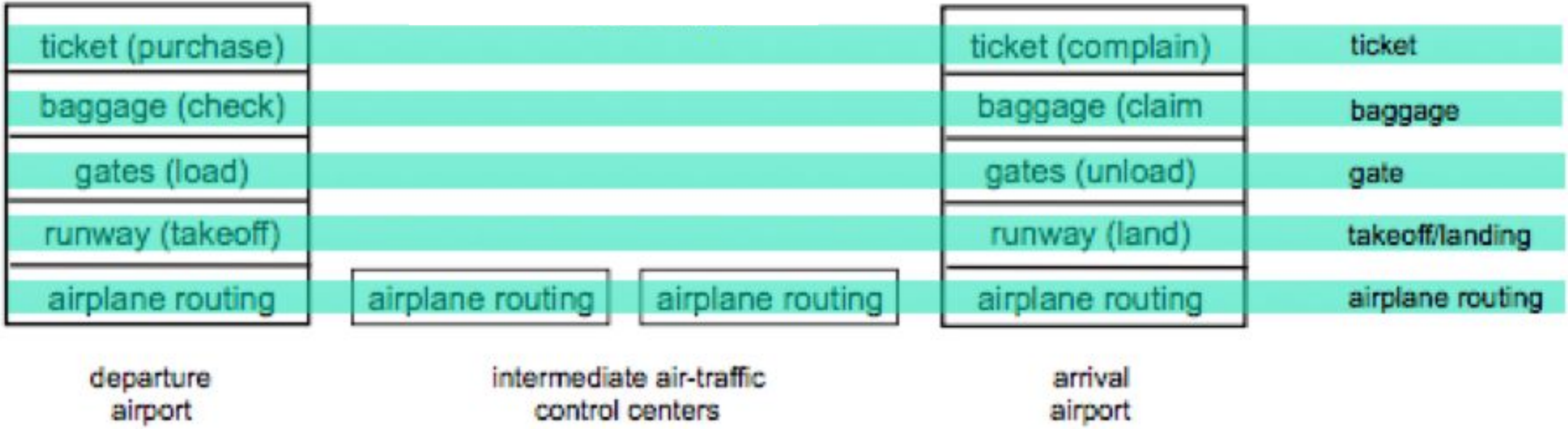
“A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.”

- *James Kurose, Keith Ross*

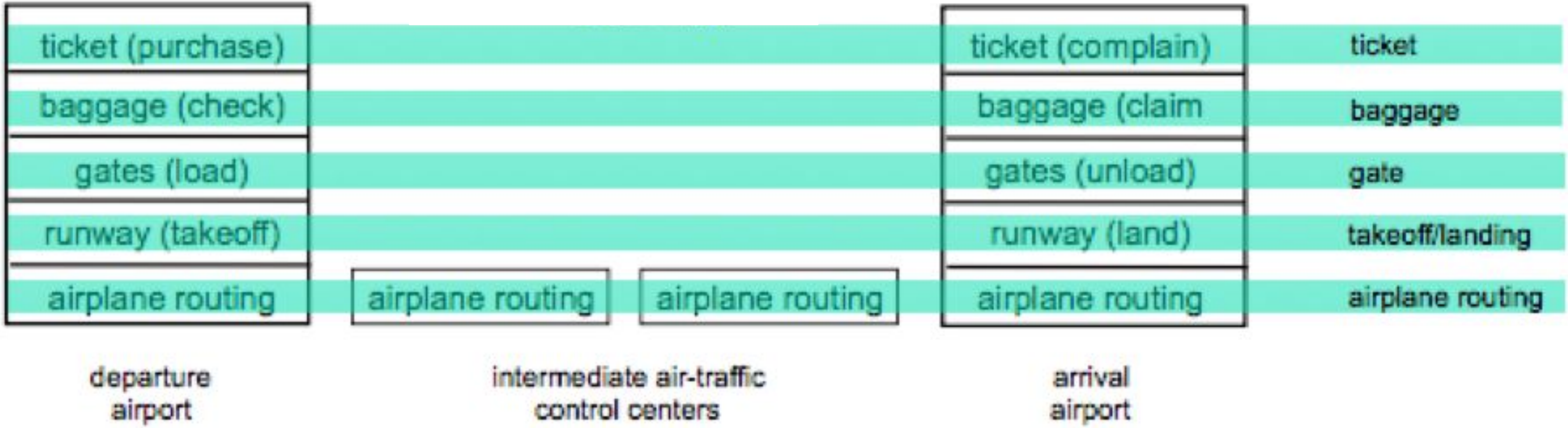
Let's organize a flight … through a series of steps



Organizing a flight … through functionality layers



Organizing a flight ... through functionality layers



- Each LAYER implements a service ... via its own internally-layer processes ... and relying on the services provided by layer below

Internet Protocol Stack

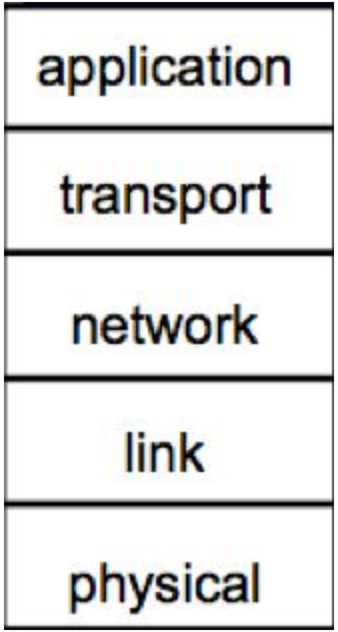
Application: support and enable end-user apps

Transport: process data transfer

Network: routing of data from source to destination

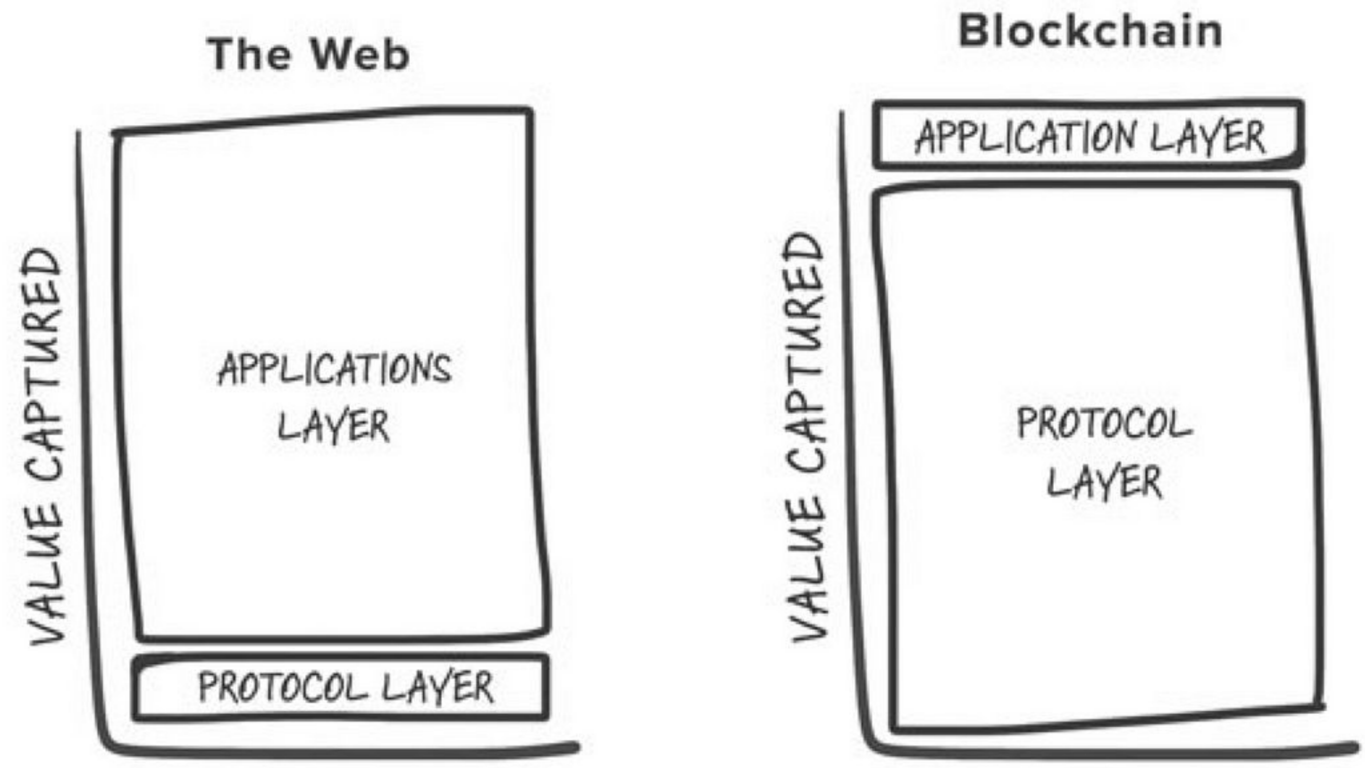
Link: data transfer between neighboring network elements (e.g. WiFi)

Physical: bits “on the wire” (hardware)

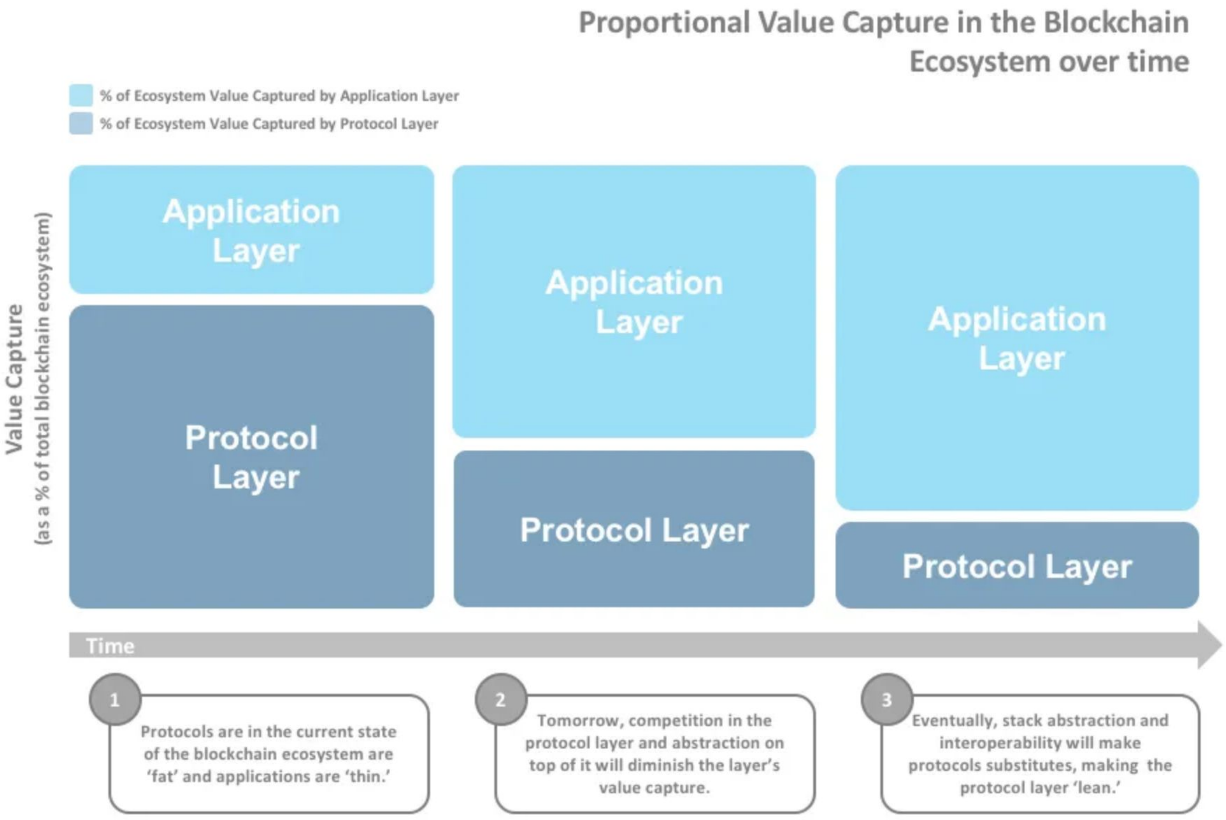


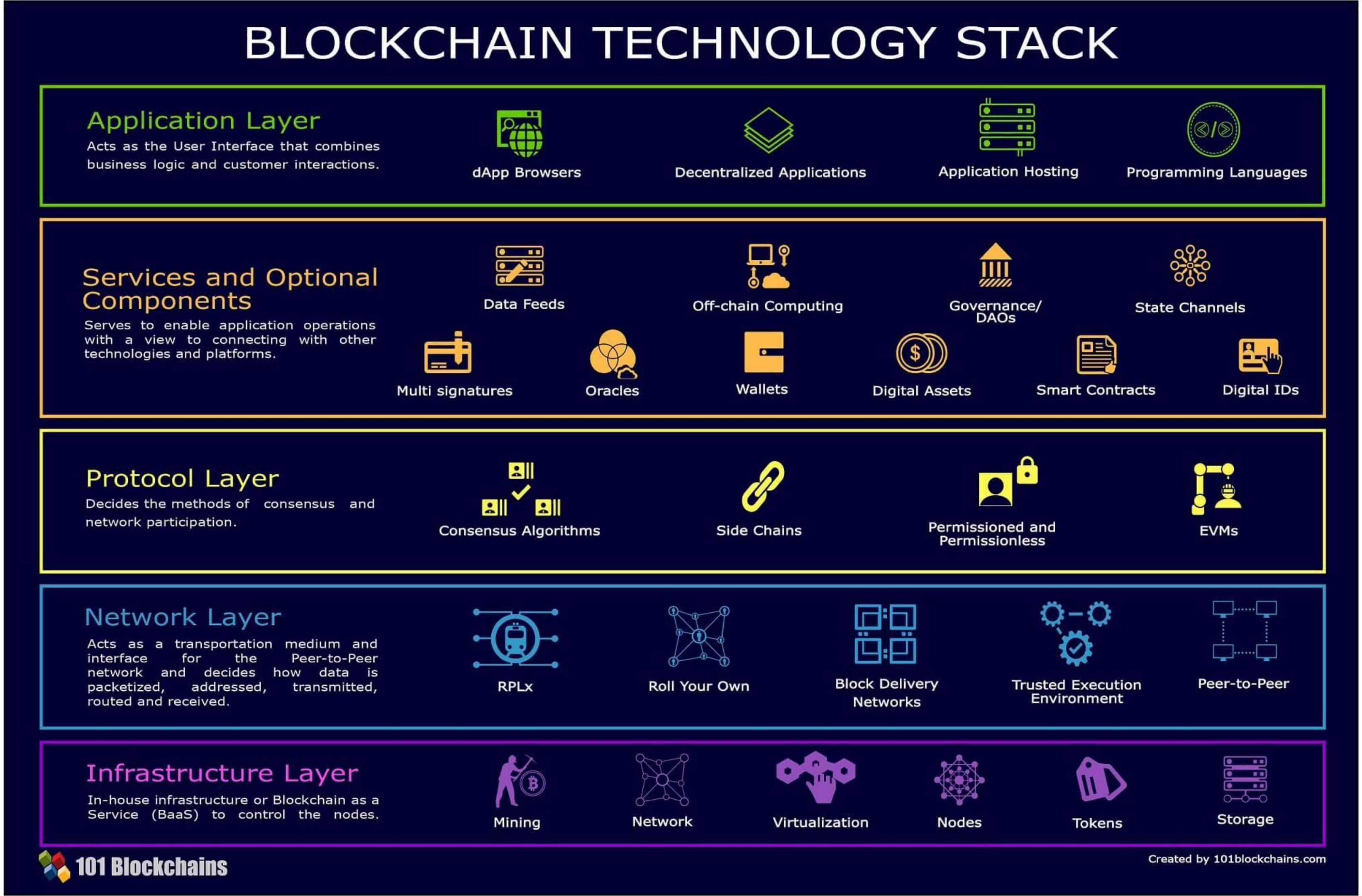
Why is all this important?!

"Fat Protocols" (by Joel Monegro, USV)



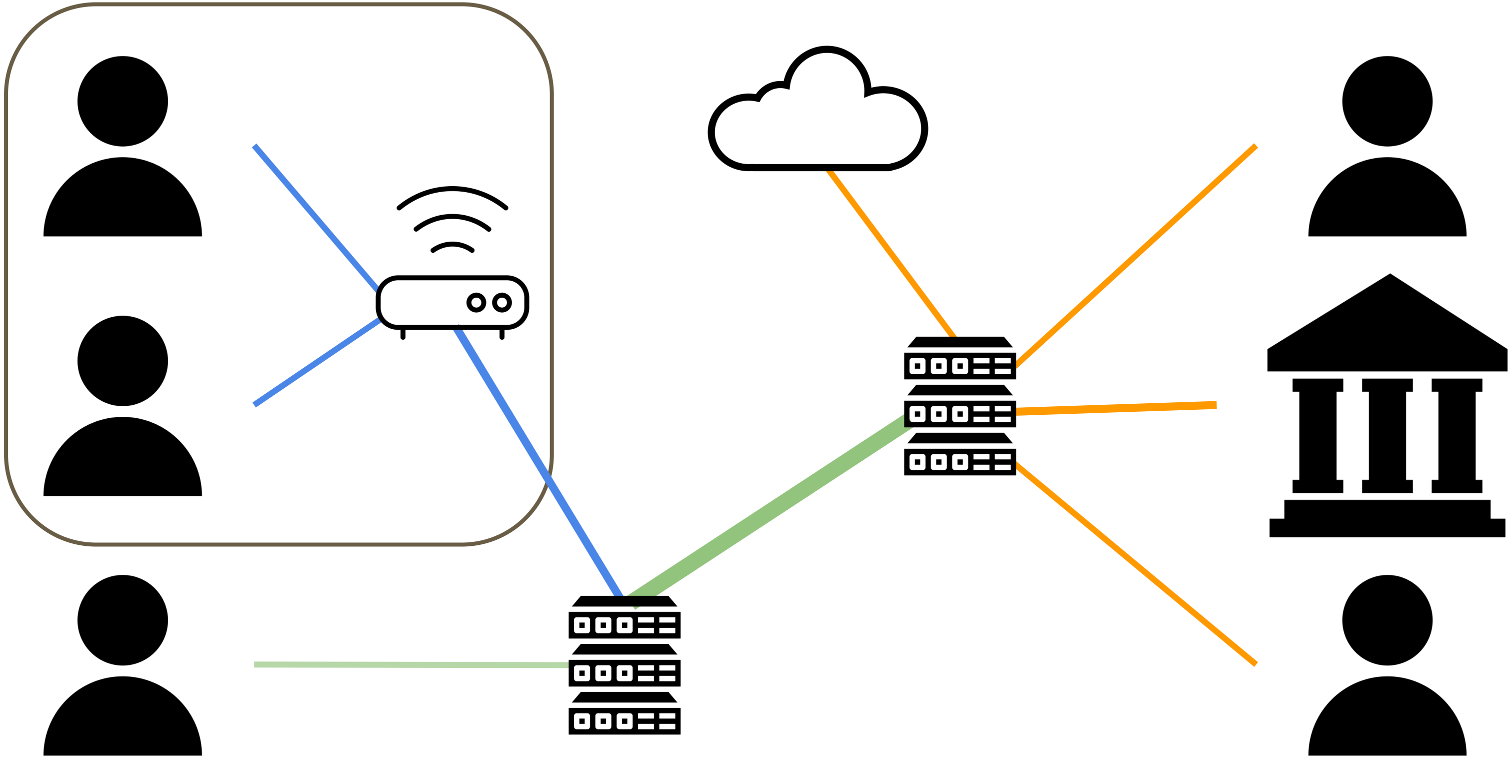
“Fat Protocols & Value Capture ” (Johnson Nakano)



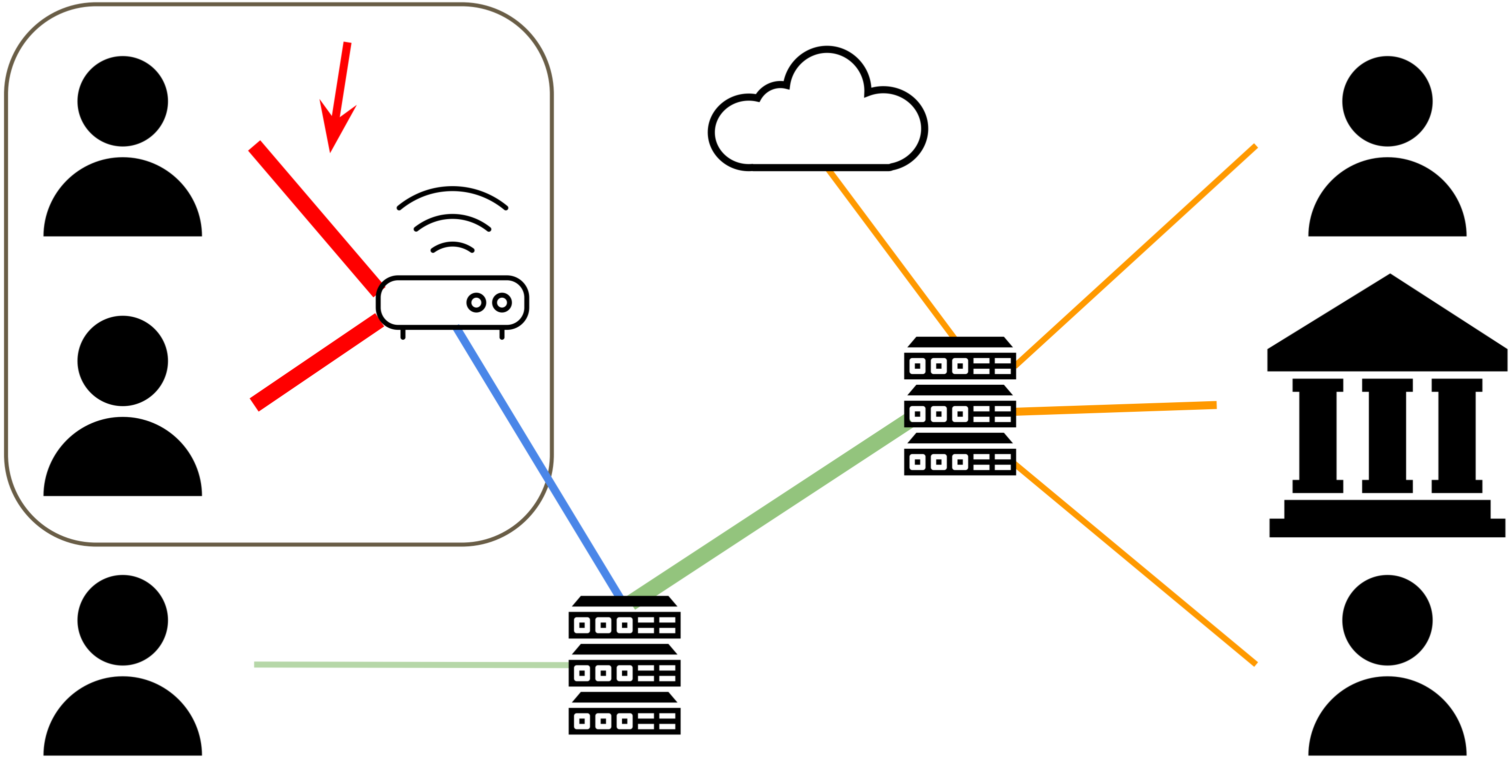

BLOCKCHAIN TECHNOLOGY STACK
Application Layer
Acts as the User Interface that combines business logic and customer interactions.
dApp Browsers
Decentralized Applications
Application Hosting
Programming Languages
Services and Optional Components
Serves to enable application operations with a view to connecting with other technologies and platforms.
Data Feeds
Off-chain Computing
Governance/ DAOs
State Channels
Multi signatures
Oracles
Wallets
Digital Assets
Smart Contracts
Digital IDs
Protocol Layer
Decides the methods of consensus and network participation.
Consensus Algorithms
Side Chains
Permissioned and Permissionless
EVMs
Network Layer
Acts as a transportation medium and interface for the Peer-to-Peer network and decides how data is packetized, addressed, transmitted, routed and received.
RPLx
Roll Your Own
Block Delivery Networks
Trusted Execution Environment
Peer-to-Peer
Infrastructure Layer
In-house infrastructure or Blockchain as a Service (BaaS) to control the nodes.
Mining
Network
Virtualization
Nodes
Tokens
Storage
101 Blockchains
Created by 101blockchains.com

Back to networks … and security

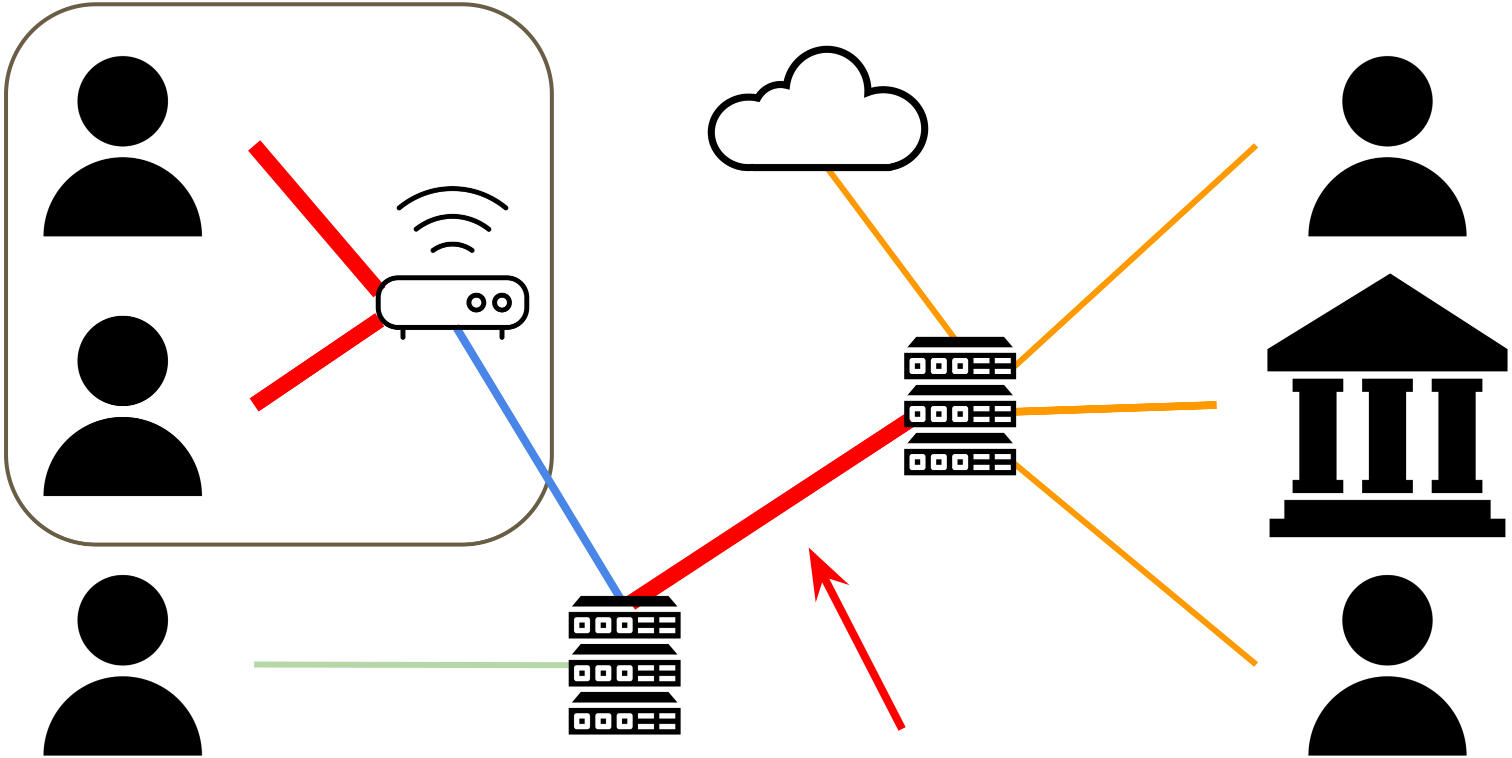
So what do we need to protect?



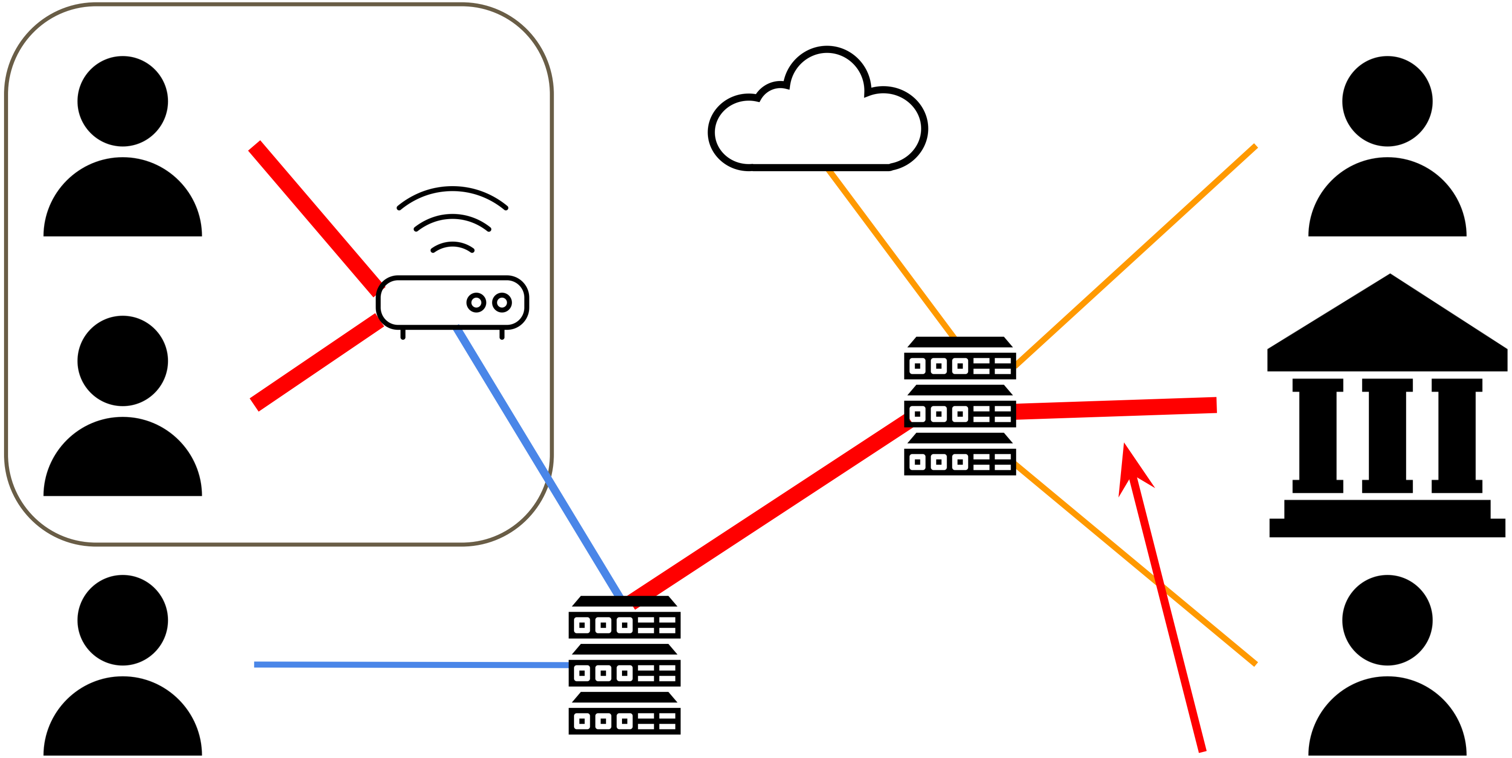
So what do we need to protect?



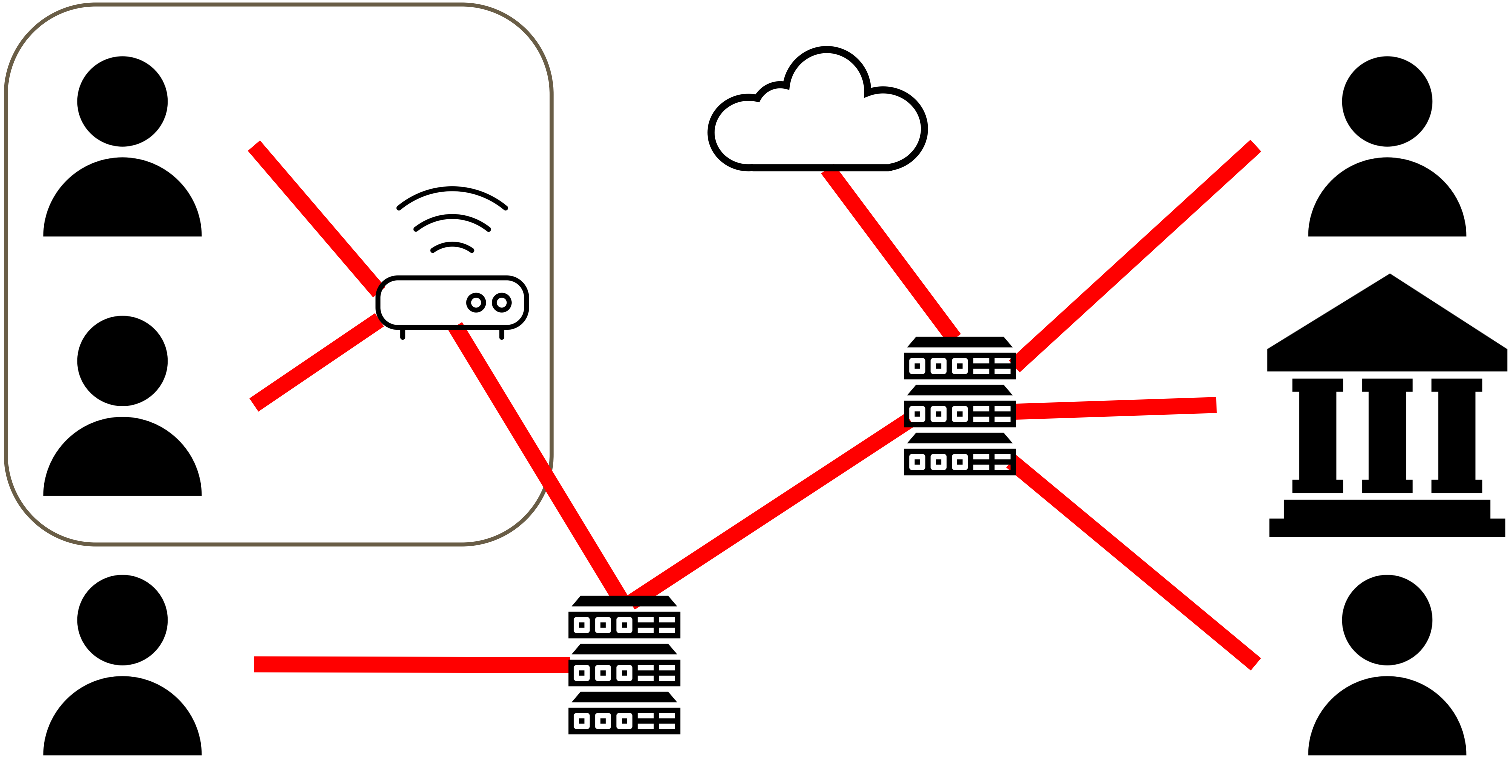
So what do we need to protect?



So what do we need to protect?



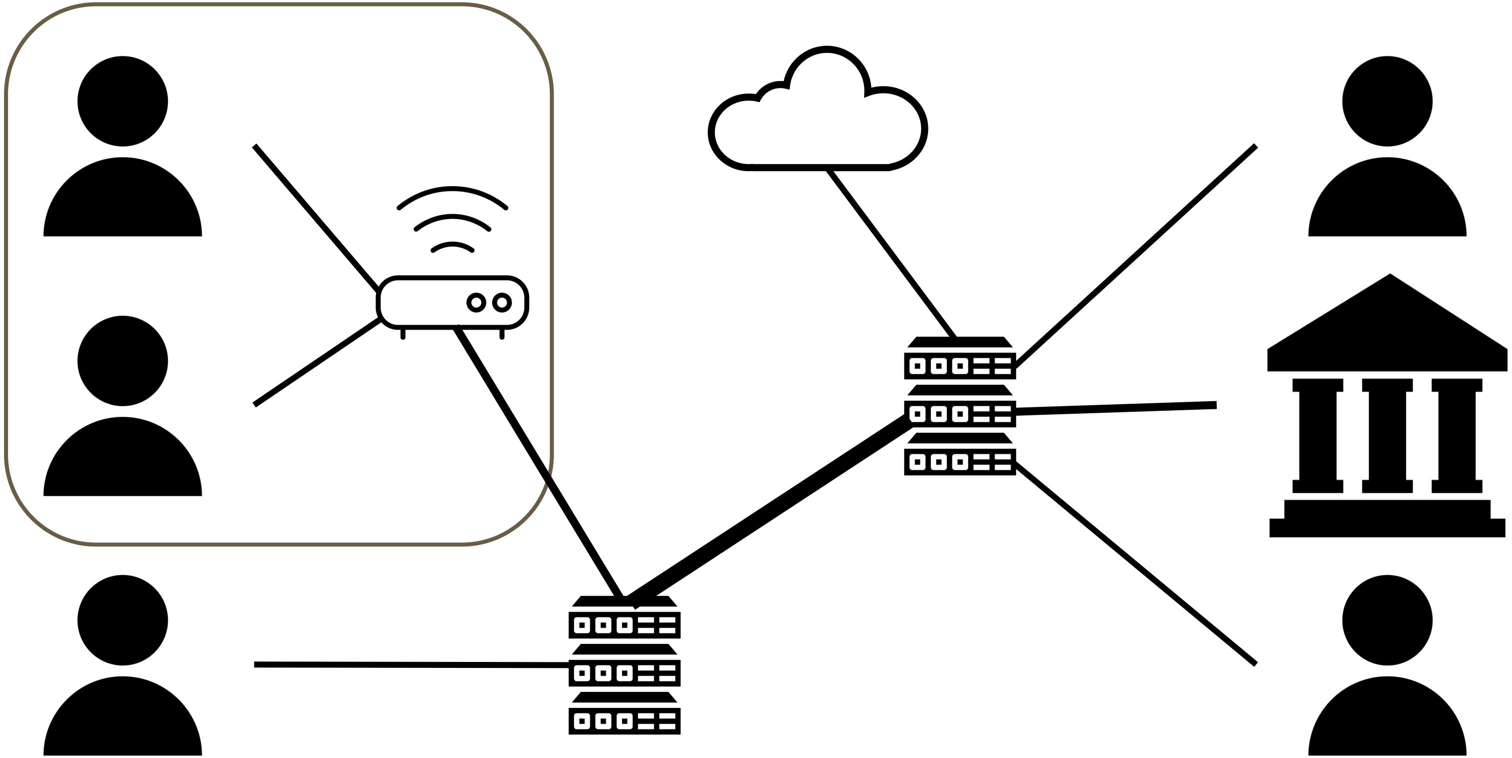
So what do we need to protect?



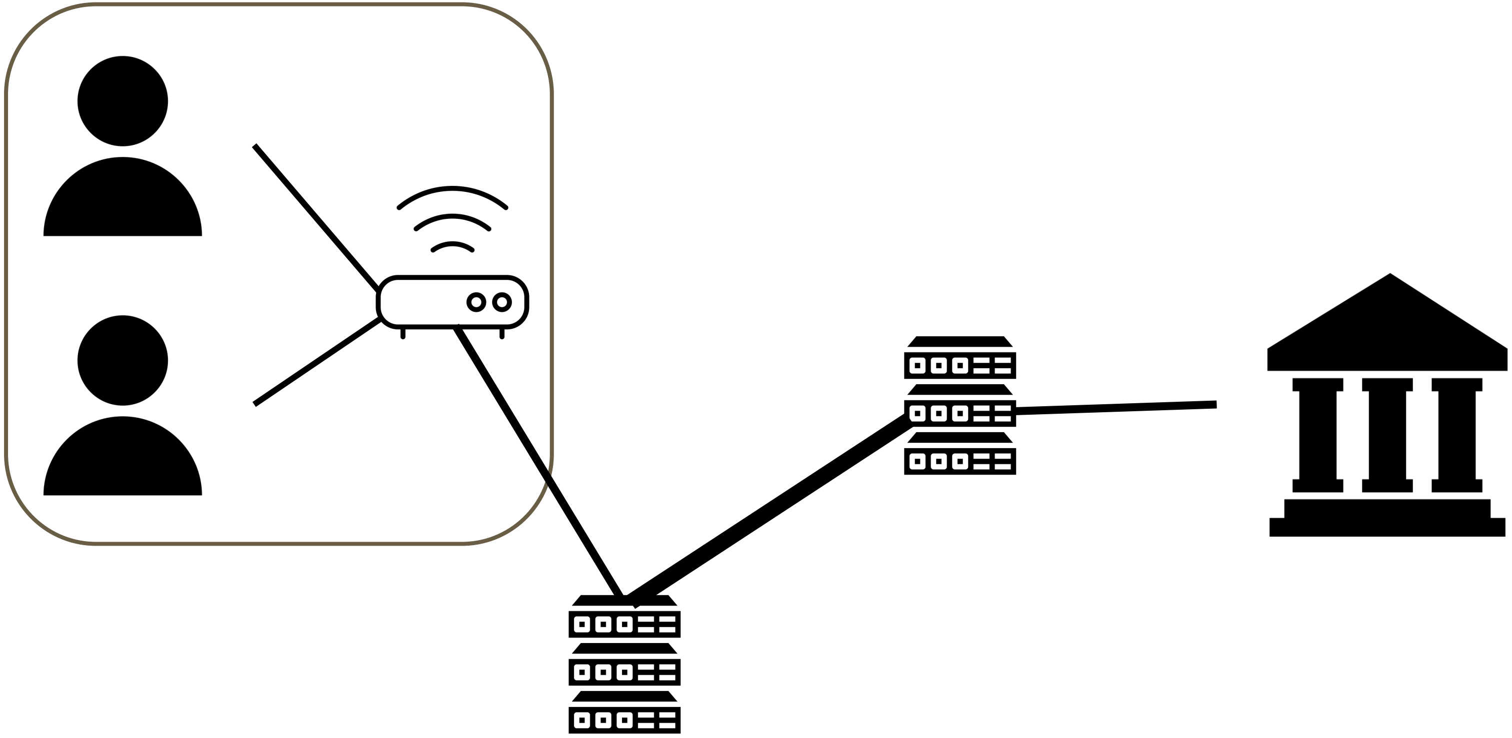
II. The Bad Guys

Types of attackers, and the cryptographic techniques we can use to circumvent them.

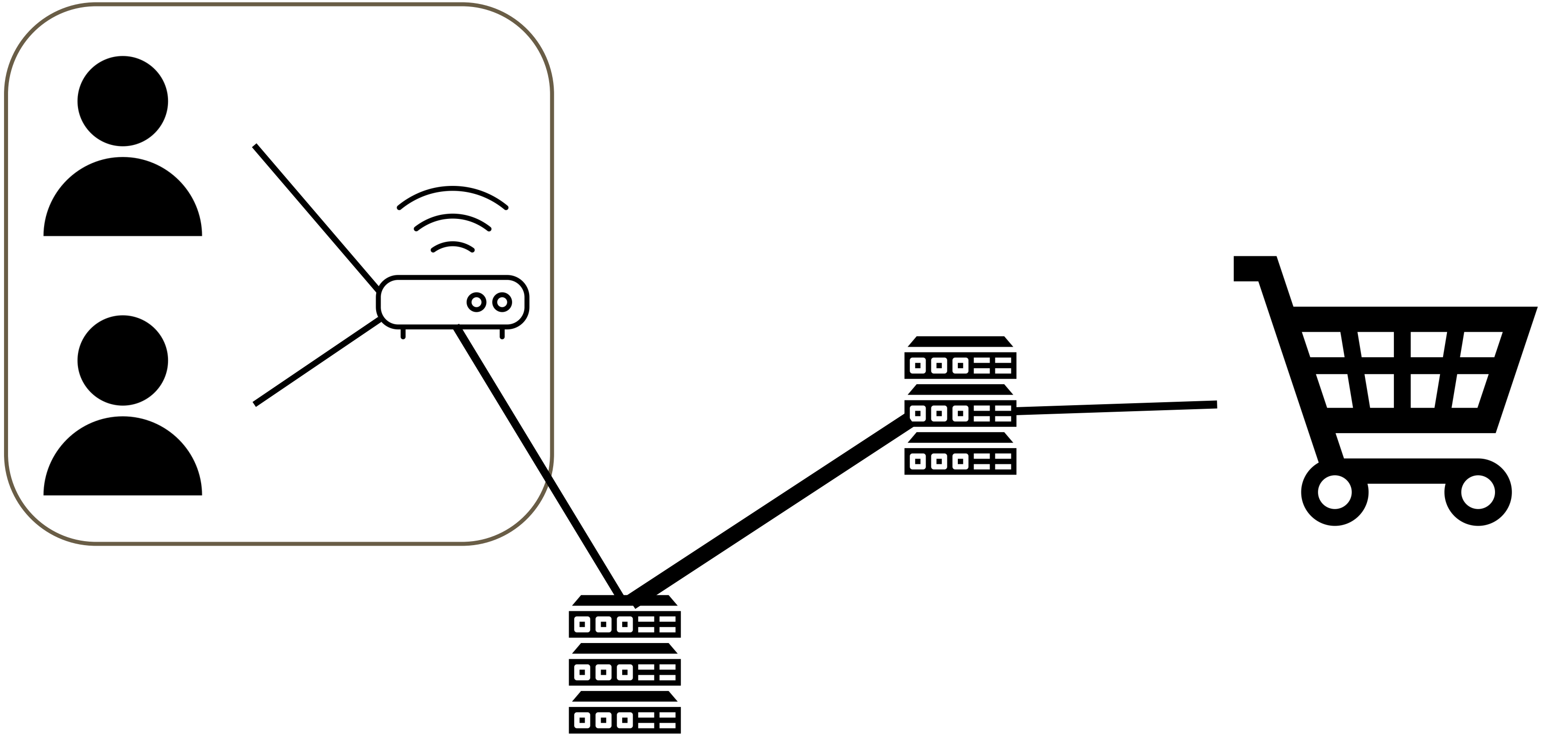
The Snooper



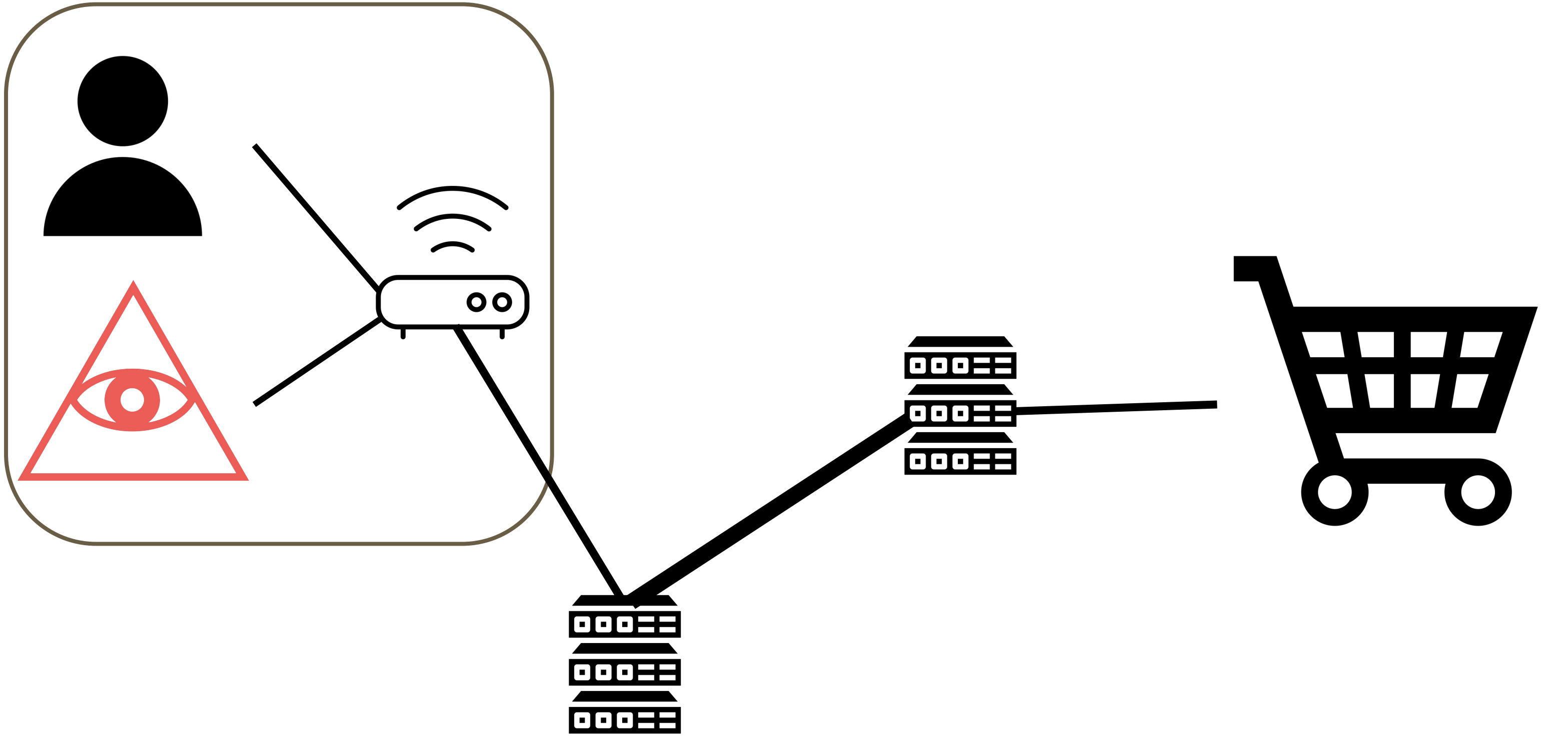
The Snooper



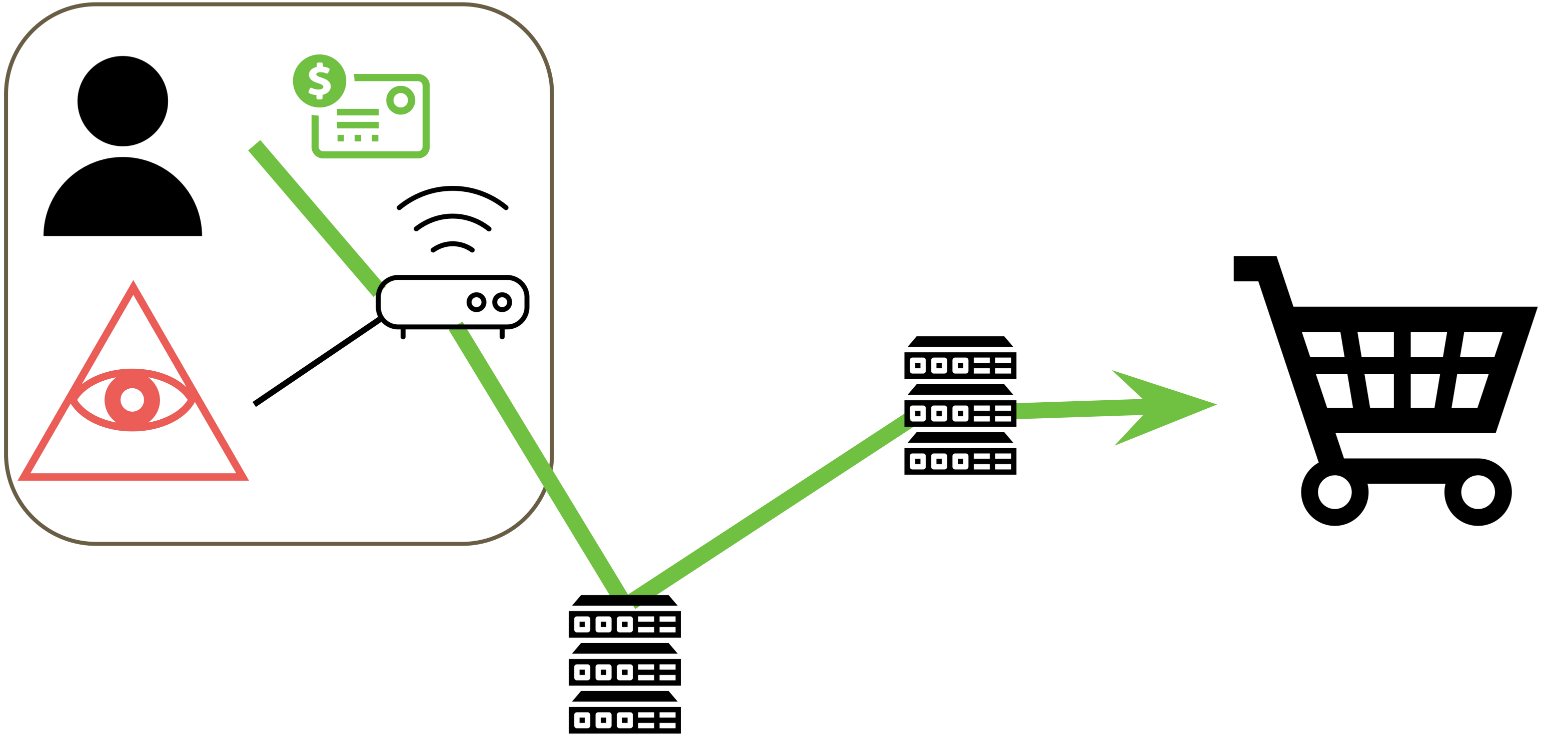
The Snooper



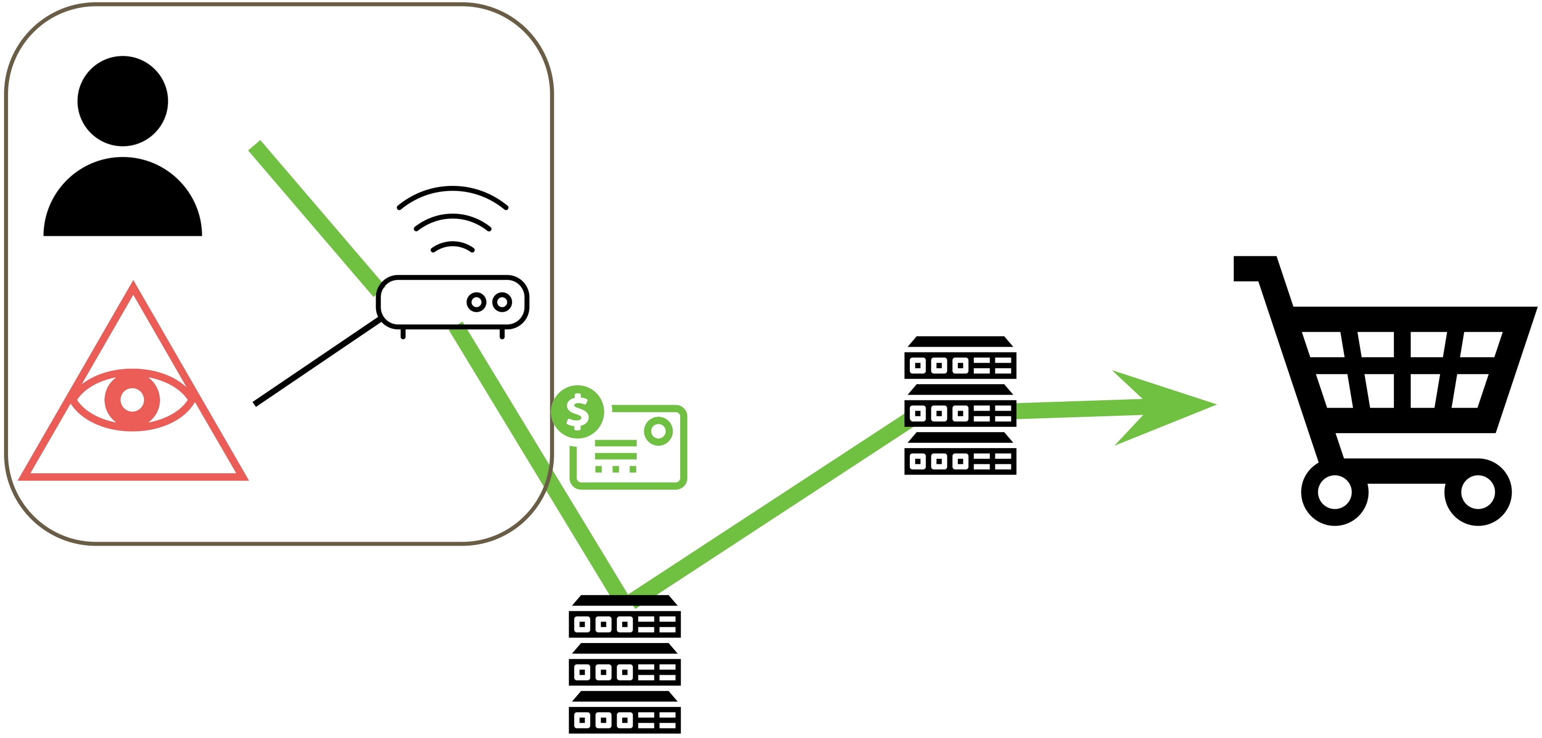
The Snooper



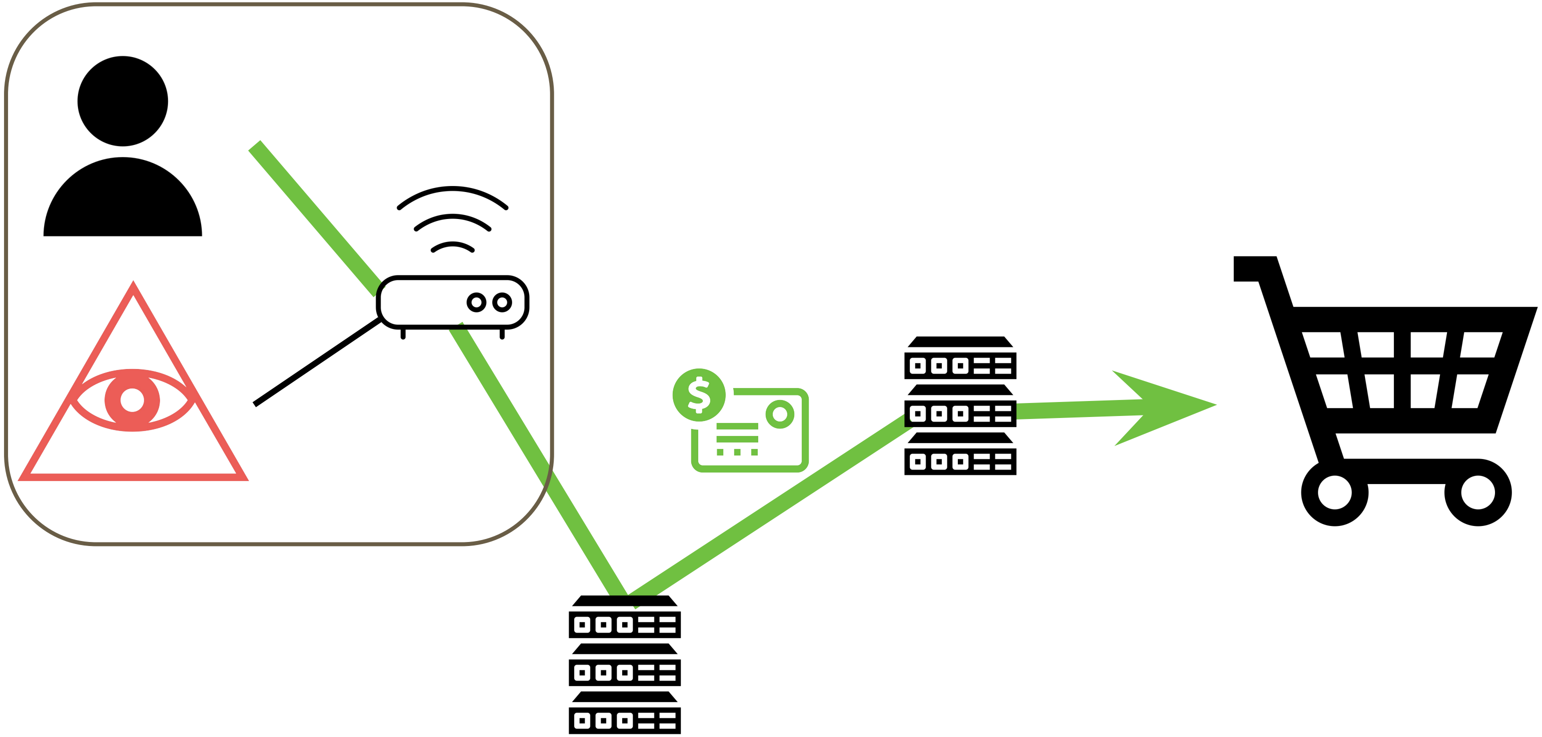
The Snooper



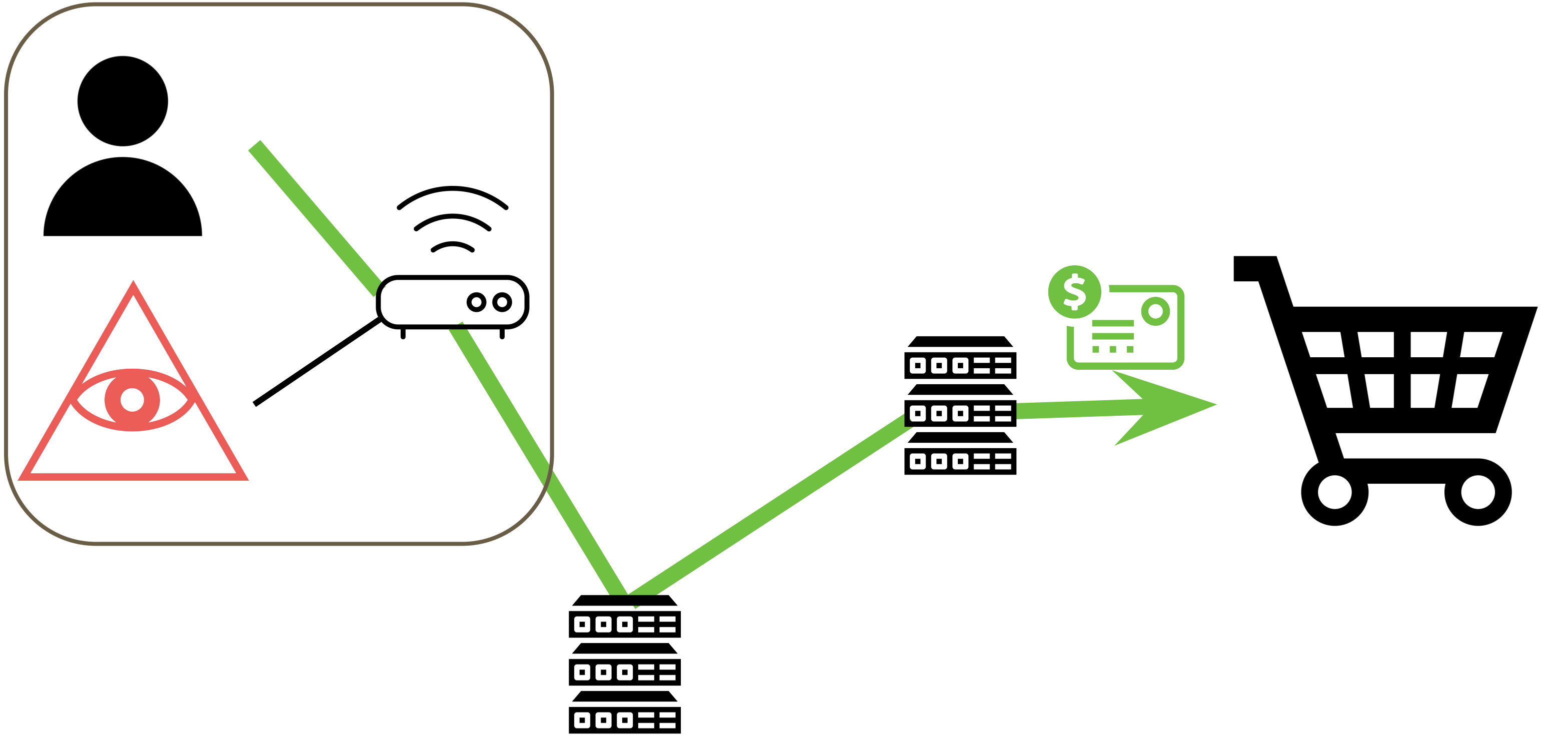
The Snooper



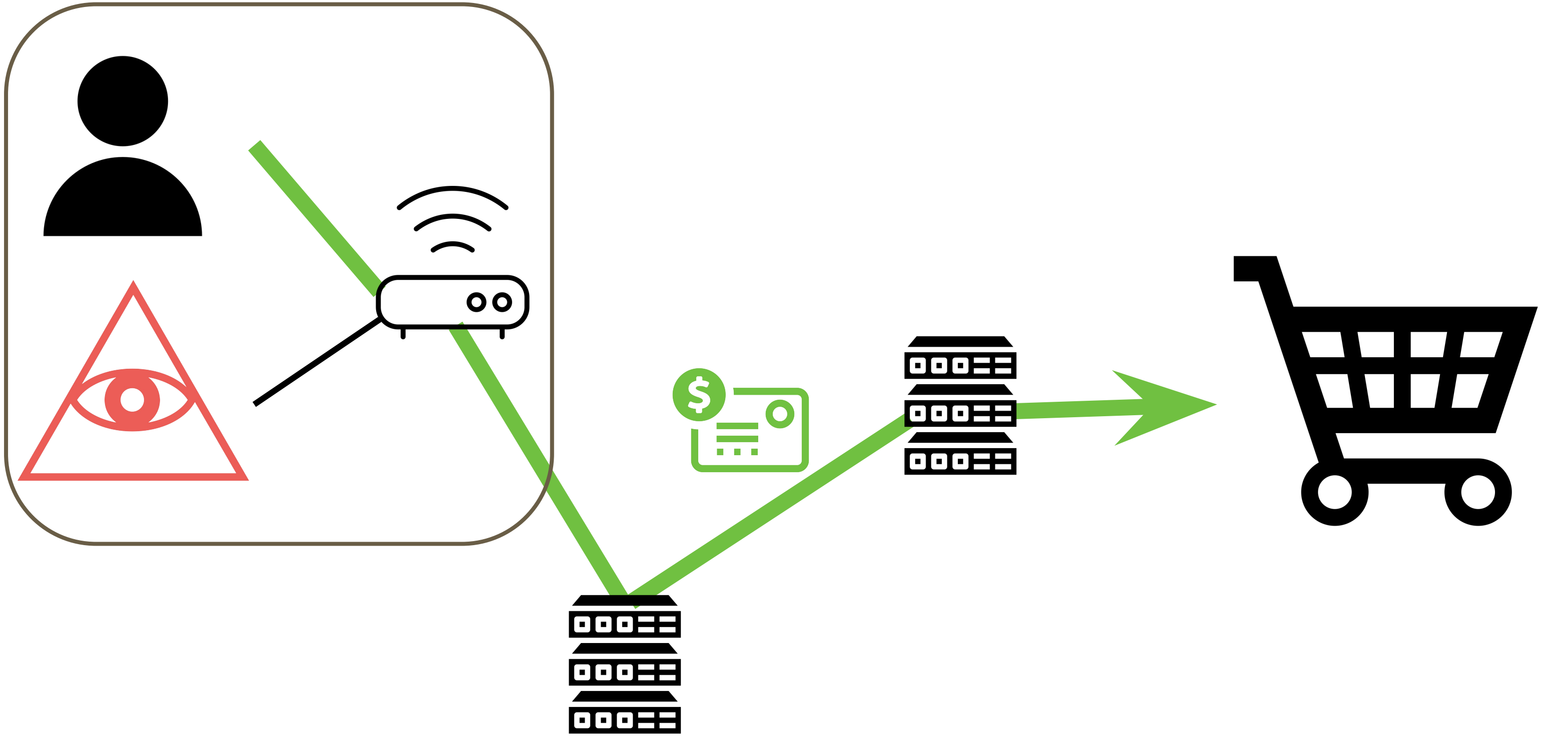
The Snooper



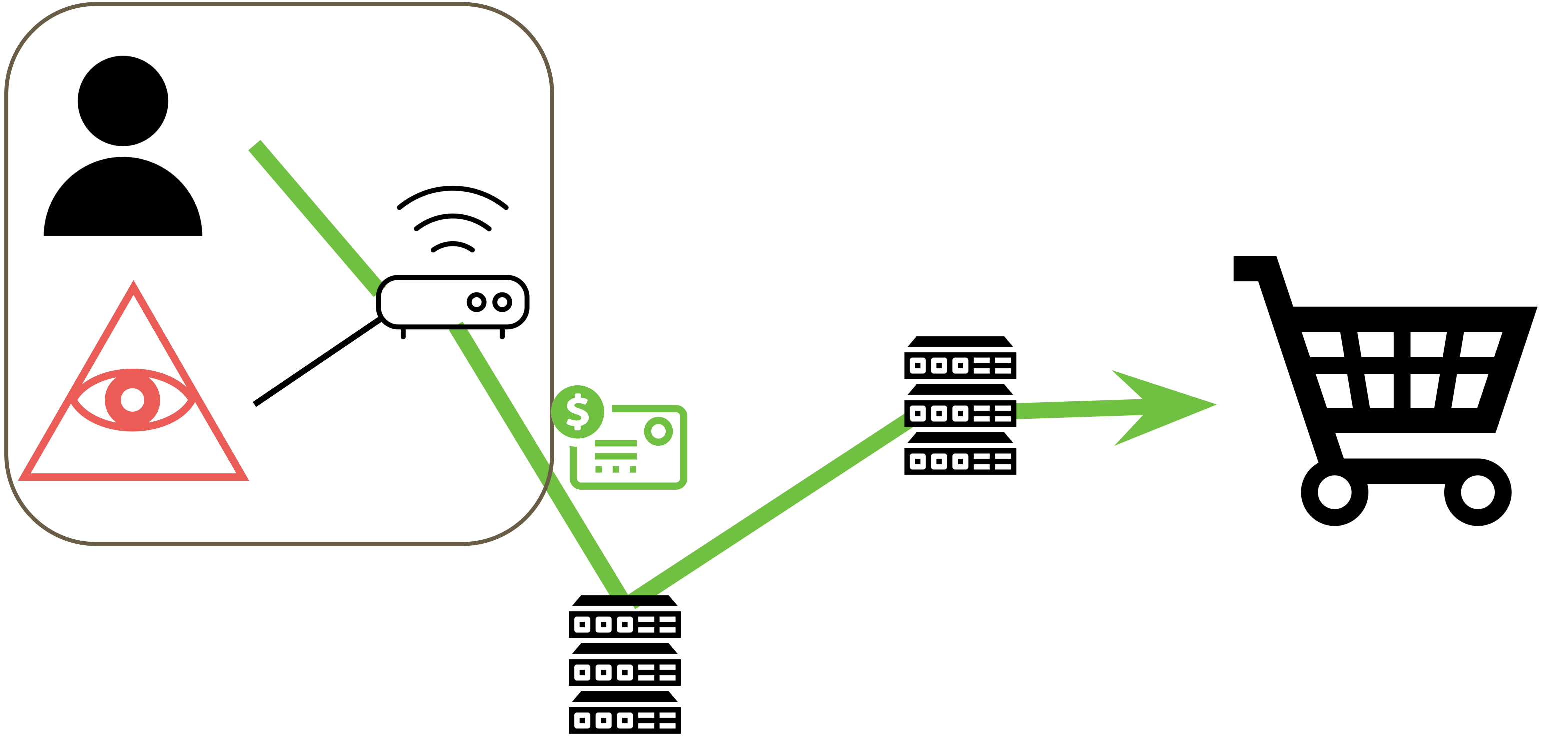
The Snooper



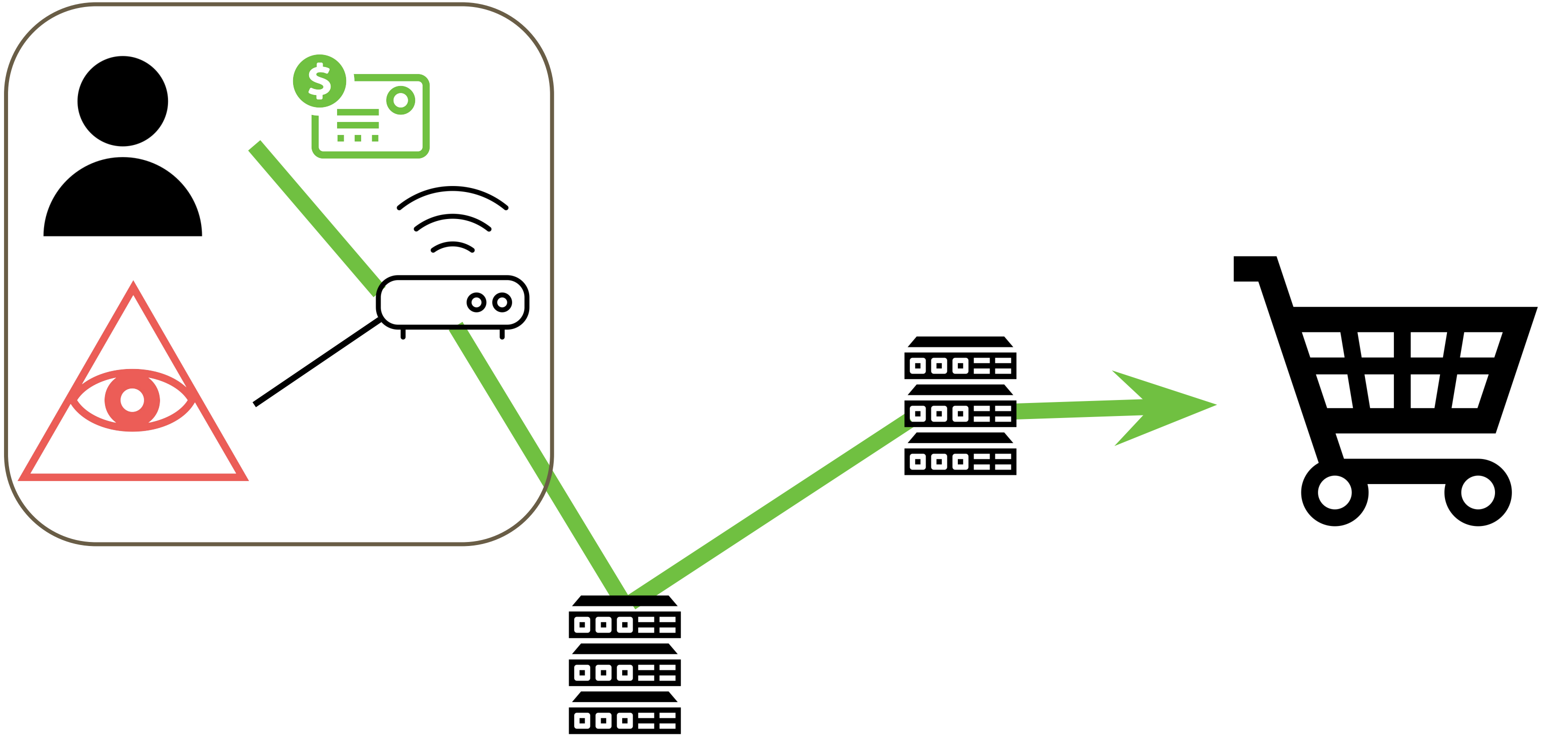
The Snooper



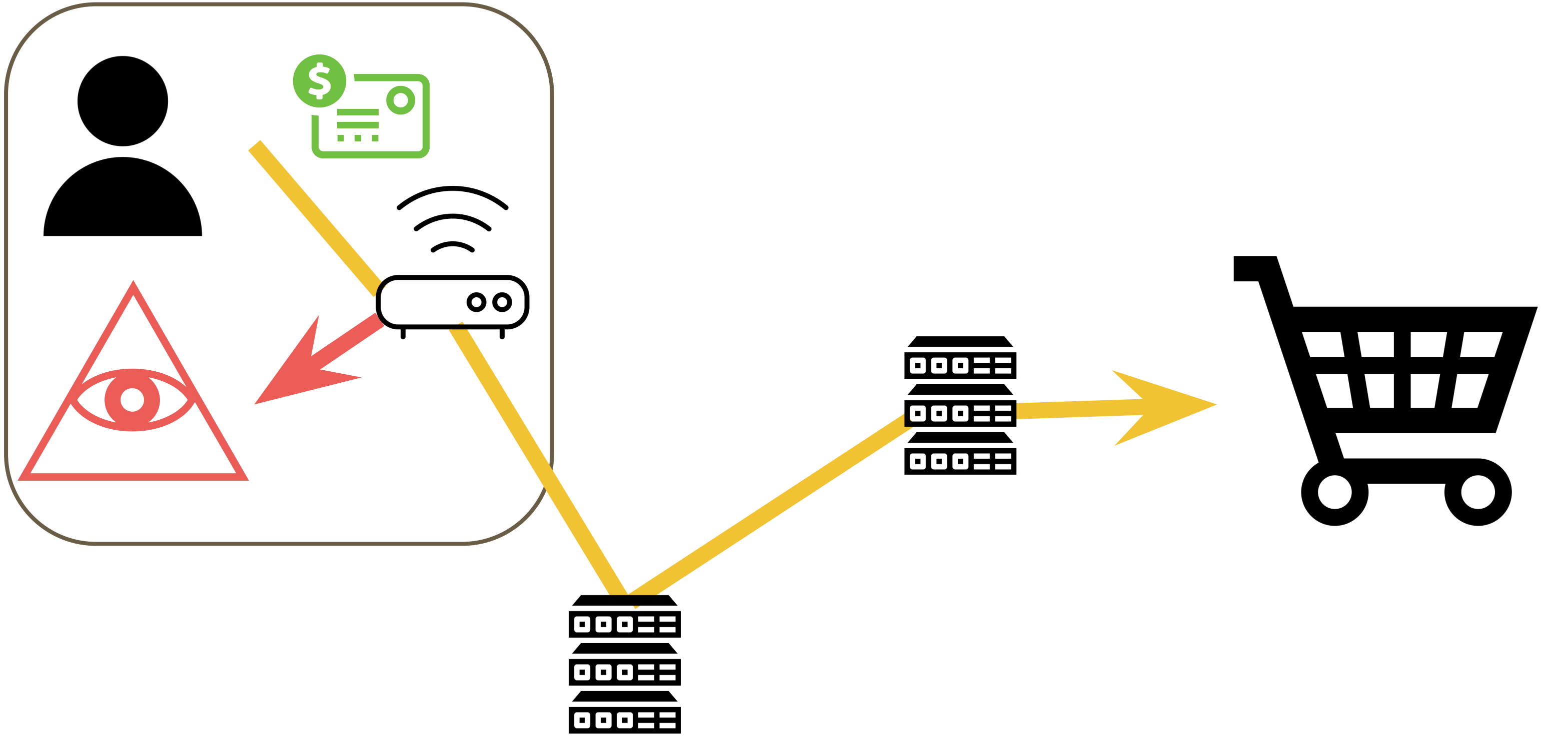
The Snooper



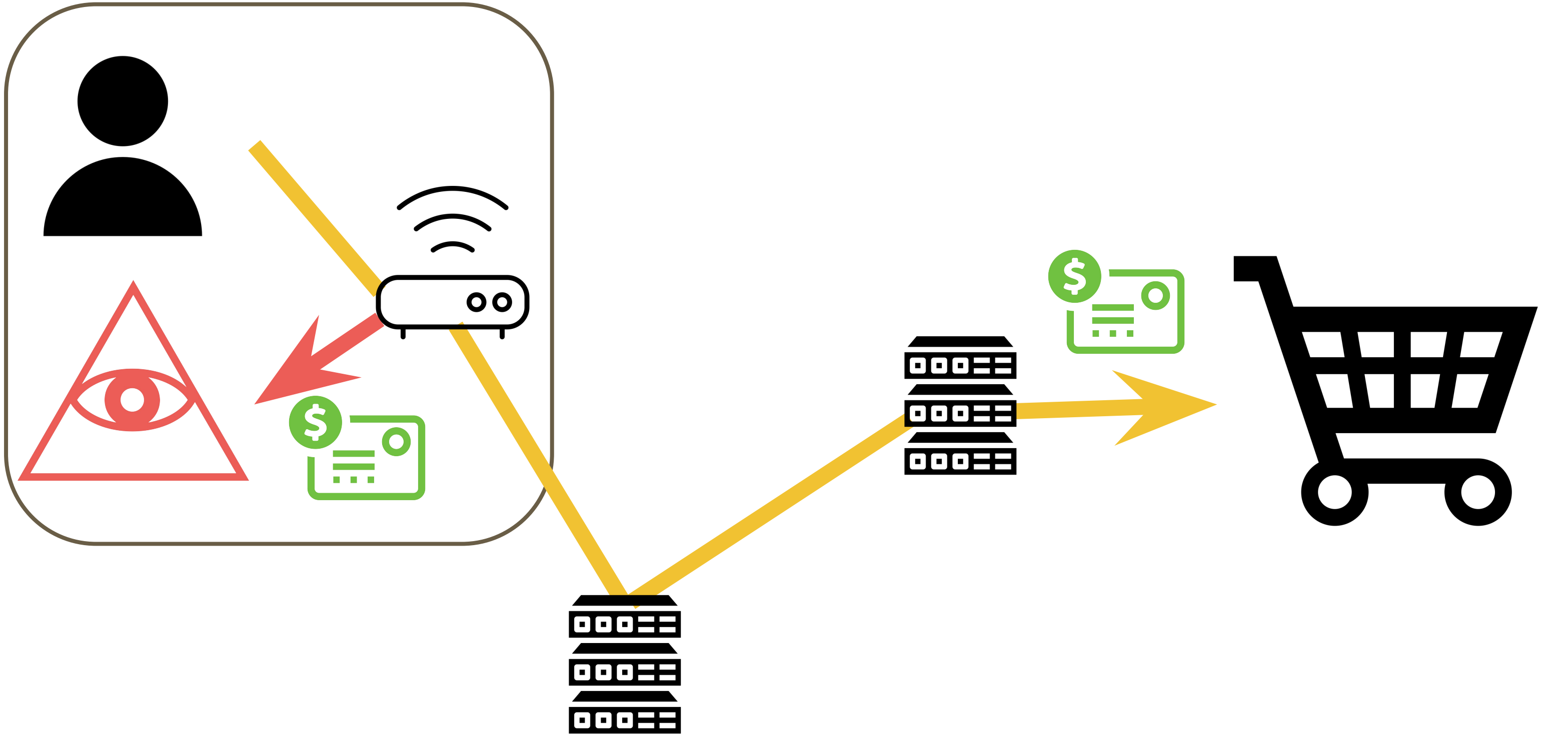
The Snooper



The Snooper



The Snooper



P.C. 90

OPENED BY

EXAMINER 524

…M & HAAS CO.
INCORPORATED
…ST WASHINGTON SQUARE
…ILADELPHIA, PA.

PHILADELPHIA
JUN 20
6 PM
1940
WILLIAM PENN

VIA AIR MAIL

…ar Avion
…y Air Mail
…FORM 2978

Mr. L. Ogilvie,
Long Ashton Station,
Department of Agriculture,
Research Station,
Long Ashton, England.


Ecrit en Allemande

Mr.

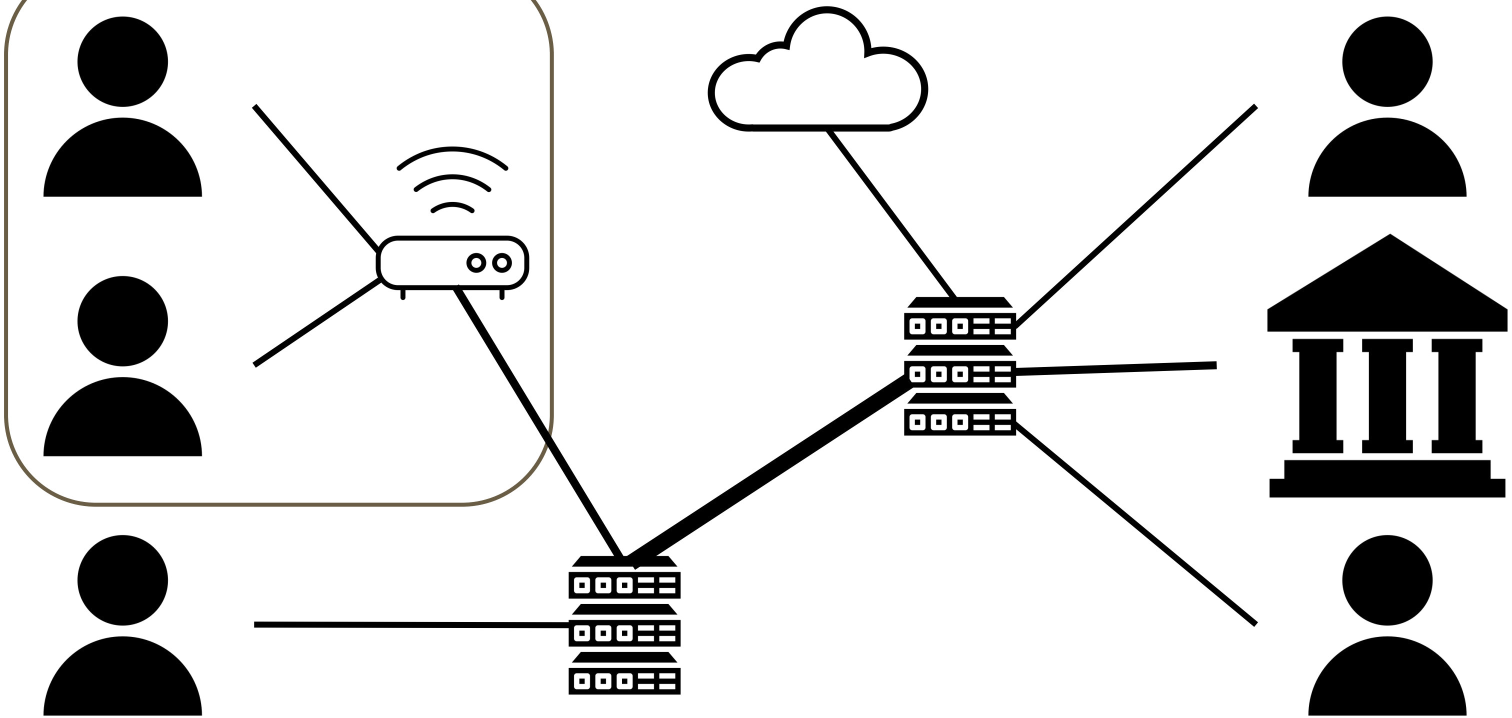
Sally Levy

210 West 101st. Street

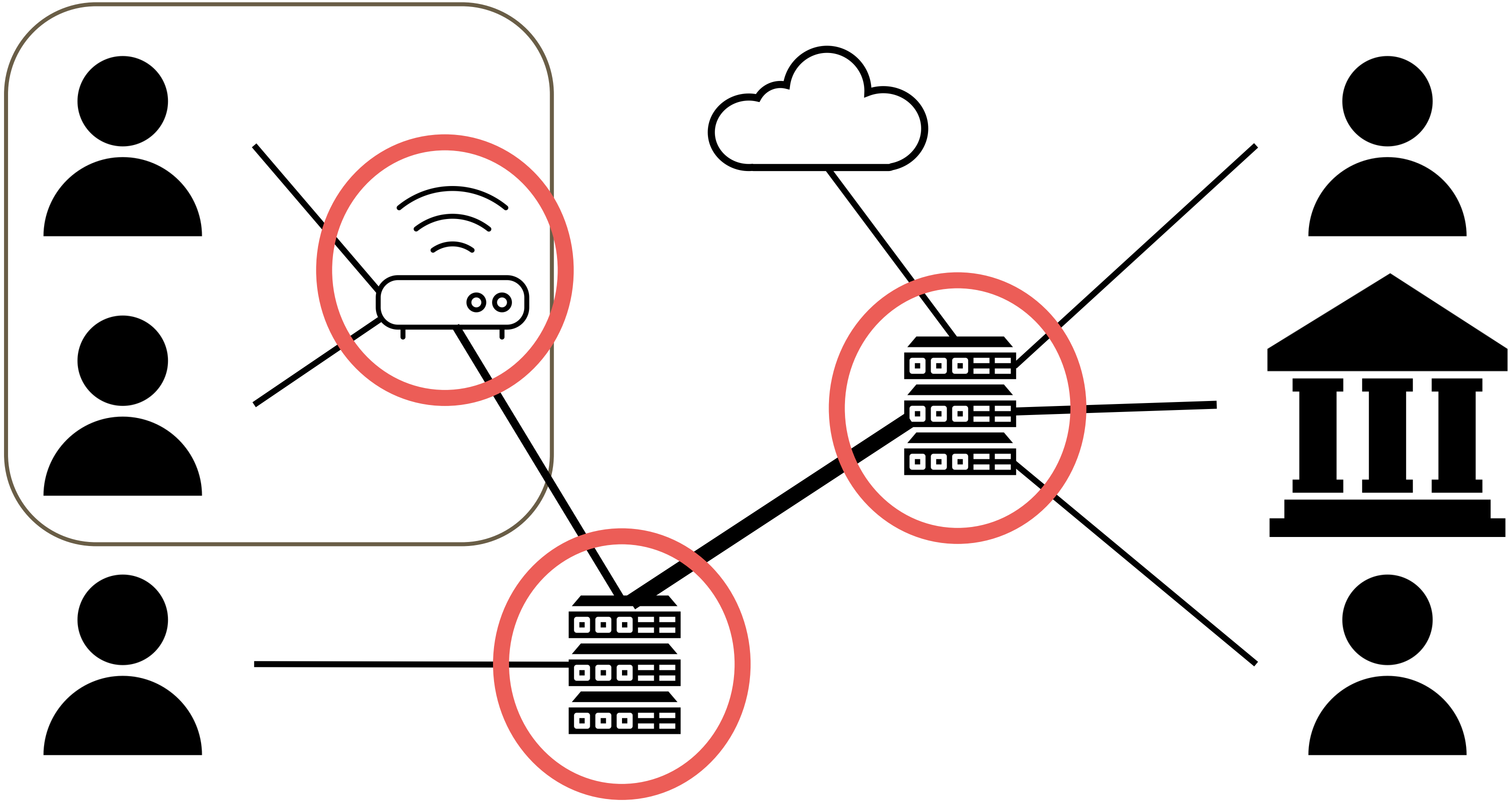
New-York City

U S A

The Snooper



The Snooper



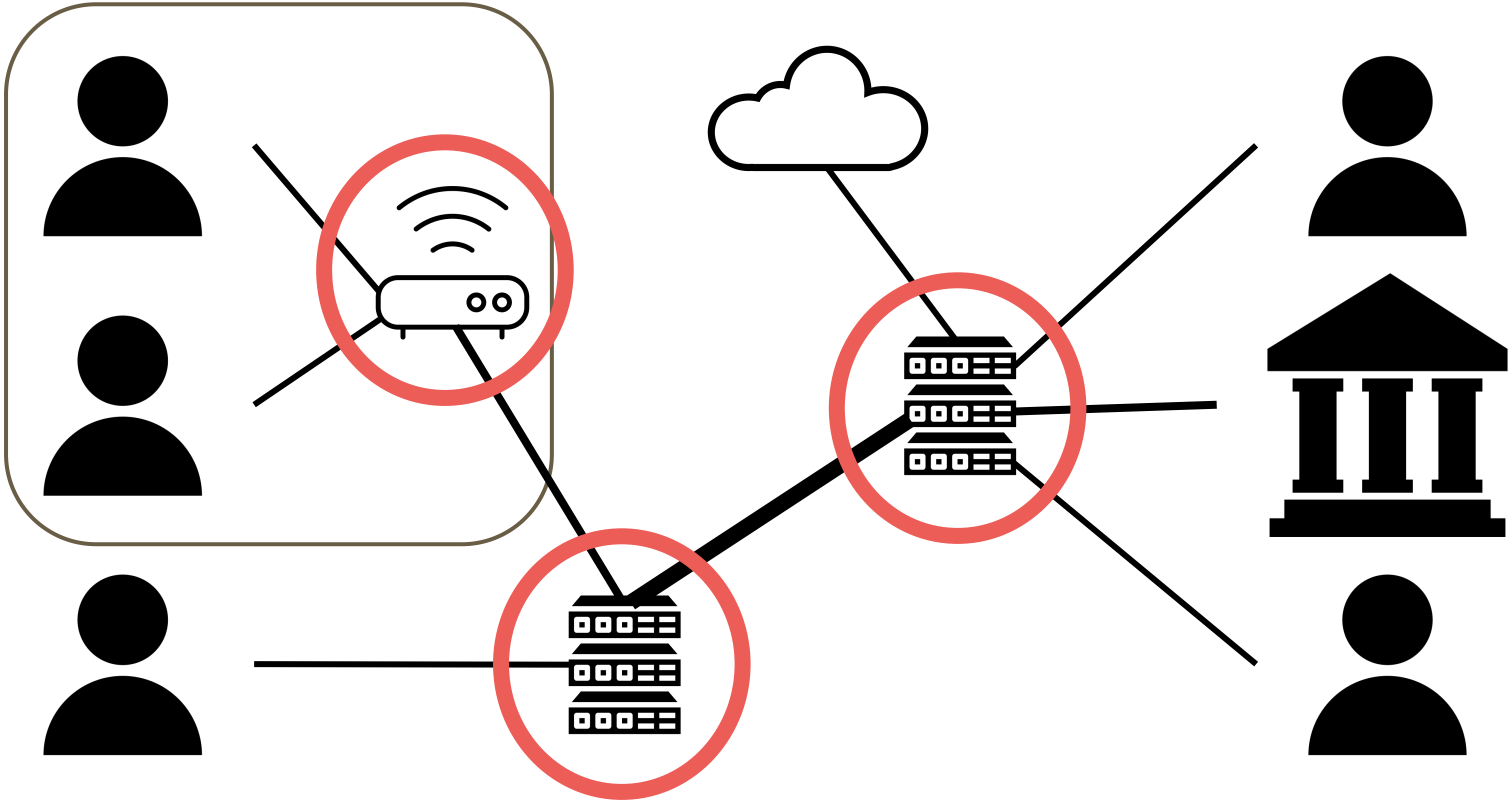
```
➜ ~ traceroute google.com
traceroute to google.com (216.58.219.206), 64 hops max, 52 byte packets
 1  cc-wlan-1-vlan3562-1.net.columbia.edu (160.39.252.2)  2.698 ms  2.311 ms  1.555 ms
 2  phi-core-1-x-cc-wlan-1.net.columbia.edu (128.59.255.225)  1.683 ms  1.698 ms  1.653 ms
 3  nyser111-gw-1-x-phi-core-1.net.columbia.edu (128.59.255.14)  2.106 ms  2.007 ms  1.816 ms
 4  nyser32-gw-1-x-nyser111-gw-1.net.columbia.edu (128.59.255.9)  8.161 ms  2.492 ms  3.124 ms
 5  be4222.rcr24.jfk01.atlas.cogentco.com (38.122.8.209)  2.472 ms  2.381 ms  2.582 ms
 6  be2897.ccr42.jfk02.atlas.cogentco.com (154.54.84.213)  2.725 ms  2.260 ms  2.754 ms
 7  be2061.ccr21.jfk05.atlas.cogentco.com (154.54.3.70)  2.898 ms  4.139 ms  2.952 ms
 8  tata.jfk05.atlas.cogentco.com (154.54.12.18)  3.705 ms  2.854 ms  2.881 ms
 9  if-ae-12-2.tcore1.n75-new-york.as6453.net (66.110.96.5)  2.821 ms  2.897 ms  3.346 ms
10  72.14.214.68 (72.14.214.68)  3.015 ms
    72.14.195.232 (72.14.195.232)  3.461 ms
    72.14.218.224 (72.14.218.224)  3.865 ms
11  209.85.248.242 (209.85.248.242)  3.952 ms  3.901 ms
    216.239.50.106 (216.239.50.106)  4.658 ms
12  209.85.253.111 (209.85.253.111)  3.984 ms  4.066 ms  4.171 ms
13  lga25s40-in-f206.1e100.net (216.58.219.206)  3.642 ms  3.851 ms  3.591 ms
```

```
➜  ~ traceroute www.columbia.edu
traceroute to www-ltm.cc.columbia.edu (128.59.105.24), 64 hops max, 52 byte packets
 1  cc-wlan-1-vlan3562-1.net.columbia.edu (160.39.252.2)  14.735 ms  2.005 ms  1.733 ms
 2  phi-core-1-x-cc-wlan-1.net.columbia.edu (128.59.255.225)  2.264 ms  1.882 ms  3.439 ms
 3  cc-conc-1-x-phi-core-1.net.columbia.edu (128.59.255.214)  1.956 ms  1.706 ms  2.532 ms
 4  columbia.university (128.59.105.24)  1.833 ms  34.477 ms  2.024 ms
```

```
➜ ~ traceroute cam.ac.uk
traceroute to cam.ac.uk (131.111.150.25), 64 hops max, 52 byte packets
 1  cc-wlan-1-vlan3562-1.net.columbia.edu (160.39.252.2)  31.050 ms  3.855 ms  7.104 ms
 2  phi-core-1-x-cc-wlan-1.net.columbia.edu (128.59.255.225)  6.714 ms  8.490 ms  3.632 ms
 3  nyser111-gw-1-x-phi-core-1.net.columbia.edu (128.59.255.14)  434.333 ms  314.247 ms  6.011 ms
 4  nyser32-gw-1-x-nyser111-gw-1.net.columbia.edu (128.59.255.9)  13.434 ms  3.637 ms  5.680 ms
 5  nyc-9208-columbia.nysernet.net (199.109.4.13)  38.134 ms  2.071 ms  1.959 ms
 6  i2-newy-nyc-9208.nysernet.net (199.109.5.2)  2.150 ms  2.233 ms  2.052 ms
 7  internet2.mx1.ams.nl.geant.net (62.40.124.46)  80.376 ms  85.414 ms  85.330 ms
 8  ae2.mx1.lon.uk.geant.net (62.40.98.80)  86.359 ms  84.861 ms  89.197 ms
 9  janet-gw.mx1.lon.uk.geant.net (62.40.124.198)  88.979 ms  101.630 ms  90.211 ms
10  ae28.lowdss-sbr1.ja.net (146.97.33.18)  101.747 ms  88.167 ms  105.850 ms
11  146.97.38.10 (146.97.38.10)  110.981 ms  93.755 ms  98.262 ms
12  146.97.65.106 (146.97.65.106)  92.656 ms  91.787 ms  131.232 ms
13  university-of-cambridge.cambab-rbr1.eastern.ja.net (146.97.130.2)  90.627 ms  96.497 ms  98.185 ms
14  d-dw.s-dw.net.cam.ac.uk (193.60.88.2)  91.853 ms  102.465 ms  163.091 ms
15  d-dw.s-dw.net.cam.ac.uk (193.60.88.2)  91.447 ms  91.686 ms  92.272 ms
16  outside.fw-srv.net.cam.ac.uk (128.232.128.6)  90.952 ms  92.305 ms  127.274 ms
17  link-srv.uis.fw-srv.net.cam.ac.uk (128.232.129.2)  94.121 ms  90.736 ms  91.246 ms
18  primary.admin.cam.ac.uk (131.111.150.25)  91.621 ms  101.475 ms  93.549 ms
```

```
➜  ~ traceroute cam.ac.uk
traceroute to cam.ac.uk (131.111.150.25), 64 hops max, 52 byte packets
 1  cc-wlan-1-vlan3562-1.net.columbia.edu (160.39.252.2)  31.050 ms  3.855 ms  7.104 ms
 2  phi-core-1-x-cc-wlan-1.net.columbia.edu (128.59.255.225)  6.714 ms  8.490 ms  3.632 ms
 3  nyser111-gw-1-x-phi-core-1.net.columbia.edu (128.59.255.14)  434.333 ms  314.247 ms  6.011 ms
 4  nyser32-gw-1-x-nyser111-gw-1.net.columbia.edu (128.59.255.9)  13.434 ms  3.637 ms  5.680 ms
 5  nyc-9208-columbia.nysernet.net (199.109.4.13)  38.134 ms  2.071 ms  1.959 ms
 6  i2-newy-nyc-9208.nysernet.net (199.109.5.2)  2.150 ms  2.233 ms  2.052 ms
 7  internet2.mx1.ams.nl.geant.net (62.40.124.46)  80.376 ms  85.414 ms  85.330 ms
 8  ae2.mx1.lon.uk.geant.net (62.40.98.80)  86.359 ms  84.861 ms  89.197 ms
 9  janet-gw.mx1.lon.uk.geant.net (62.40.124.198)  88.979 ms  101.630 ms  90.211 ms
10  ae28.lowdss-sbr1.ja.net (146.97.33.18)  101.747 ms  88.167 ms  105.850 ms
11  146.97.38.10 (146.97.38.10)  110.981 ms  93.755 ms  98.262 ms
12  146.97.65.106 (146.97.65.106)  92.656 ms  91.787 ms  131.232 ms
13  university-of-cambridge.cambab-rbr1.eastern.ja.net (146.97.130.2)  90.627 ms  96.497 ms  98.185 ms
14  d-dw.s-dw.net.cam.ac.uk (193.60.88.2)  91.853 ms  102.465 ms  163.091 ms
15  d-dw.s-dw.net.cam.ac.uk (193.60.88.2)  91.447 ms  91.686 ms  92.272 ms
16  outside.fw-srv.net.cam.ac.uk (128.232.128.6)  90.952 ms  92.305 ms  127.274 ms
17  link-srv.uis.fw-srv.net.cam.ac.uk (128.232.129.2)  94.121 ms  90.736 ms  91.246 ms
18  primary.admin.cam.ac.uk (131.111.150.25)  91.621 ms  101.475 ms  93.549 ms
```

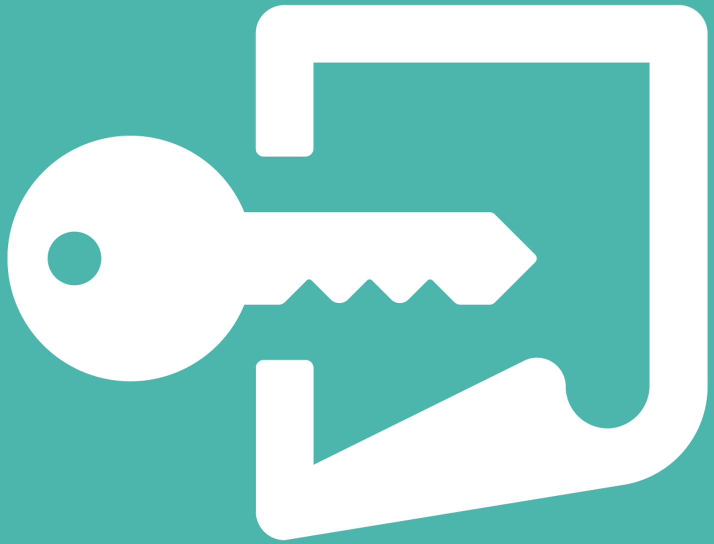
The Snooper



How to send messages and information securely, knowing any info transmitted over the internet can be stolen and we cannot trust anyone?

0. Please Don’t Tell

A brief primer on the codes and ciphers used throughout history to protect information.



Plaintext vs. Ciphertext

Plaintext	`I love the sun`
Ciphertext	`w jd7h bmg vns`

Cipher Shift (or substitution), aka Caesar Cipher

Plaintext	`I love the sun`
Ciphertext	`? ???? ??? ???`

Cipher Shift (zero or no shift)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cipher Shift (zero or no shift)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cipher Shift (shift of one)

Plaintext Alphabet			C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Cipher Shift (shift of one)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	B	C	D	E	F	G	H	I	J	K	L	M	

Cipher Shift (+1)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	B	C	D	E	F	G	H	I	J	K	L	M	N

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Plaintext `i love the sun`

Ciphertext `??????????????`

Cipher Shift (+1)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	B	C	D	E	F	G	H	I	J	K	L	M	N

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Plaintext `i love the sun`

Ciphertext

Cipher Shift (+1)

Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	B	C	D	E	F	G	H	I	J	K	L	M	N

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Plaintext `i love the sun`

Ciphertext `j`

Cipher Shift (+1)

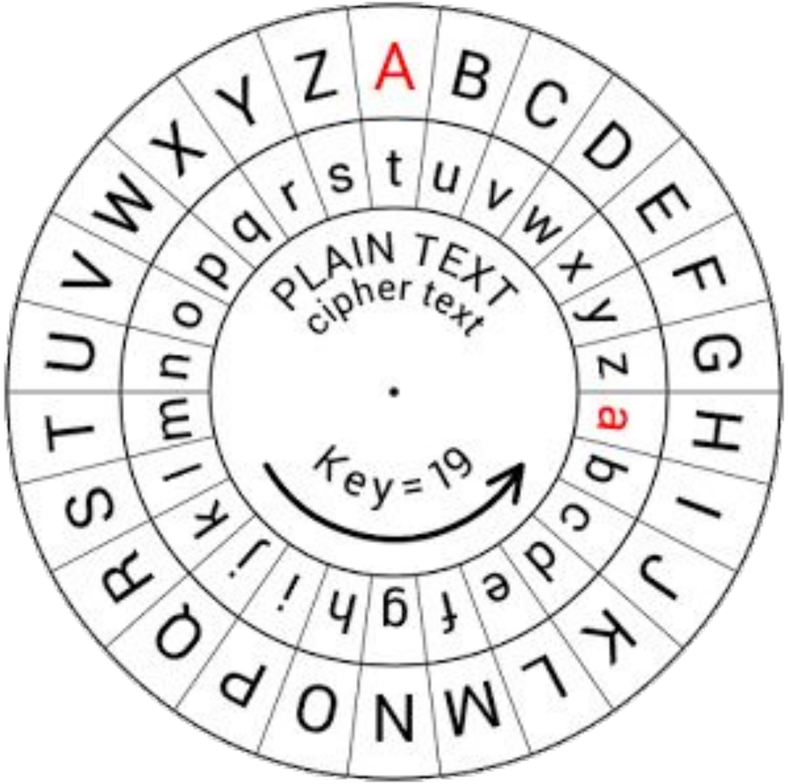
Plaintext Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Ciphertext Alphabet	B	C	D	E	F	G	H	I	J	K	L	M	N

Plaintext Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext Alphabet	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Plaintext `i love the sun`

Ciphertext `j mpwf uif tvo`

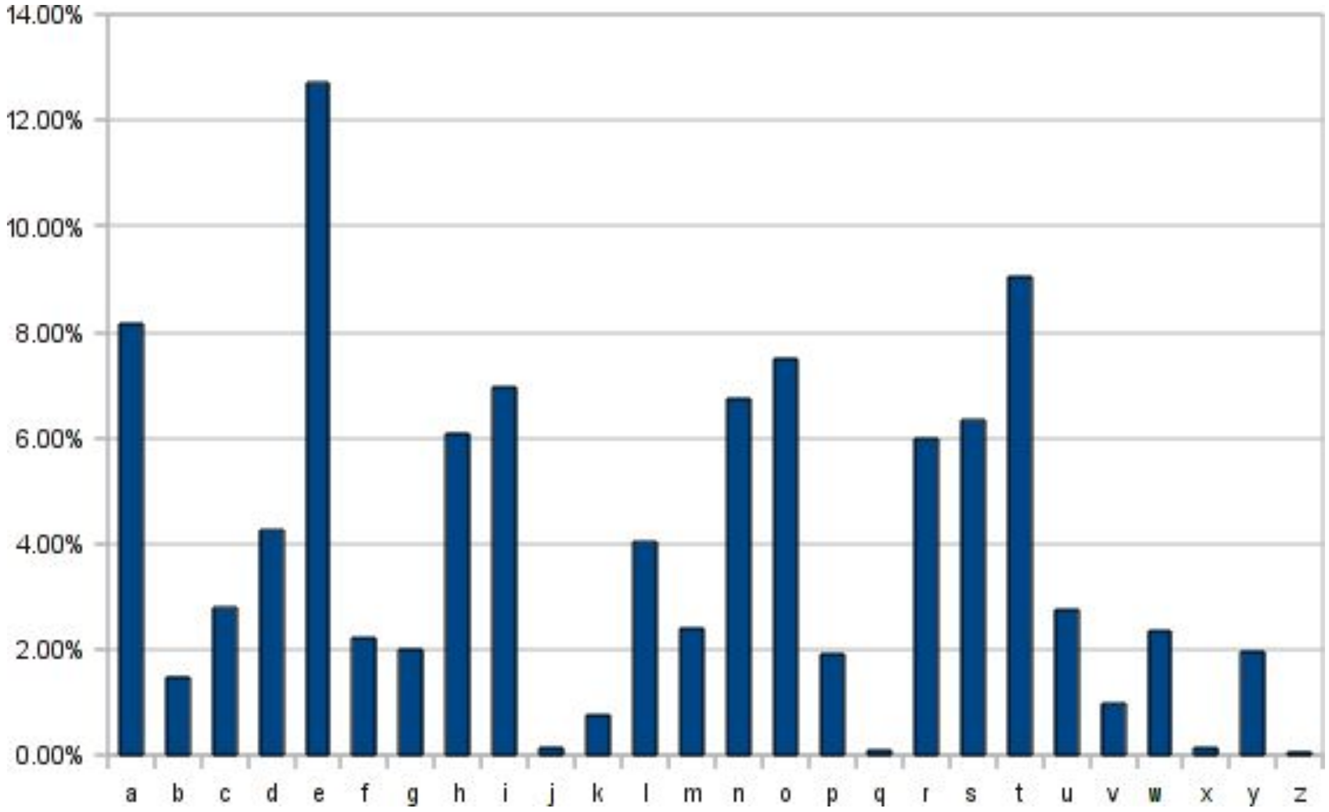
Cipher Shift Wheel



A brief history of how Caesar Cipher was broken ... and rest is history

Cipher Shift Decoded (or rather, decrypted!)

Cipher Shift Decoded (or rather, decrypted!)



An in-class exercise … time to become Code Breakers

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? B ___ ___ ___ ___ ___

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? _B_ _E_ __ __ __ __

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? __B__ __E__ __R__ ___ ___ ___

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? _B_ _E_ _R_ _L_ __ __

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? _B_ _E_ _R_ _L_ _I_ __

Cipher Shift Decoded (or rather, decrypted!)

Can you guess? B E R L I N

Another Cipher Shift Decoded (with numbers)

Can you guess? 2 ___ ___ ___ ___ ___

Another Cipher Shift Decoded (with numbers)

Can you guess? _2_ _5_ ___ ___ ___ ___

Another Cipher Shift Decoded (with numbers)

Can you guess? __2__ __5__ __8__ ___ ___ ___

But wait a minute!?

Another exercise …
time to become REAL Code Breakers!

Let's try to break a coded message

MPQZCP HP NLY ELWV LMZFE ESP DAPNTQTND ZQ
XZOPCY NCJAEZRCLASJ, MWZNVNSLTYD, ZC
MTENZTY, HP XFDE QTCDE ELWV LMZFE ESP CZWP
ZQ XLESPXLETND, FYOPCDELYOTYR SZH TE TD
LAAWTPO LYO SZH TE TD QFYOLXPYELW EZ LWW
ESLE EPNSYZWZRJ LTXD EZ LNSTPGP.

Let's try to break a code by hand (you have 10 mins)

E: 21	D: 10	A: 5
L: 18	W: 9	H: 4
Z: 17	C: 8	V: 3
P: 16	Q: 6	J: 3
T: 15	X: 6	R: 3
Y: 11	O: 6	G: 1
N: 10	M: 5	
S: 10	F: 5	

Let's try to break a coded message (key=11)

Before we can talk about the specifics of modern cryptography, blockchains, or bitcoin, we must first talk about the role of mathematics, understanding how it is applied and how it is fundamental to all that technology aims to achieve.

Let's talk about DATA ...

A little “bit” of data

Unit	Size	Comments
Bit (b)	1 or 0	Short for Binary Digit, after the binary code
Byte (B)	8 bits	WHY 8?

American Standard Code for Information Interchange

ASCII (character encoding standard/protocol)

Binary Decoding (8-bit)

0	0011 0000	C	0100 0011	P	0101 0000	c	0110 0011	p	0111 0000
1	0011 0001	D	0100 0100	Q	0101 0001	d	0110 0100	q	0111 0001
2	0011 0010	E	0100 0101	R	0101 0010	e	0110 0101	r	0111 0010
3	0011 0011	F	0100 0110	S	0101 0011	f	0110 0110	s	0111 0011
4	0011 0100	G	0100 0111	T	0101 0100	g	0110 0111	t	0111 0100
5	0011 0101	H	0100 1000	U	0101 0101	h	0110 1000	u	0111 0101
6	0011 0110	I	0100 1001	V	0101 0110	I	0110 1001	v	0111 0110
7	0011 0111	J	0100 1010	W	0101 0111	j	0110 1010	w	0111 0111
8	0011 1000	K	0100 1011	X	0101 1000	k	0110 1011	x	0111 1000
9	0011 1001	L	0100 1100	Y	0101 1001	l	0110 1100	y	0111 1001
A	0100 0001	M	0100 1101	Z	0101 1010	m	0110 1101	z	0111 1010
B	0100 0010	N	0100 1110	a	0110 0001	n	0110 1110		
		O	0100 1111	b	0110 0010	o	0110 1111		

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

A little "bit" of data

Unit	Size	Comments
Bit (b)	1 or 0	Short for Binary Digit, after the binary code
Byte (B)	8 bits	Enough info to create letters and numbers (basic unit of computing)
Kilobyte (KB)	1,000 B or 2^{10} bytes	"thousands" in Greek
Megabyte (MB)	1,000 KB or 2^{20} bytes	"large" in Greek
Gigabyte (GB)	1,000 MB or 2^{30} bytes	"giant" in Greek

A little “bit” of data (cont’d)

Unit	Size	Comments
Terabyte (TB)	1,000 GB or 2^{40} bytes	“monster” in Greek, about 2 billion credit card transactions

AWS Snowball (up to 80 TB, 72 TB usable)

A little "bit" of data (cont'd)

Unit	Size	Comments
Terabyte (TB)	1,000 GB or 2^{40} bytes	"monster" in Greek, about 2 billion credit card transactions
Petabyte (PB)	1,000 TB or 2^{50} bytes	Google process more than 1 PB per hour
Exabyte (EB)	1,000 PB or 2^{60} bytes	In 2009, the entire internet was estimated at ~500 EB. In 2013, annual internet traffic flow surpassed 667 EB (Cisco)

AWS Snowmobile!

A little "bit" of data (cont'd)

Unit	Size	Comments
Zettayte (ZB)	1,000 EB or 2^{70} bytes	About 615 billion newspapers (88 copies for every human being)
Yottabyte (YB)	1,000 ZB or 2^{80} bytes	Waaaay too big! Currently, all the combined hard-drives and storage capacity in the world are estimated at <0.0004 YB!

Plaintext vs. Binary Ciphertext (in “old” ASCII)

Plaintext	H E L L O
Binary	1001000 1000101 1001100 1001100 1001111

Key = ?

Plaintext vs. Binary Ciphertext (in “old” ASCII)

Plaintext	H E L L O
Binary	1001000 1000101 1001100 1001100 1001111
Key = David	1000100 1000001 1010110 1001001 1000100

Plaintext vs. Binary Ciphertext (in “old” ASCII)

Plaintext	H E L L O
Binary	1001000 1000101 1001100 1001100 1001111
Key = David	1000100 1000001 1010110 1001001 1000100
Ciphertext (xor)	

Boolean Logic & Logic Gates

AND

OR

NOT

NOR

NAND

XNOR

XOR

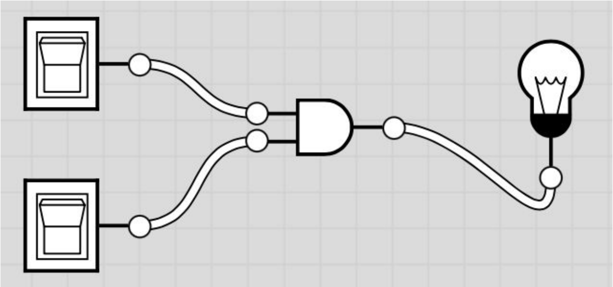
Conjunction (AND), a logical operation

AND

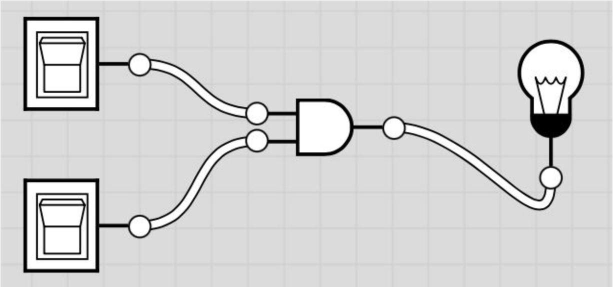
$A \cdot B$ or $A \wedge B$

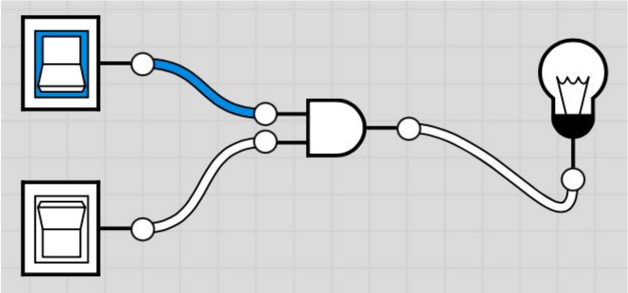
Input		Output
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Conjunction (AND), a logical operation

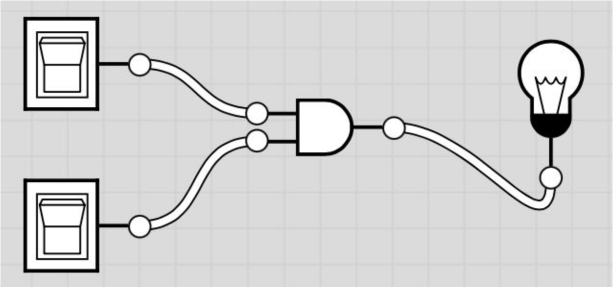


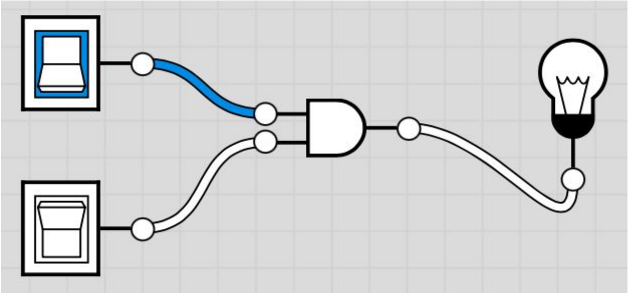
Conjunction (AND), a logical operation

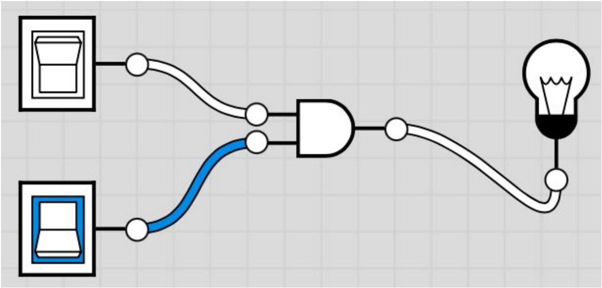




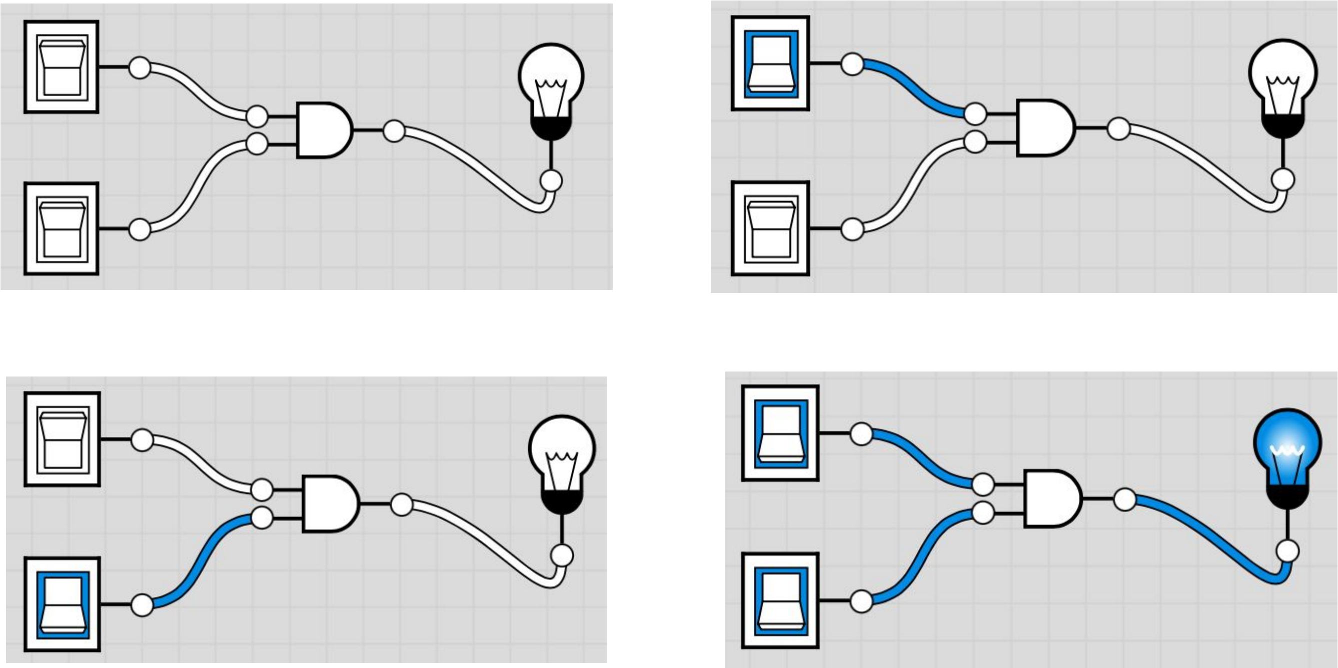
Conjunction (AND), a logical operation



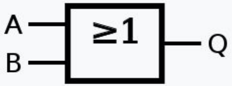




Conjunction (AND), a logical operation

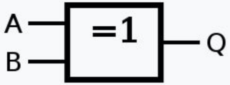


Disjunction (OR), a logical operation

OR			$A + B$ or $A \vee B$	see truth table below

Input		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive Disjunction (XOR), a logical operation

XOR			$A \oplus B$ or $A \veebar B$	see truth table below

Input		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive Disjunction (XOR), a logical operation

Input	→	Output
A	B	(A ⊕ B)
0	0	0
1	1	0
1	0	1
0	1	1

Plaintext vs. Binary Ciphertext (in "old" ASCII)

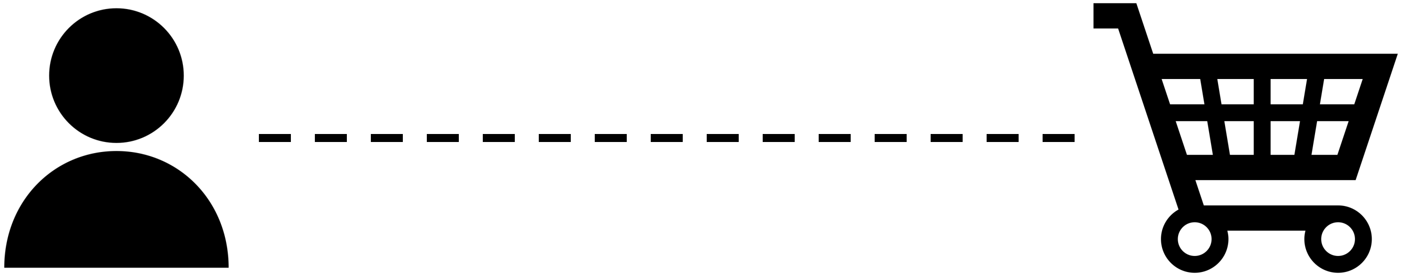
Plaintext	H E L L O
Binary	1001000 1000101 1001100 1001100 1001111
Key = David	1000100 1000001 1010110 1001001 1000100
Ciphertext (xor)	**0001100 0000100 0011010 0000101 0001011**

Plaintext vs. Binary Ciphertext (in "old" ASCII)

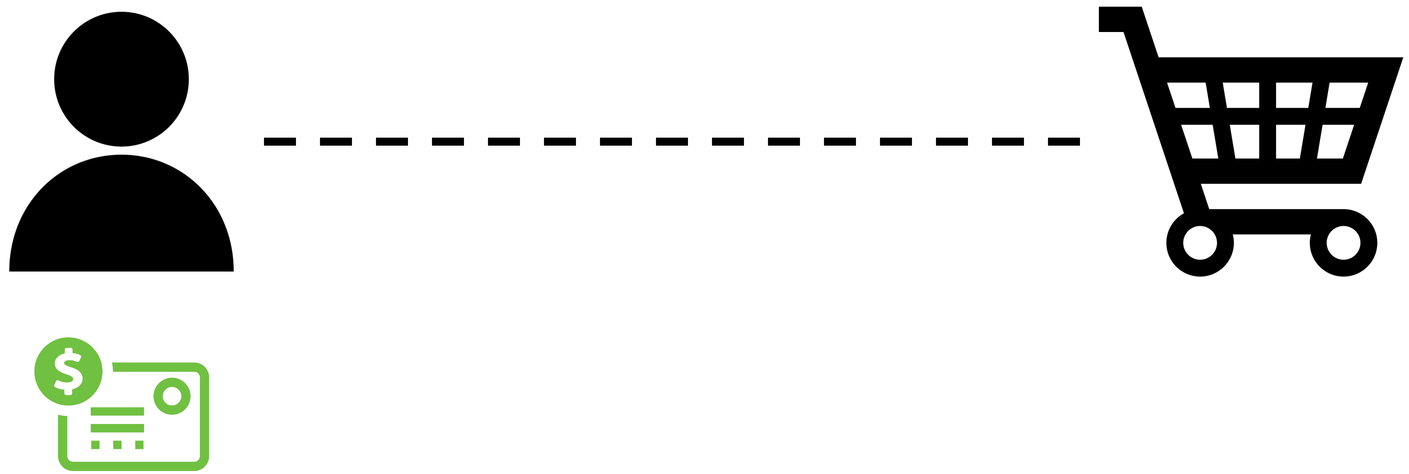
Plaintext	H	E	L	L	O
Binary	1001000	1000101	1001100	1001100	1001111
Key = Dave	**1000100**	1000001	1010110	0110010	**1000100**
	D	**a**	**v**	**e**	**D**

Let’s discuss in the context of a case model …

Securing a connection

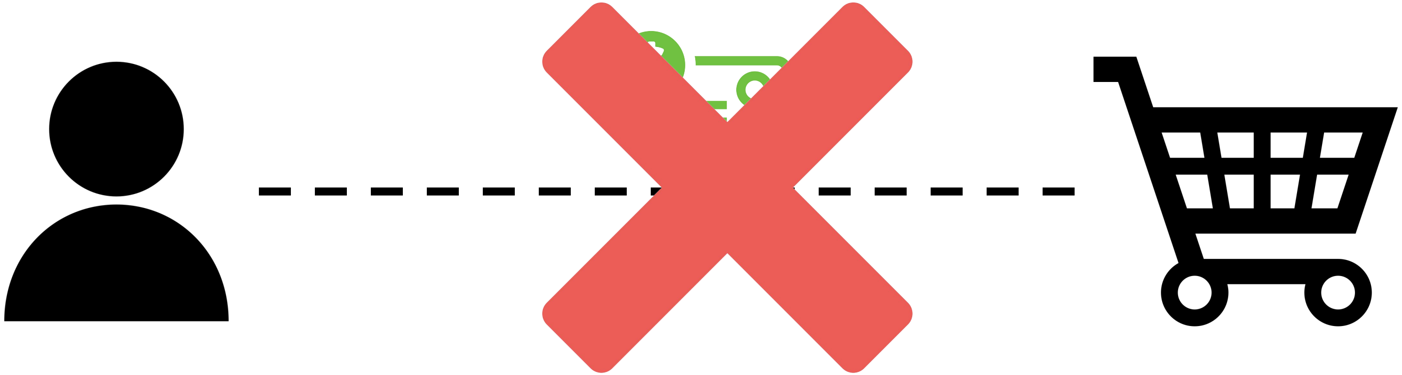


Securing a connection



Securing a connection

Securing a connection



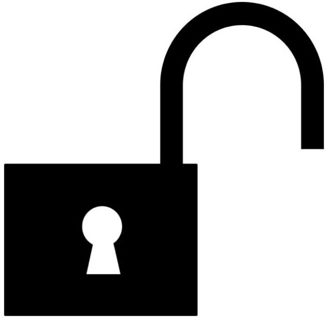
Some "Key" Definitions!

Key

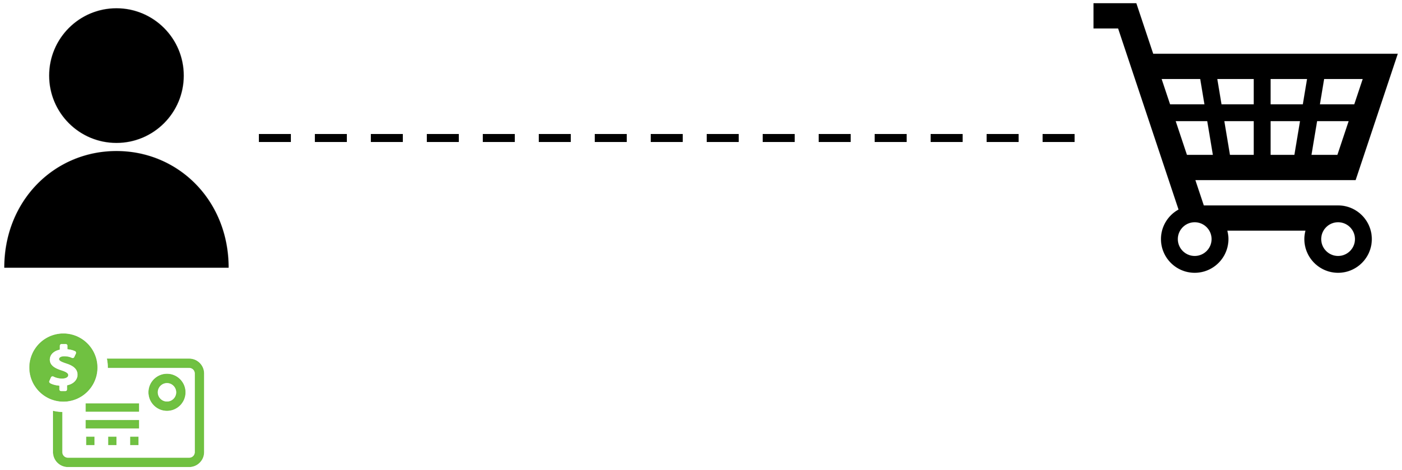
OR

Closed padlock (locked)
OR encrypted

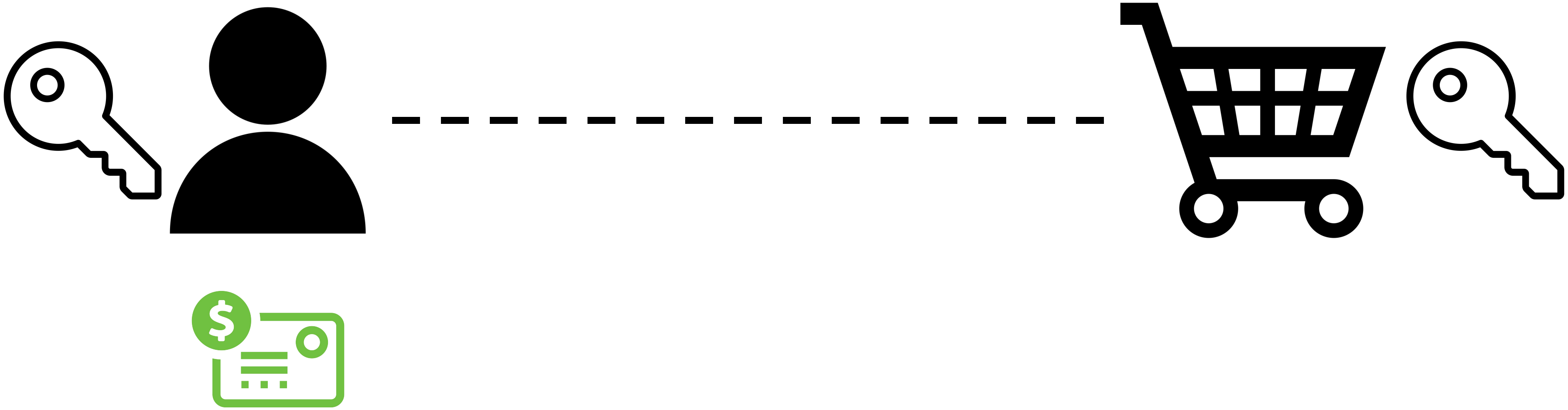
Open padlock (unlocked)
OR decrypted



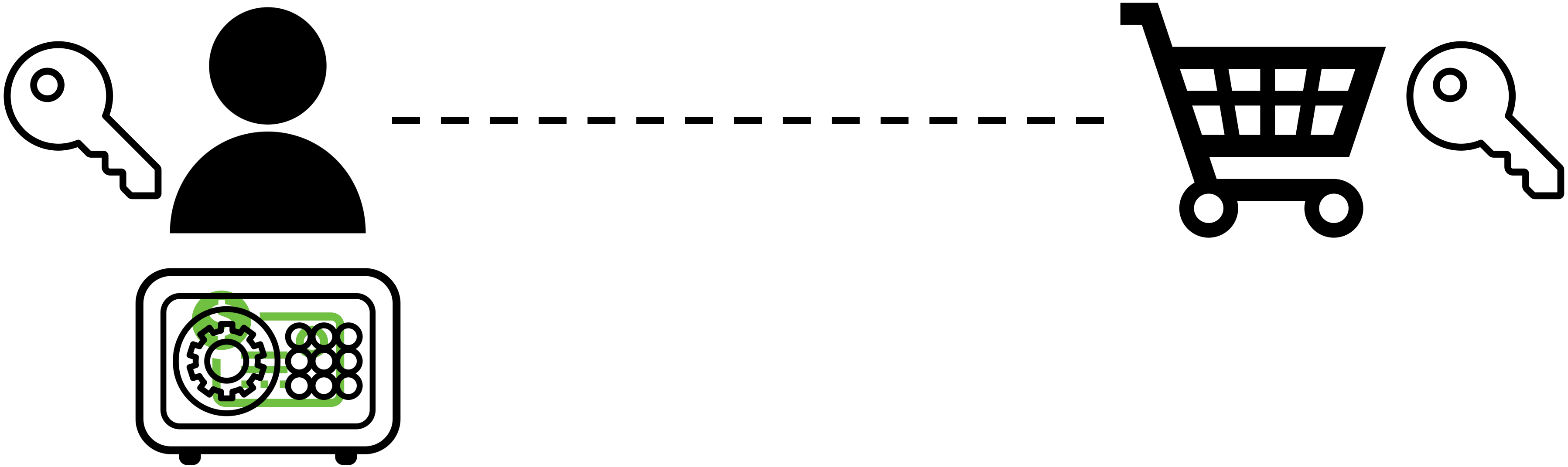
Securing a connection



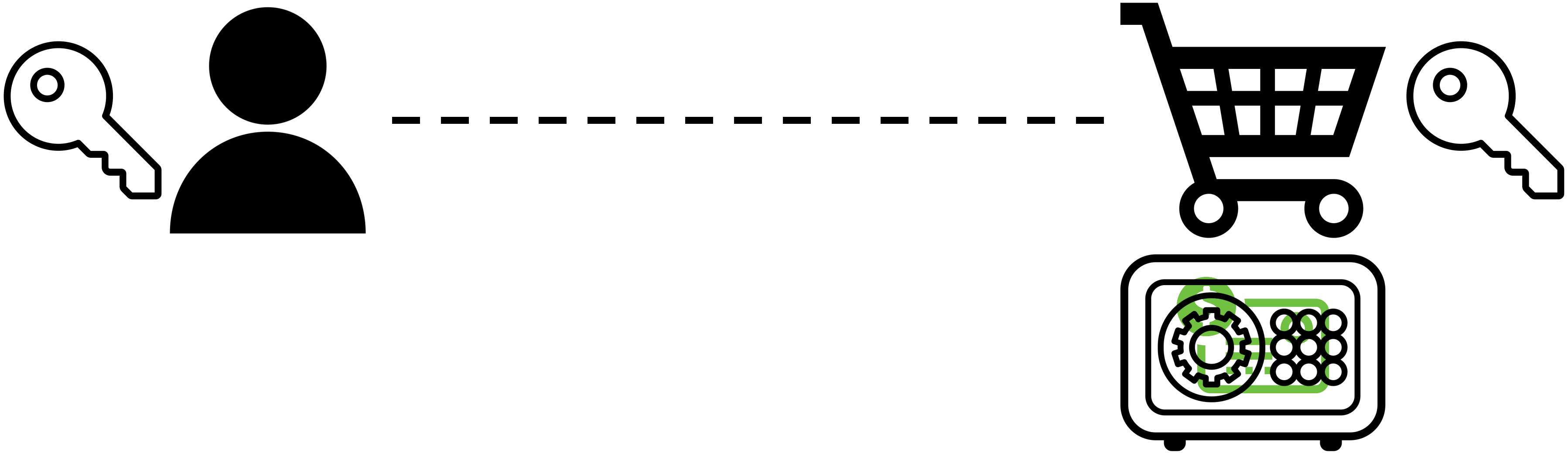
Securing a connection



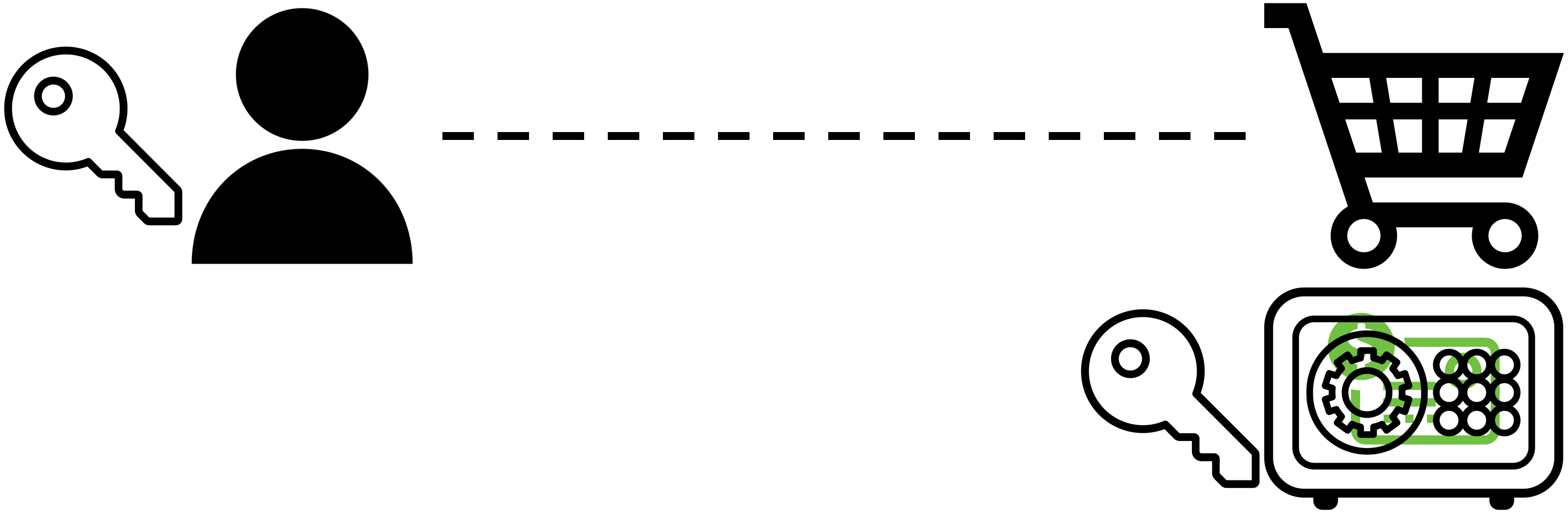
Securing a connection



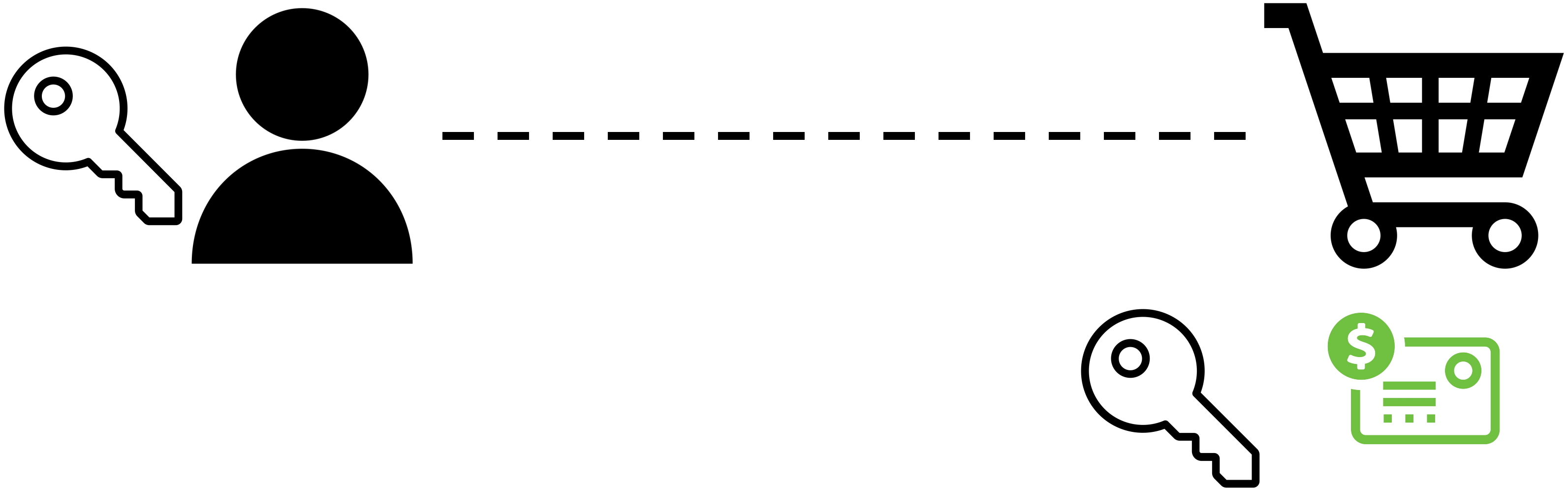
Securing a connection



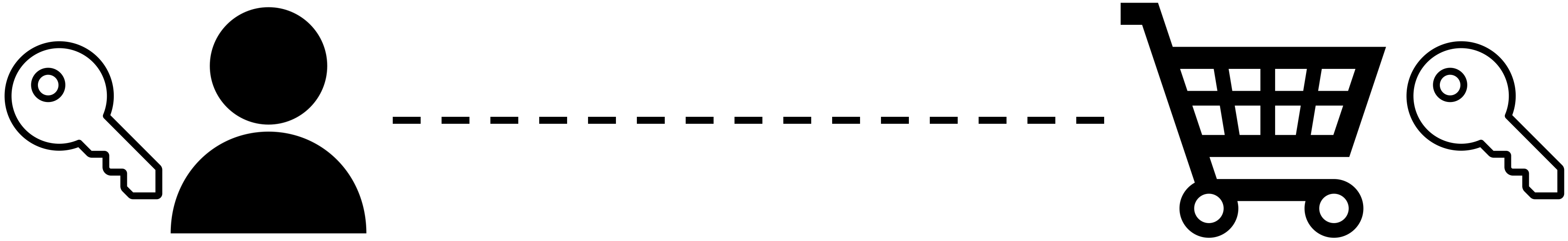
Securing a connection



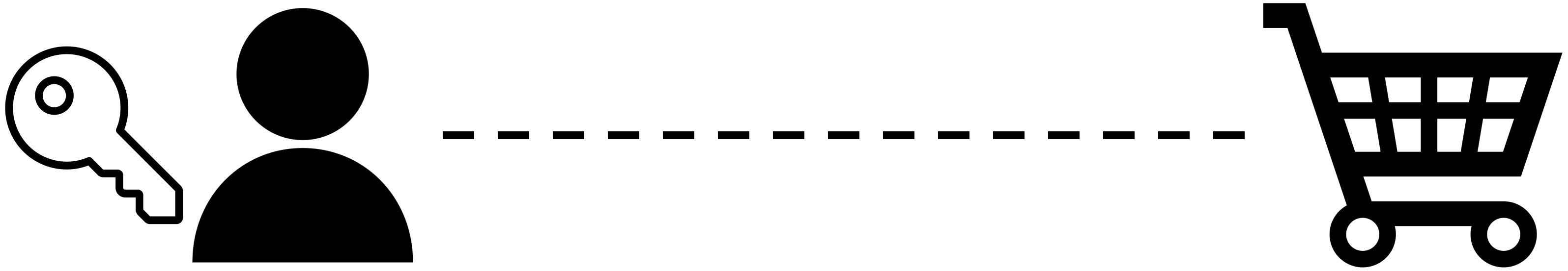
Securing a connection



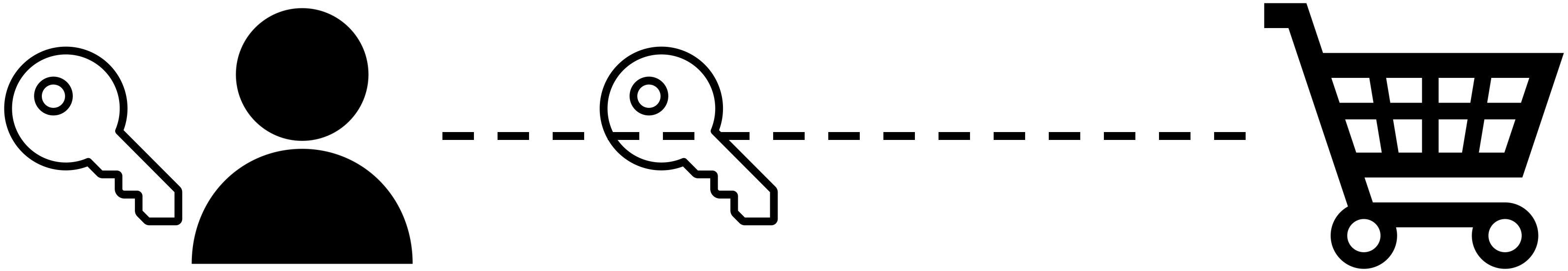
Securing a connection



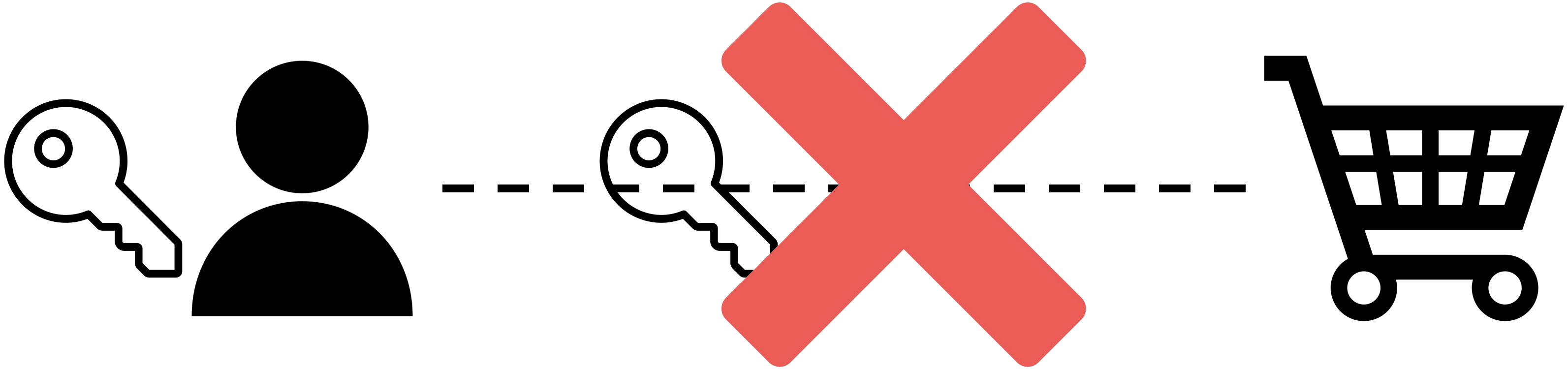
Securing a connection



Securing a connection



Securing a connection



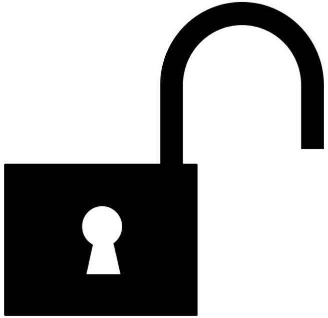
Remember these?!

Key

OR

Closed padlock (locked)

Open padlock (unlocked)



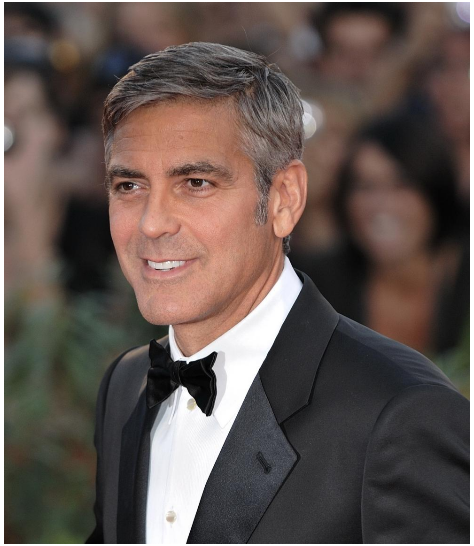
But how to safely and securely transmit the cipher-shift "key"?

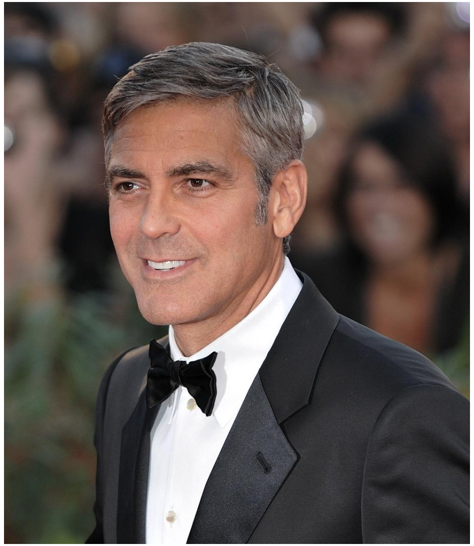
A clever thought-experiment to transmit key, esp to those you haven’t met before!

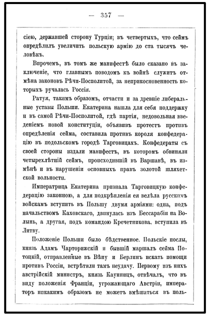
How it works? Well, …







— 357 —

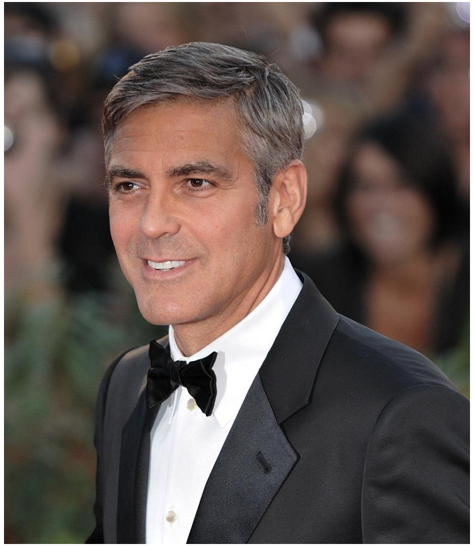
сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

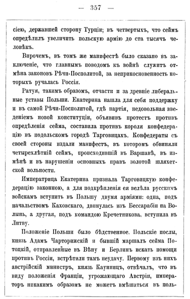
Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауницъ, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-





— 357 —

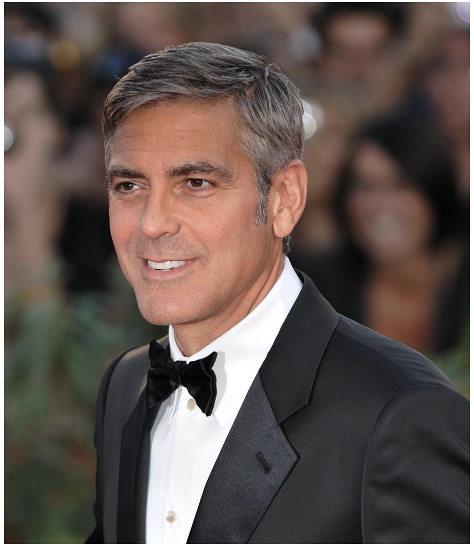
сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

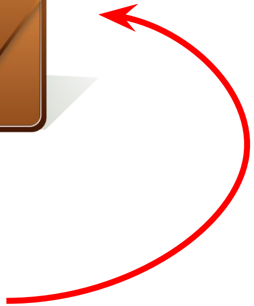
Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауницъ, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-





— 357 —

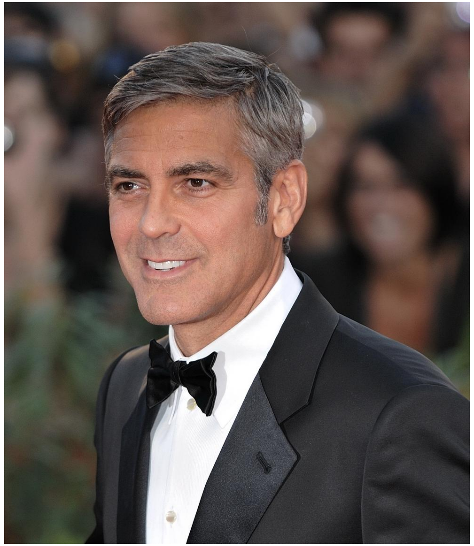
сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

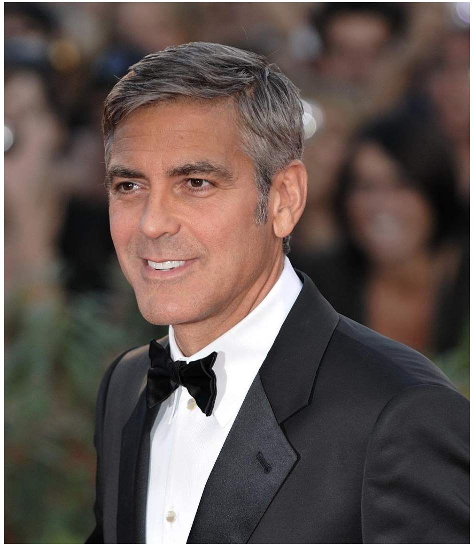
Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

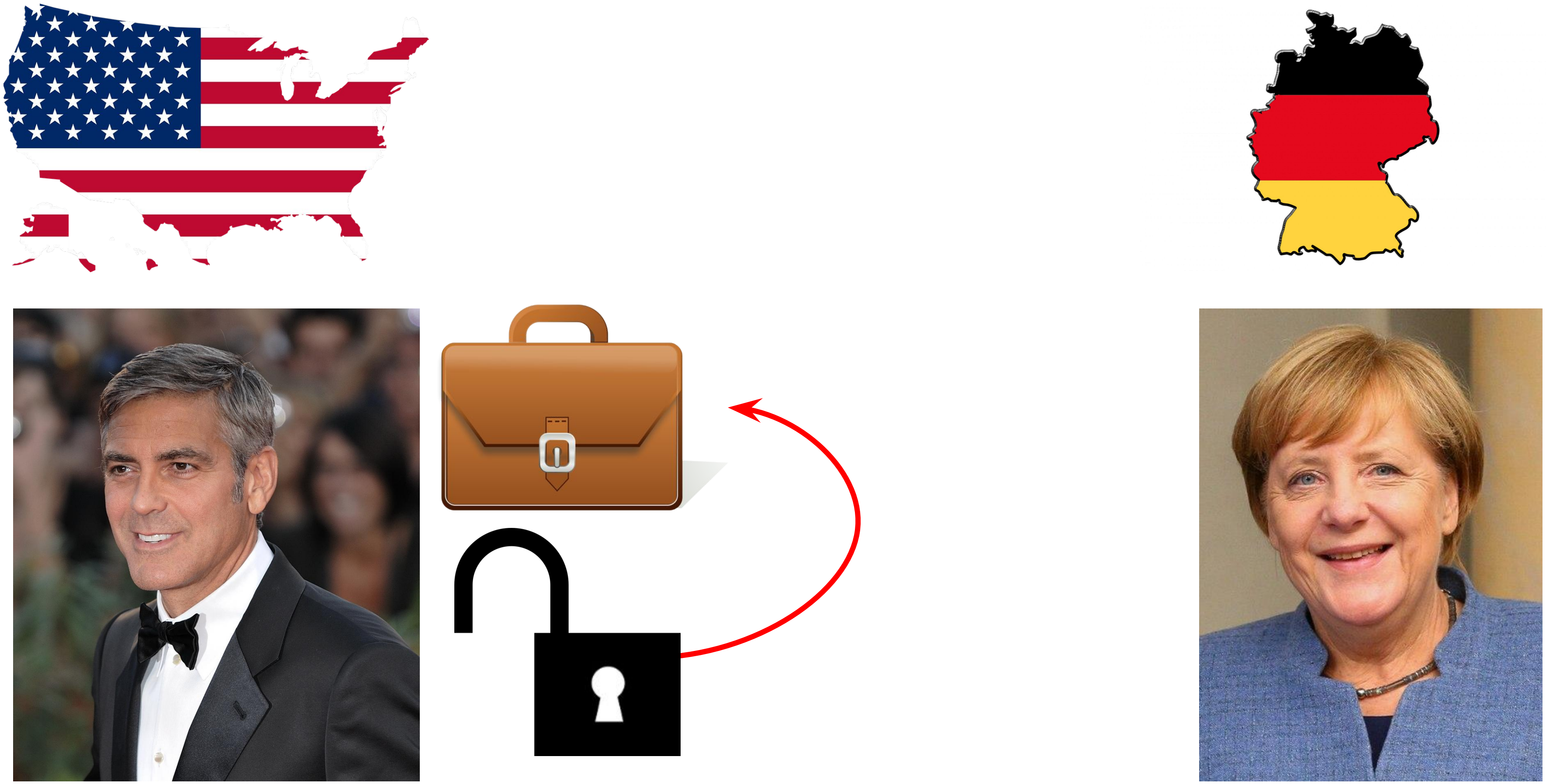
Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

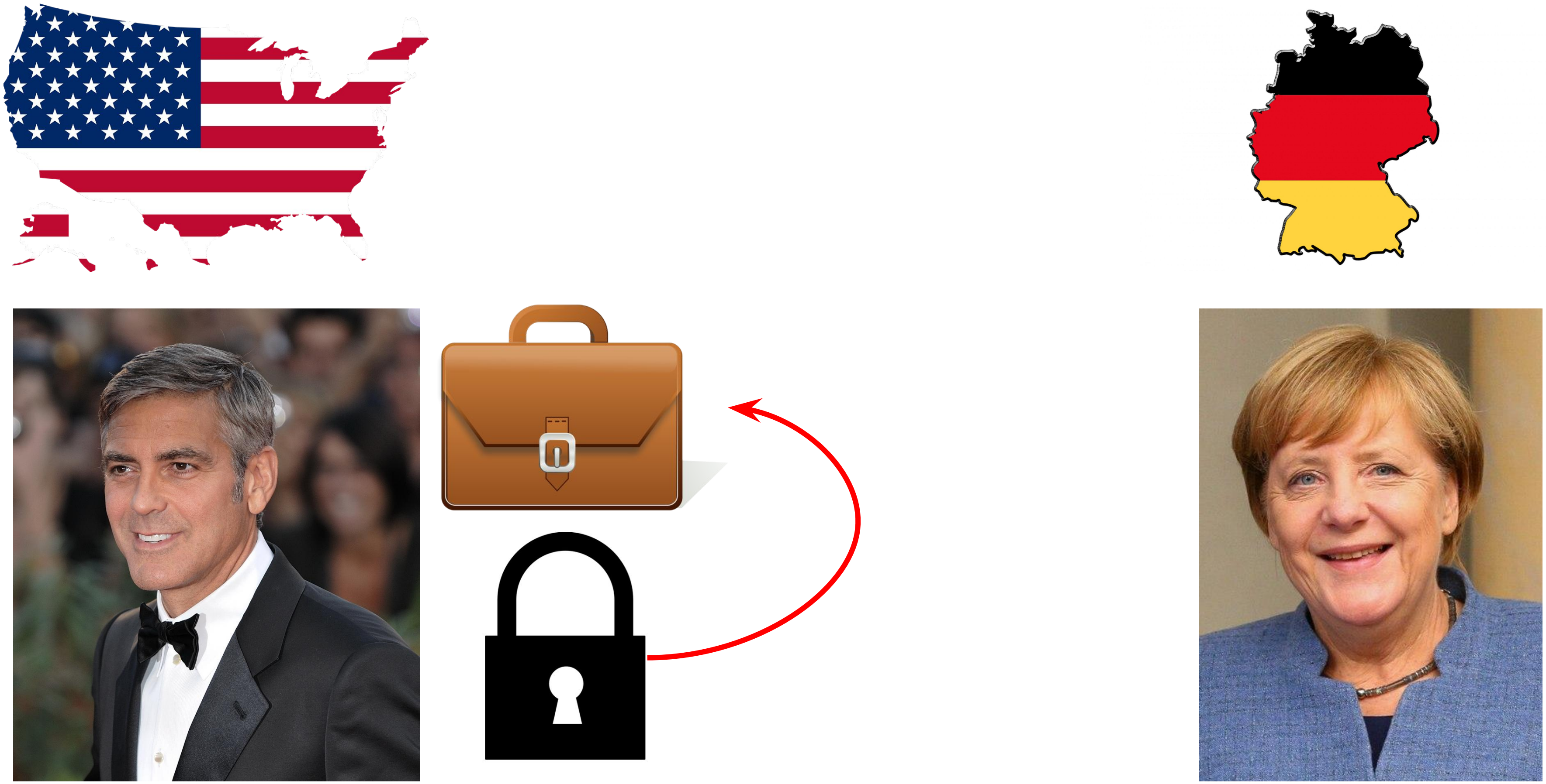
Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

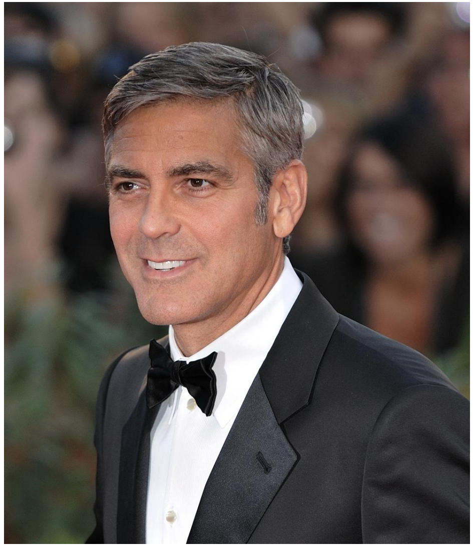
Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауниць, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-

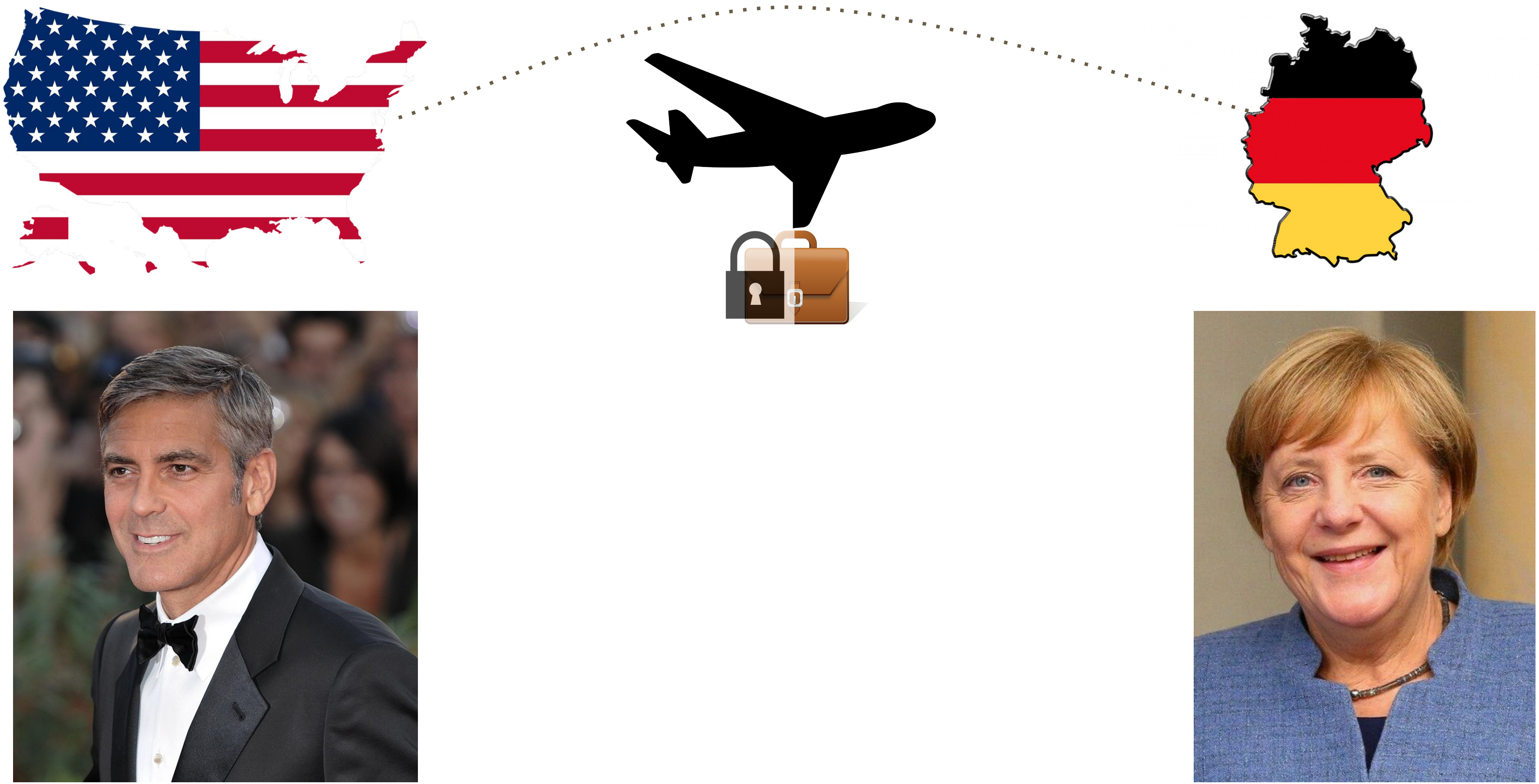


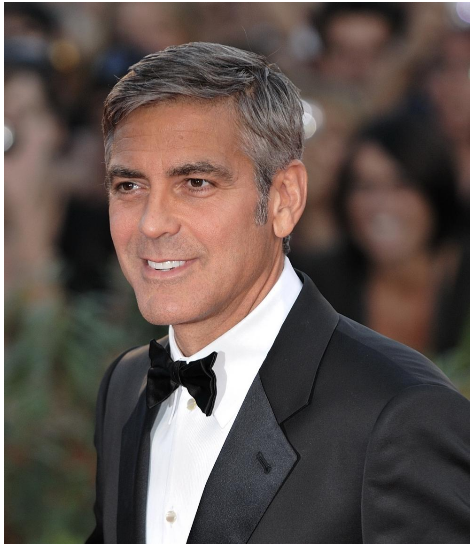


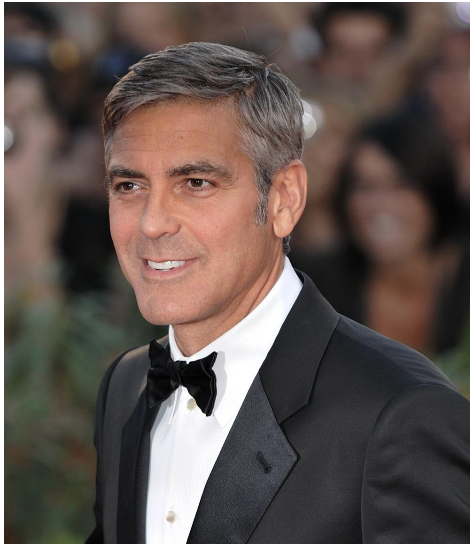


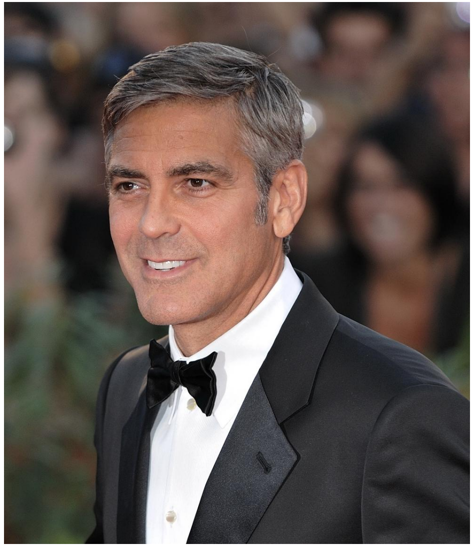


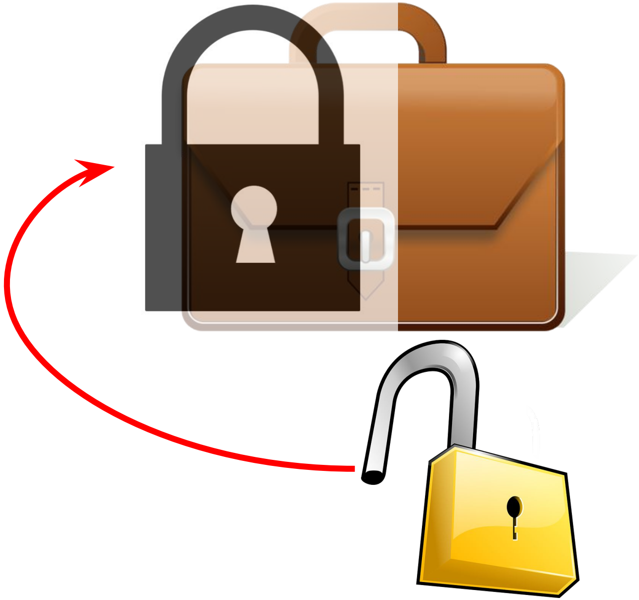




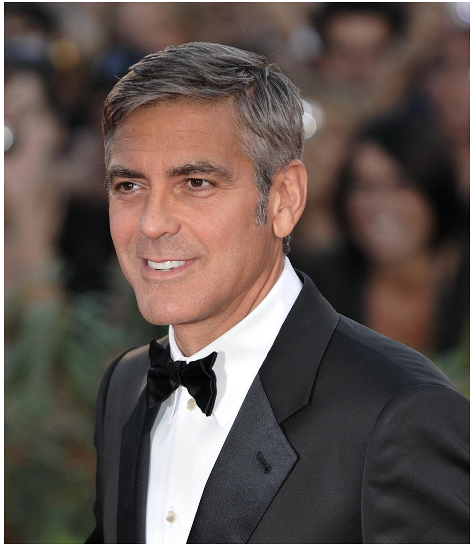




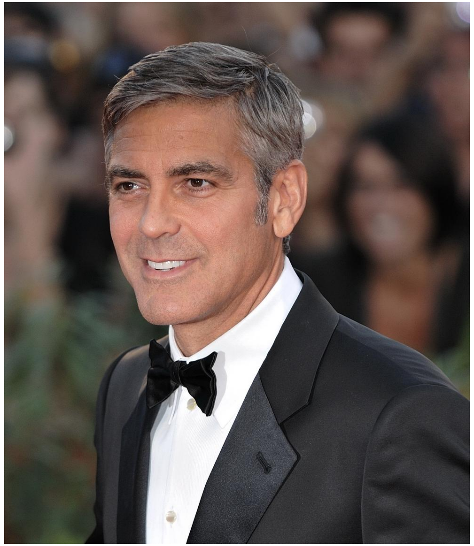




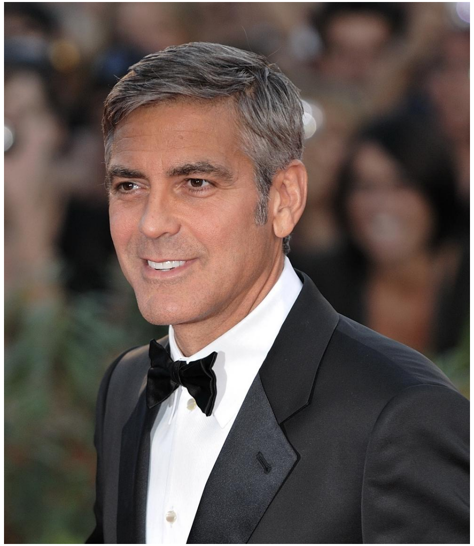




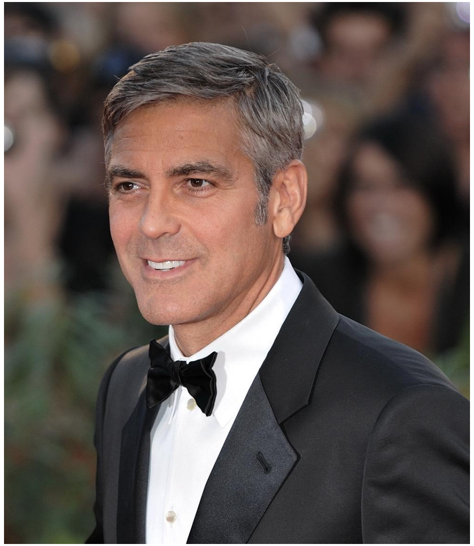


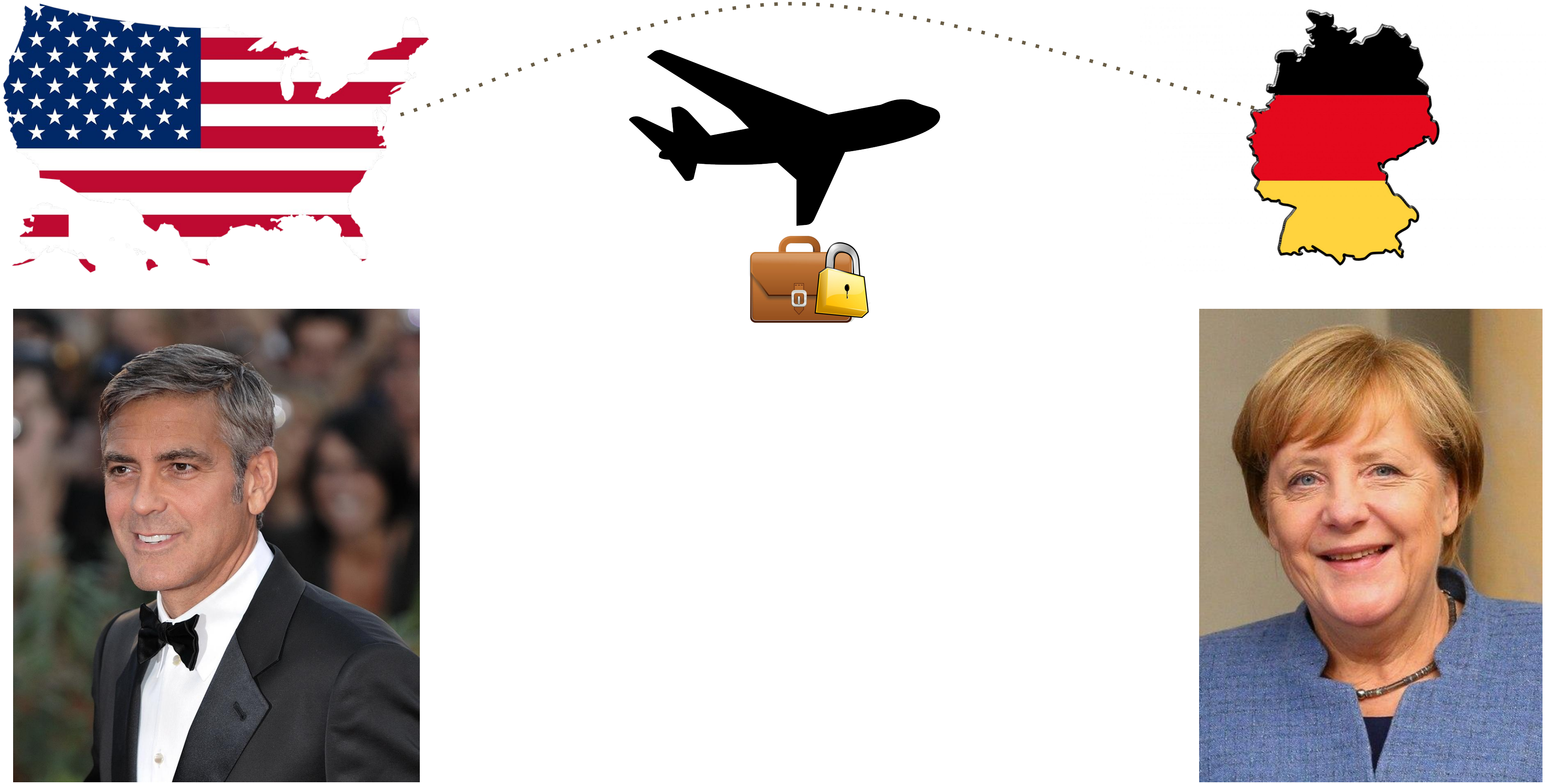




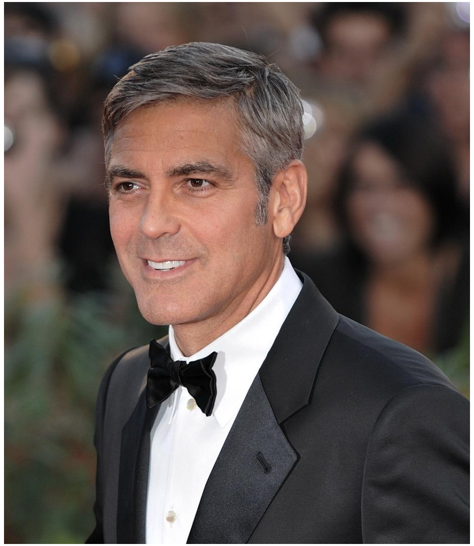




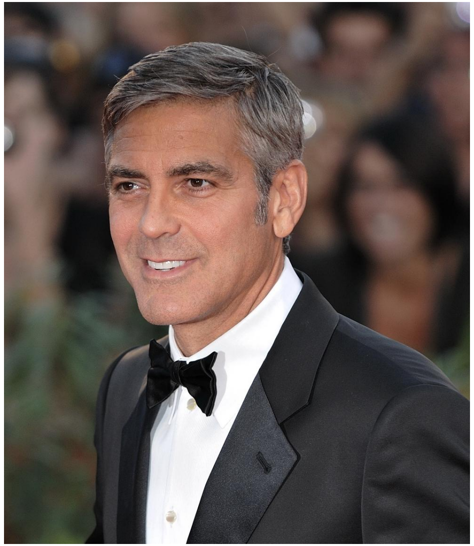



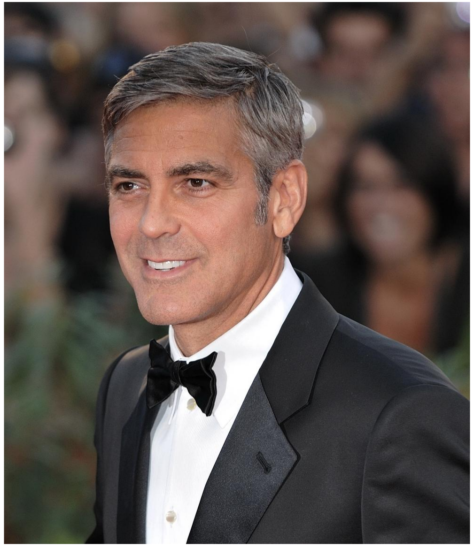


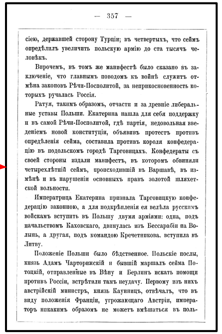










— 357 —

сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

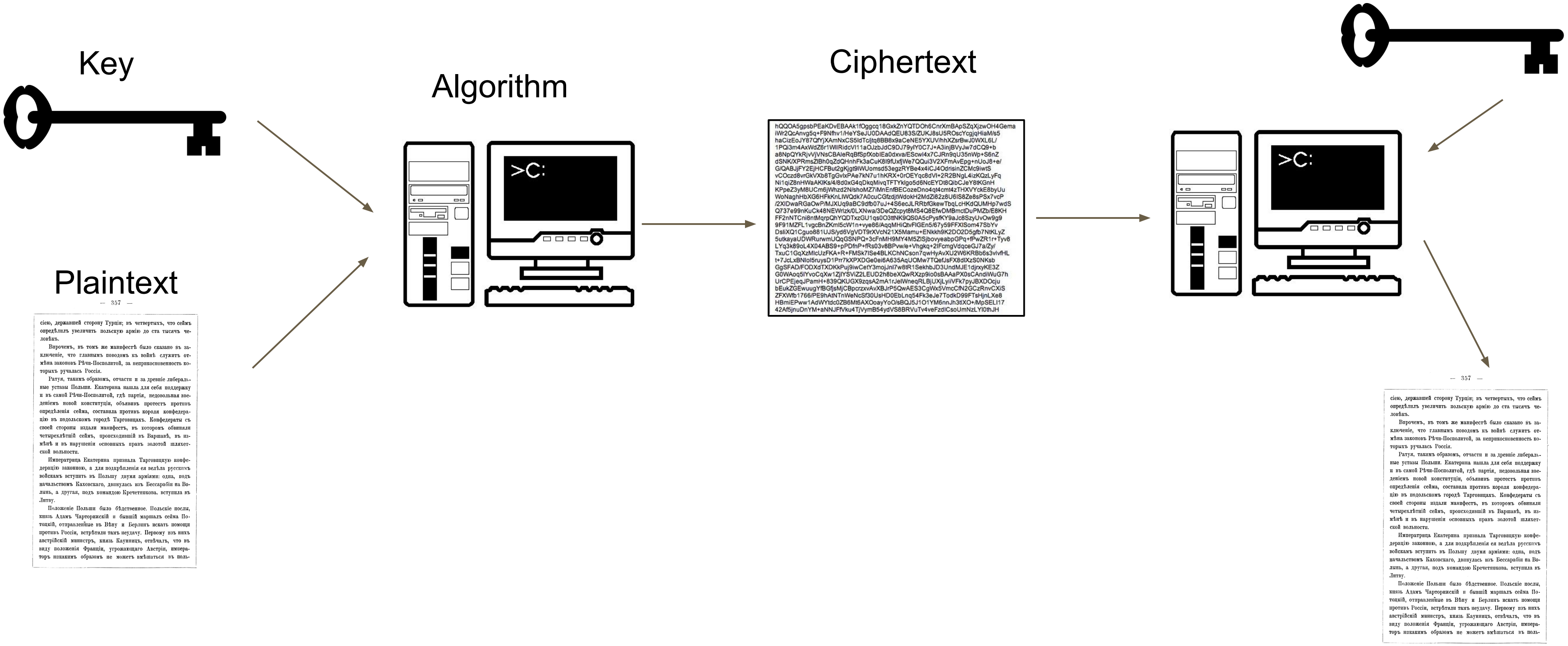
Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауницъ, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-

Isn’t that cool? We exchanged a secret (encrypted) message without having to agree to and exchange keys beforehand!

Digital Cryptography



One encryption on top of another! Remember LIFO?

A clever way to transmit key, in particular to those you haven’t met before!

How we do this in practice?

A clever way to transmit key, in particular to those you haven't met before!

"irreversible" solution = Public + Private Key Pairs

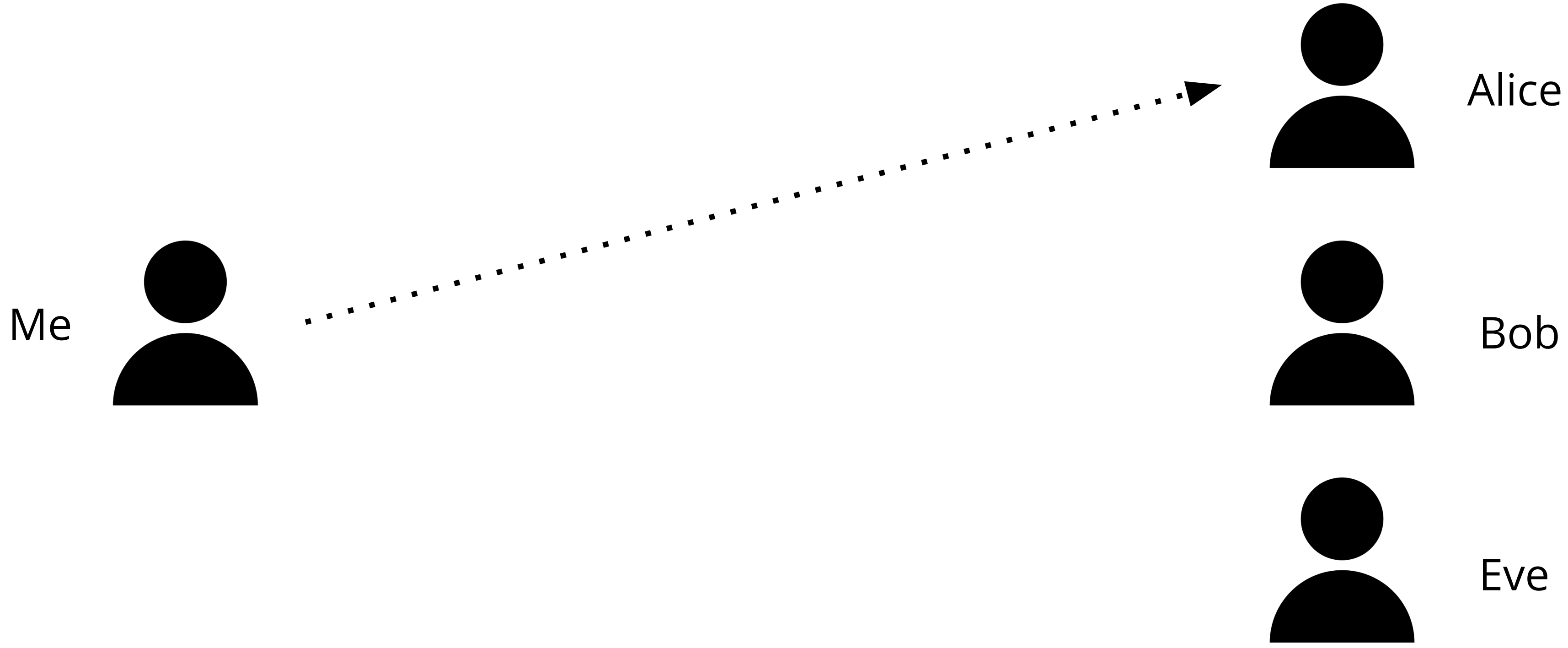
"irreversible" solution = Public + Private Key Pairs

Main Key Pair Attributes:

- Related, but separate (each unique on its own)
- They are unique to each person/user
- When one locks, only the other one can unlock
- Do NOT share private key … ever!

Let’s see how it all work ...

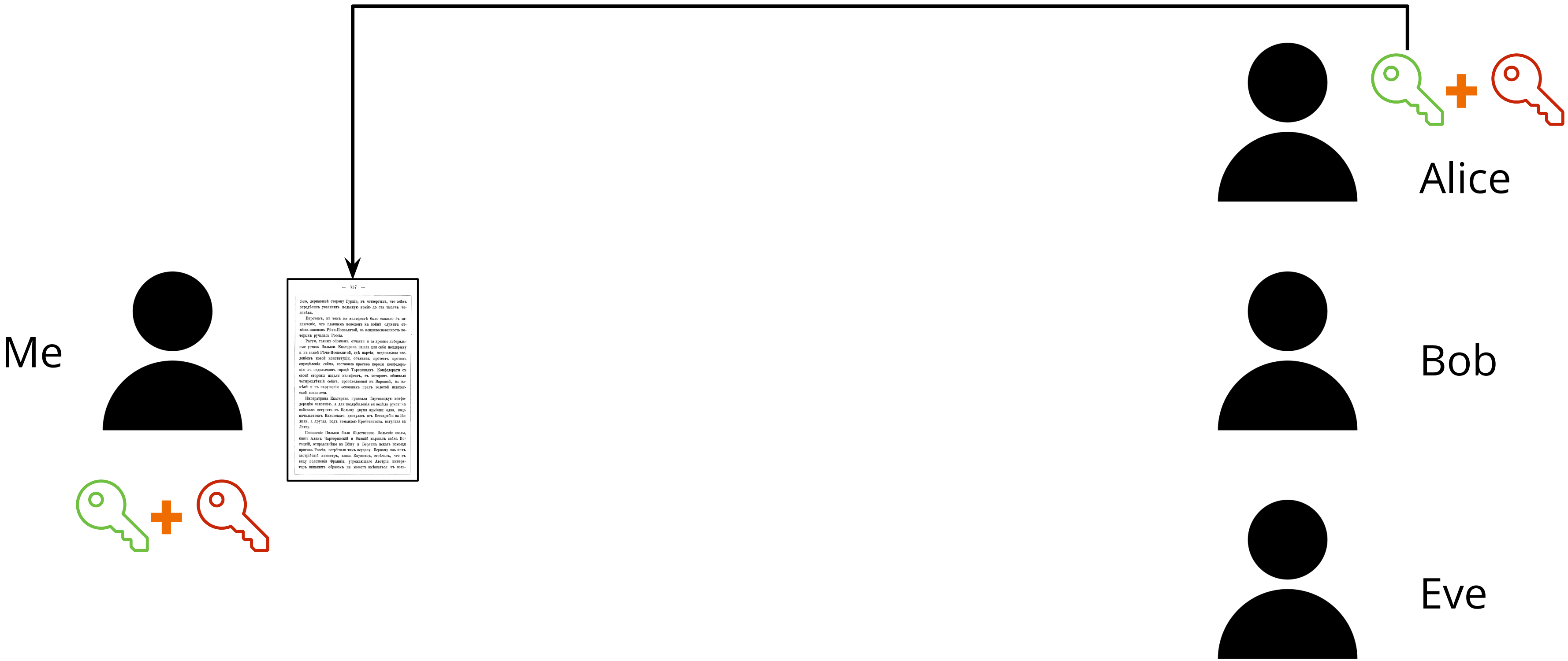
Sending an Encrypted Message with Key Pairs



Sending an Encrypted Message with Key Pairs

Sending an Encrypted Message with Key Pairs

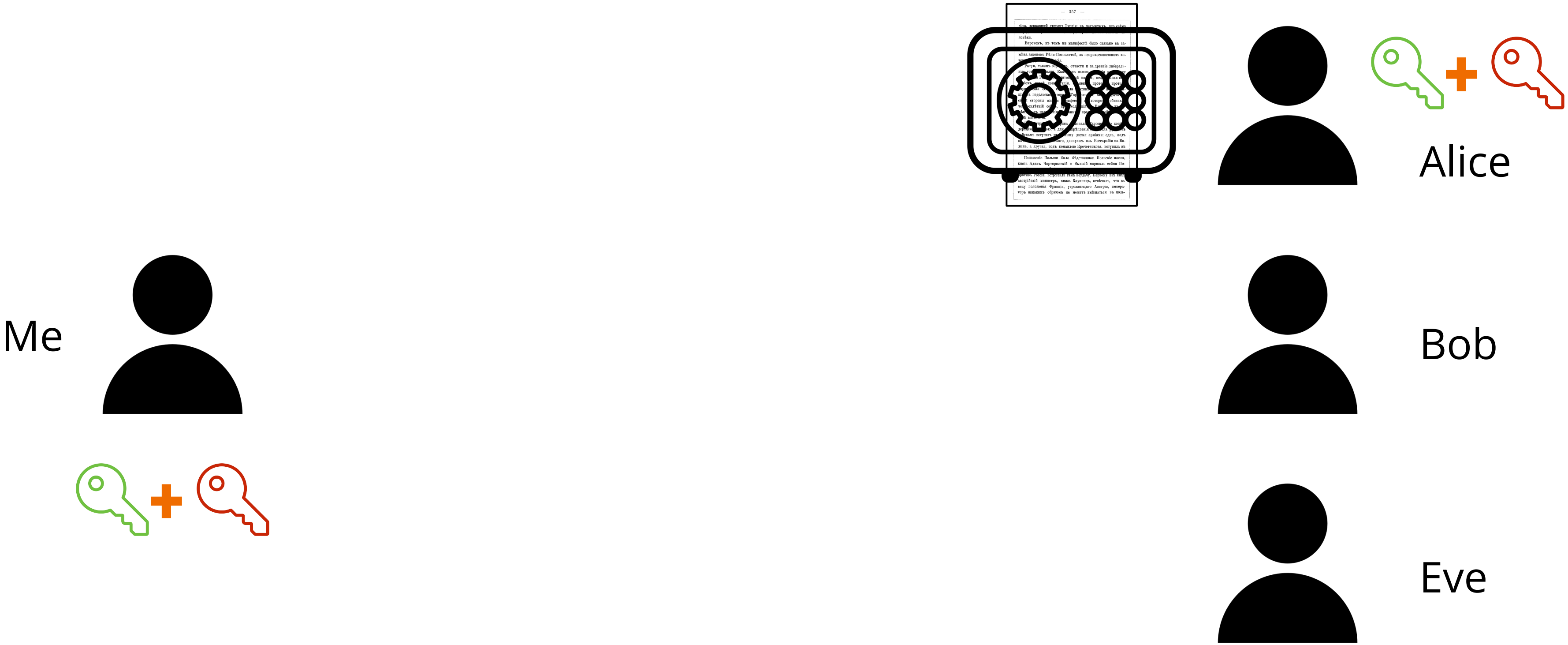
Sending an Encrypted Message with Key Pairs



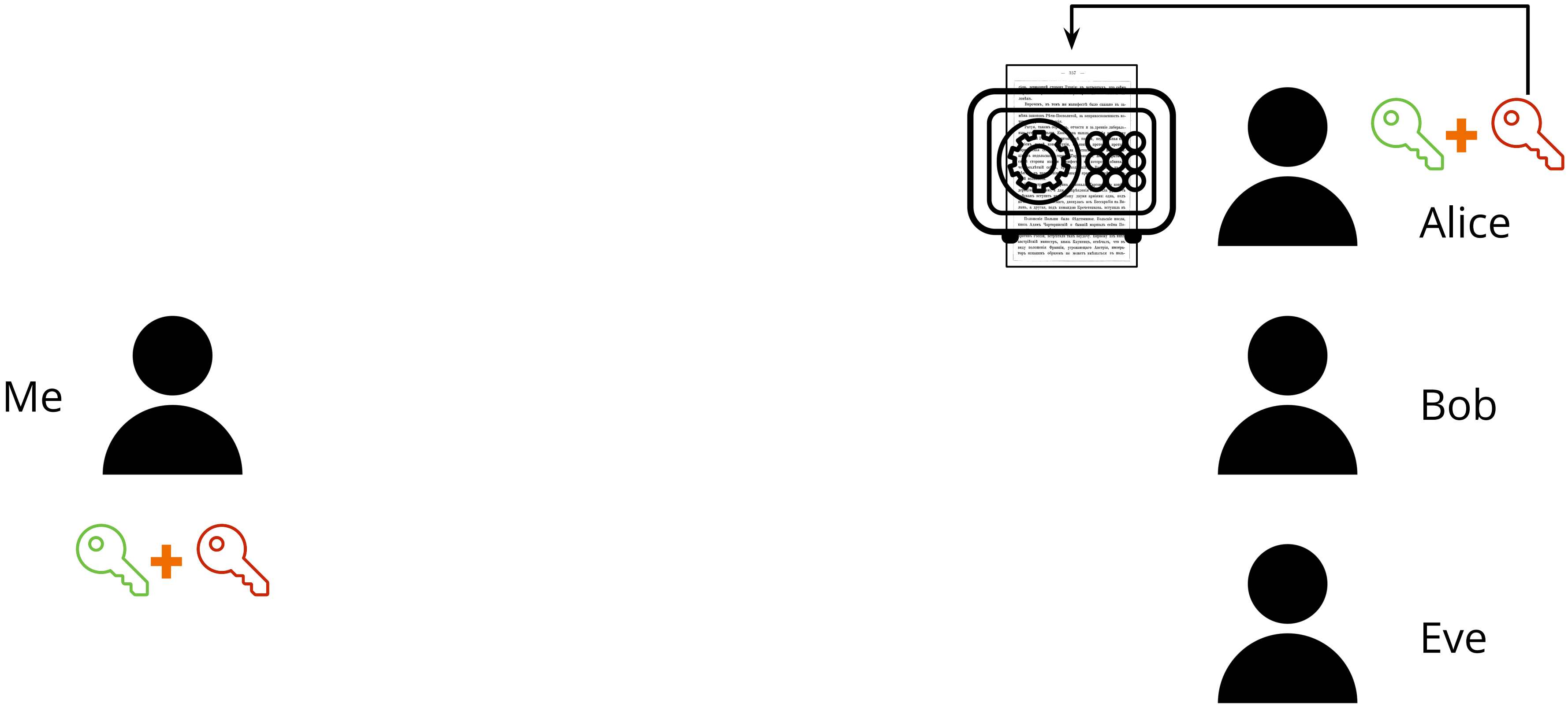
Sending an Encrypted Message with Key Pairs

Sending an Encrypted Message with Key Pairs

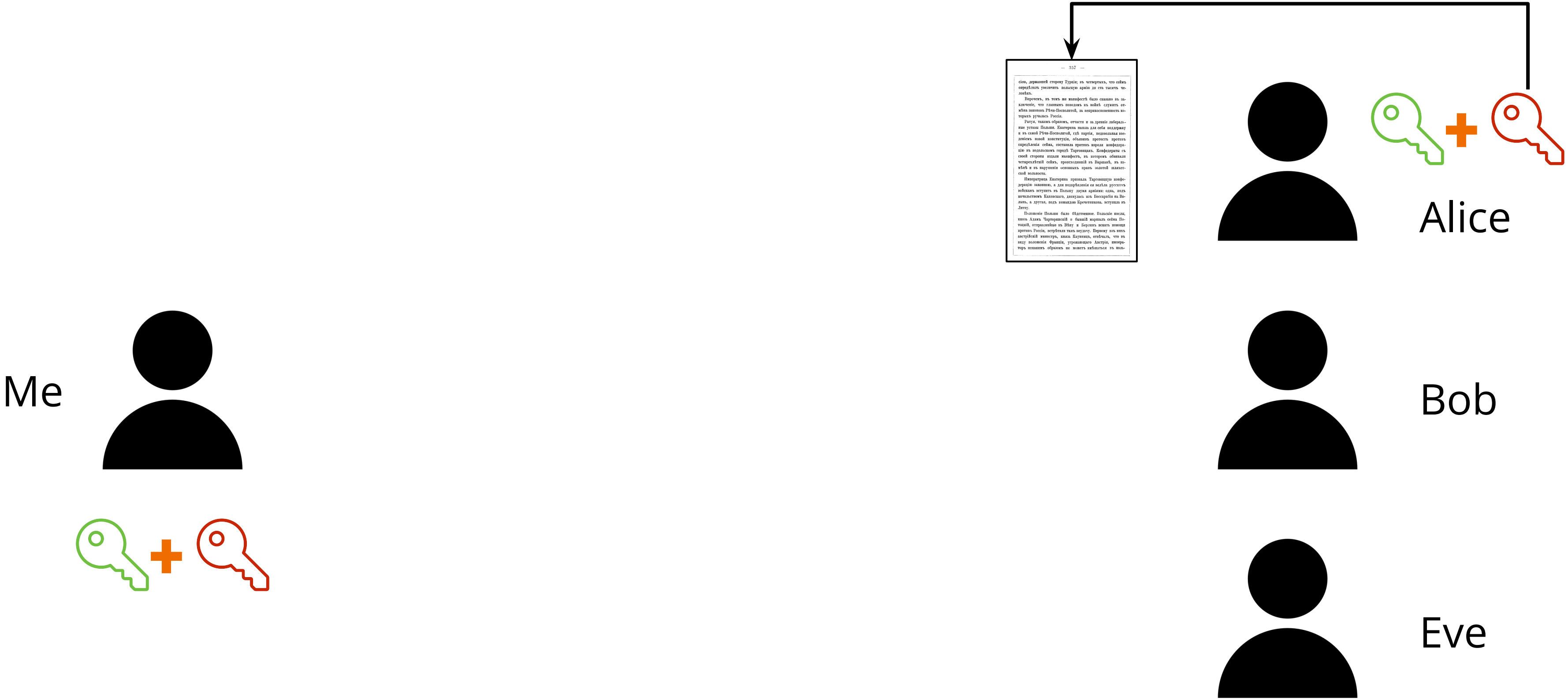
Sending an Encrypted Message with Key Pairs



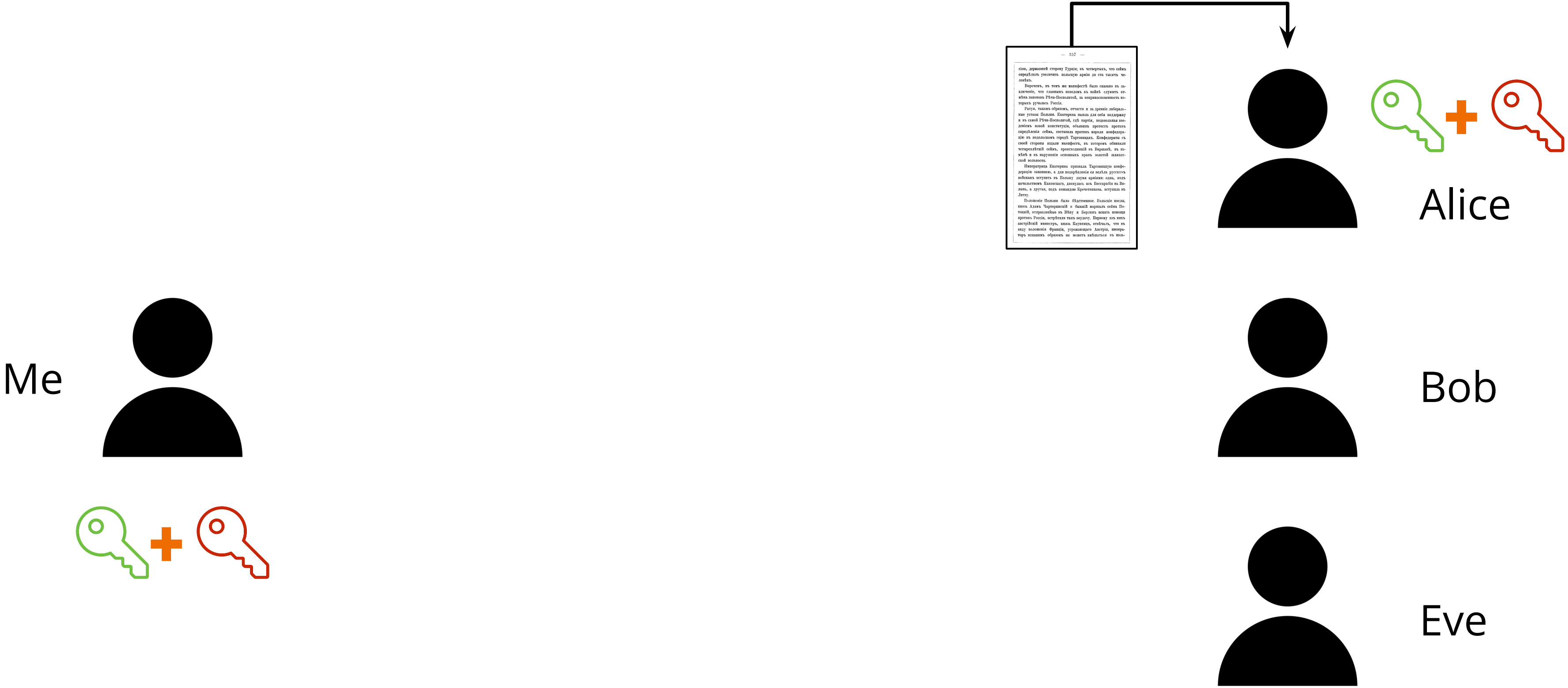
Sending an Encrypted Message with Key Pairs



Sending an Encrypted Message with Key Pairs



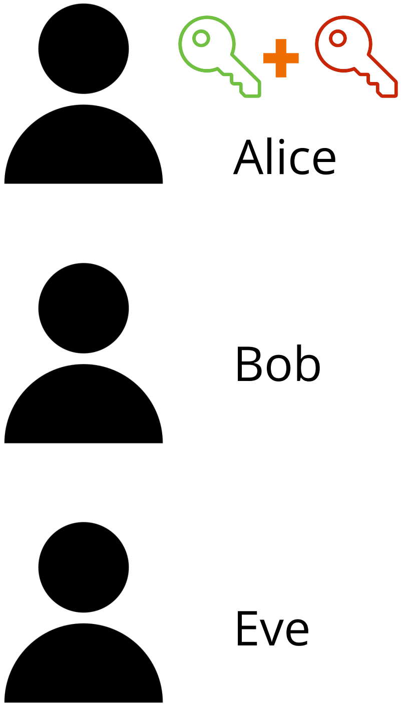
Sending an Encrypted Message with Key Pairs

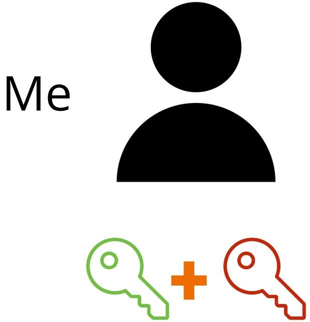


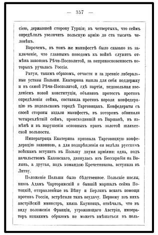
Isn’t that super cool?
But how about the following scenario …

Sending an Encrypted Message with Key Pairs

Sending an Encrypted Message with Key Pairs







— 357 —

сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

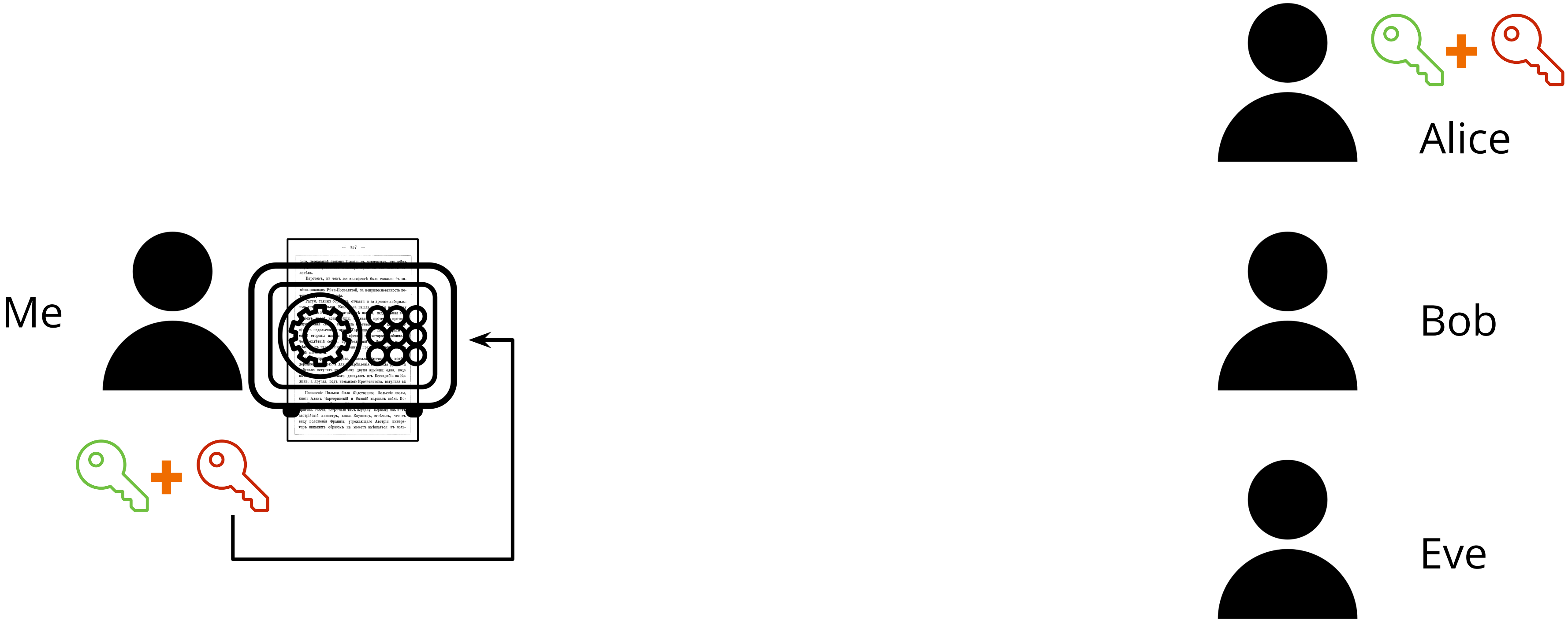
Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауниццъ, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-

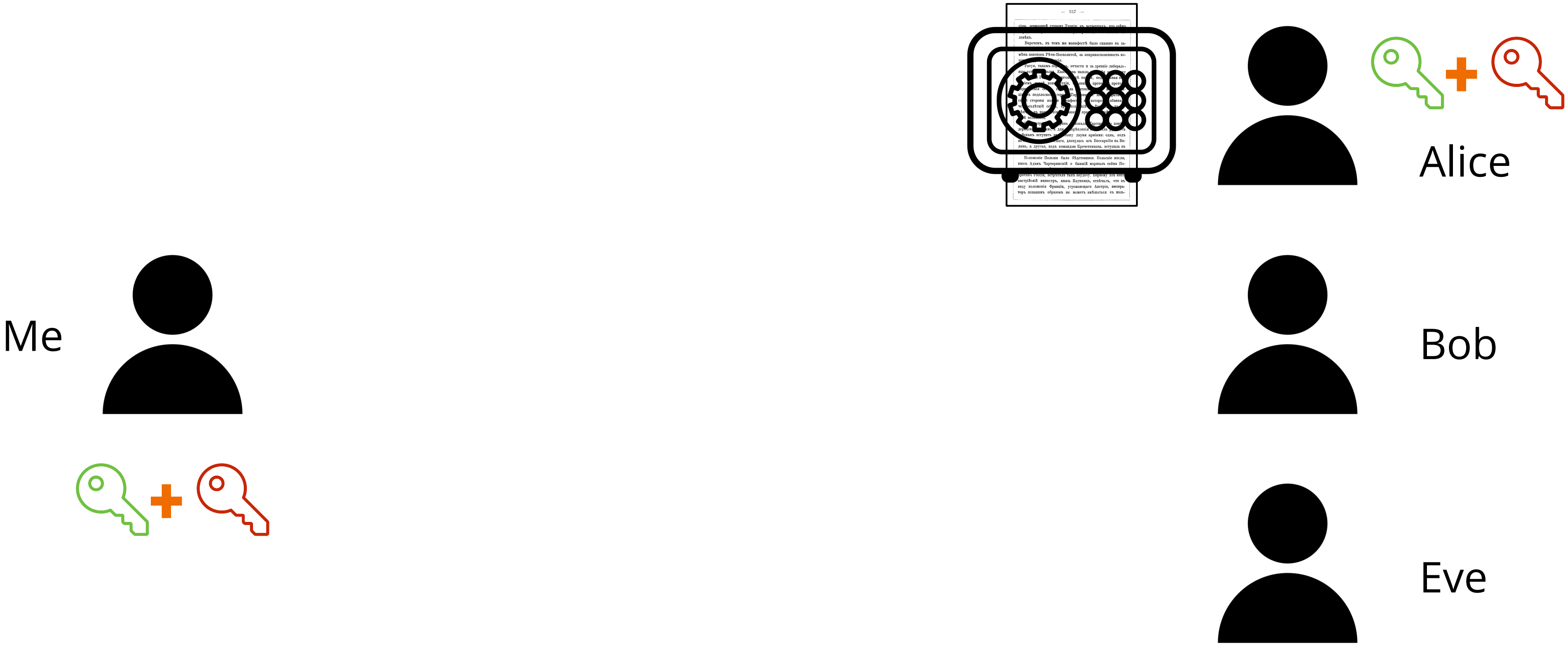
Sending an Encrypted Message with Key Pairs

Sending an Encrypted Message with Key Pairs



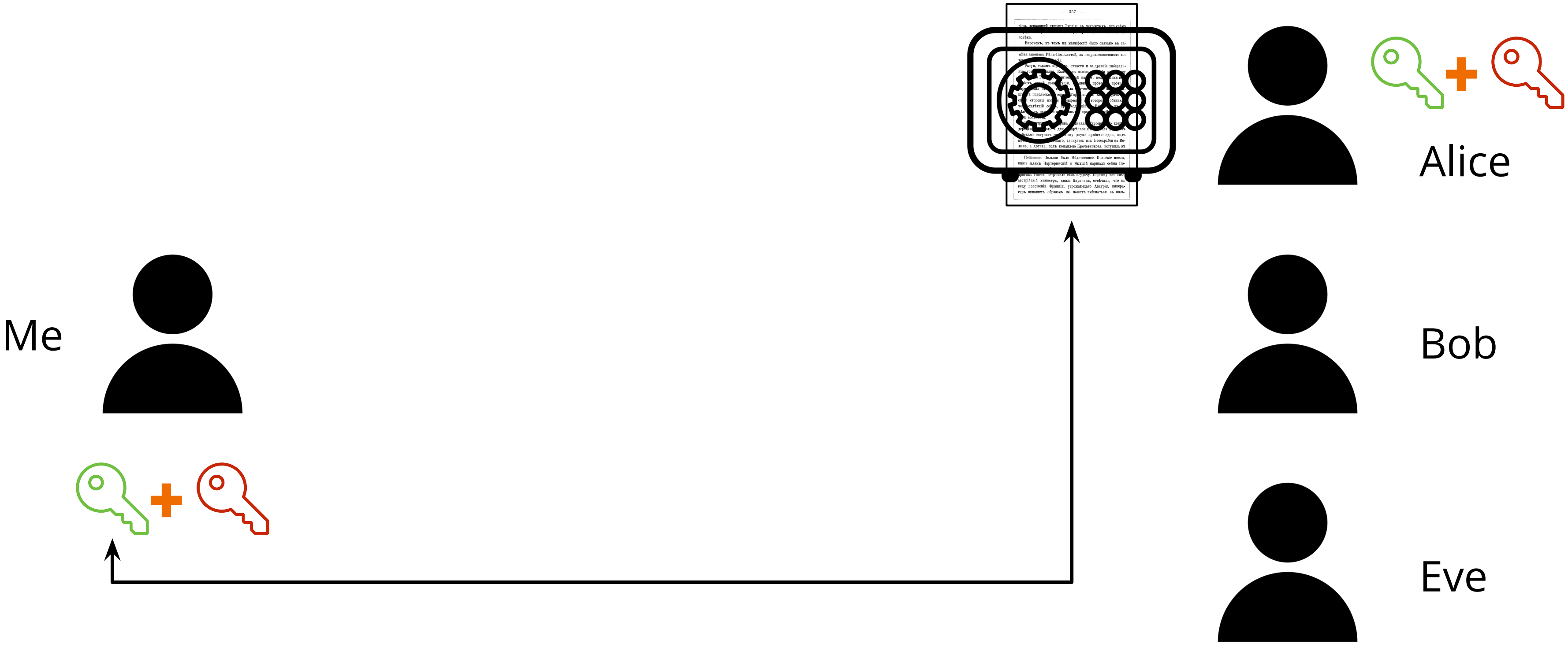
Sending an Encrypted Message with Key Pairs

Sending an Encrypted Message with Key Pairs

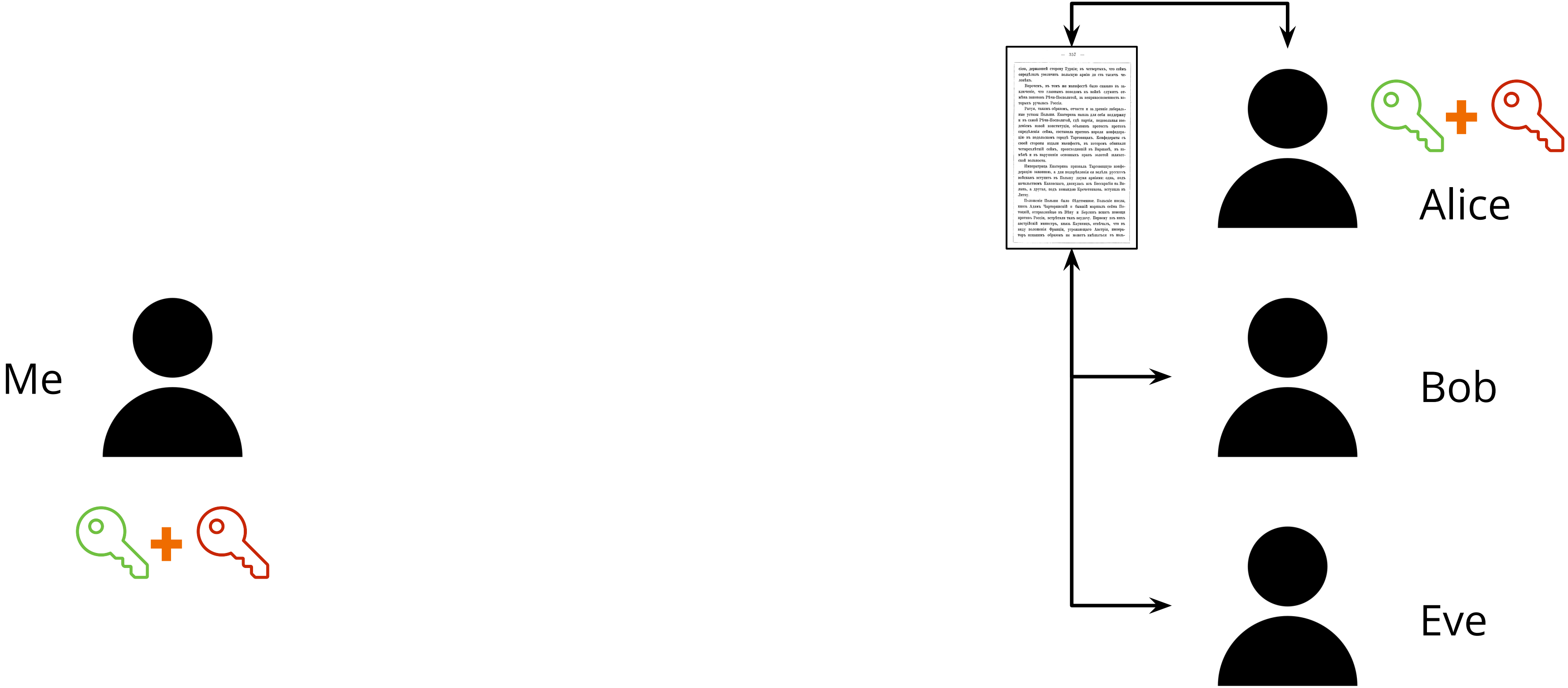


Who can decrypt this message?
What do you need to do it?

Sending an Encrypted Message with Key Pairs



Sending an Encrypted Message with Key Pairs



Who can decrypt this message? EVERYONE
What do you need to do it? MY PUBLIC KEY

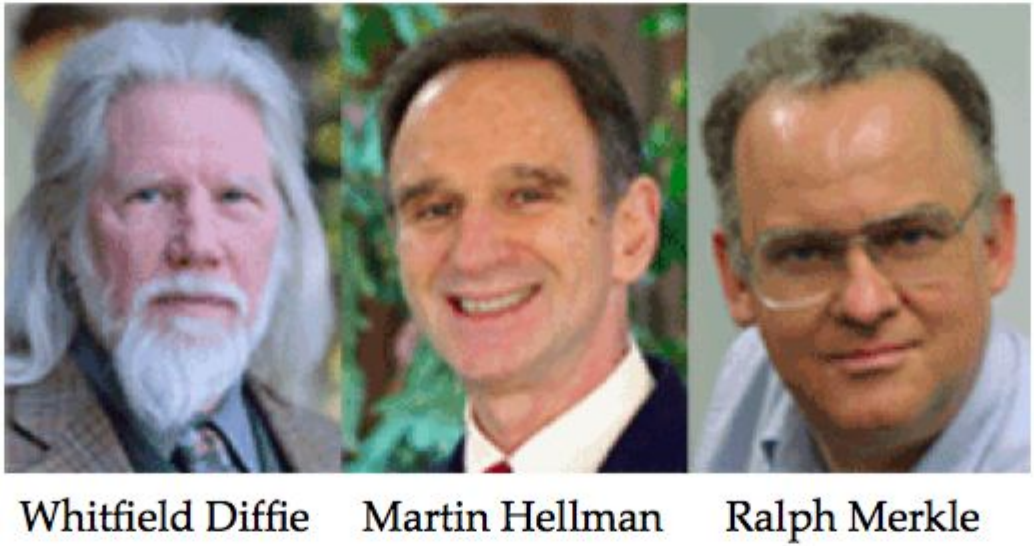
Wouldn’t that be stupid?

Who can decrypt this message? EVERYONE
What do you need to do it? MY PUBLIC KEY

Wouldn't that be stupid?
OR WOULD IT?!

Digital Signatures …
Proving Authorship

Do you know these gentlemen?



Pioneers in Cryptography

Whitfield Diffie Martin Hellman Ralph Merkle

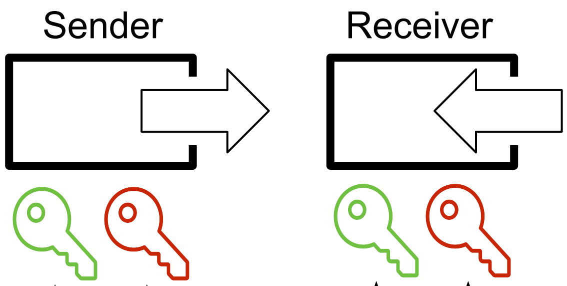
Hellman says of Merkle:

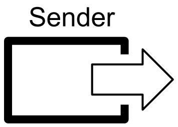
"Ralph, like us, was willing to be a fool, and the way to get to the top of the heap in terms of developing original [thought] is to be a fool, because only fools keep trying. You have idea number 1, you get excited and it flops. Then you have idea number 2, you get excited and it flops. Then you have idea number 99, you get excited and it flops. Only a fool would be excited by the 100th idea, but it might take 100 ideas before one really pays off. Unless you're foolish enough to be continually excited, you won't have the motivation and the energy to carry it through. And God rewards fools.“

Remember safe deposit boxes?

Remember safe deposit boxes?

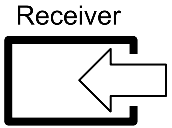
How to encrypt, sign, transmit, and decrypt a msg





1. Compose & Encrypt using Receiver Public Key
2. Sign the Msg using Sender Private Key

TRANSMIT THE MESSAGE



3. Verify Signature by using Sender Public Key
4. Decrypt & Read using Receiver's Private Key

An in-class exercise ...
some simple math ;-)

Multiplying

294 * 992 = ? (by hand)

You have 5 minutes!

Multiplying

294 * 992 = **291,648**

Multiplying vs. Factoring

294 * 992 = 291,648

Now factor **938,081 (by hand)**
You have 10 minutes!

Multiplying vs. Factoring

294 * 992 = 291,648

Now factor **938,081**

<u>1087 * 863</u> (two primes)

Use of Prime Numbers and Modular Arithmetics

There are

1,925,320,391,606,803,968,923

prime numbers below 10^{23} alone

Largest prime number discovered yet …

INDEPENDENT News Voices Sports Culture Indy/Life Video Daily E

News › Science

Largest known prime number discovered with over 23 million digits

Discovery made on computer belonging to electrical engineer who searched for the elusive number for 14 years

Josh Gabbatiss Science Correspondent | @josh_gabbatiss | Friday 5 January 2018 18:00 | 2 comments

If two plus three equals five (2+3=5) and two plus eleven equals one (2+11=1), then what is five plus eleven? (5+11=??)

Hint …


BREITLING
MPH
KM
BASE
1000M
NAUT.
STAT.

If two plus three equals five (2+3=5) and
two plus eleven equals one (2+11=1),
then what is five plus eleven? (5+11=??)

Let’s review some (simple) math … sorry!!

A few words on (math) functions

$$f(x) = x^2 + 8$$

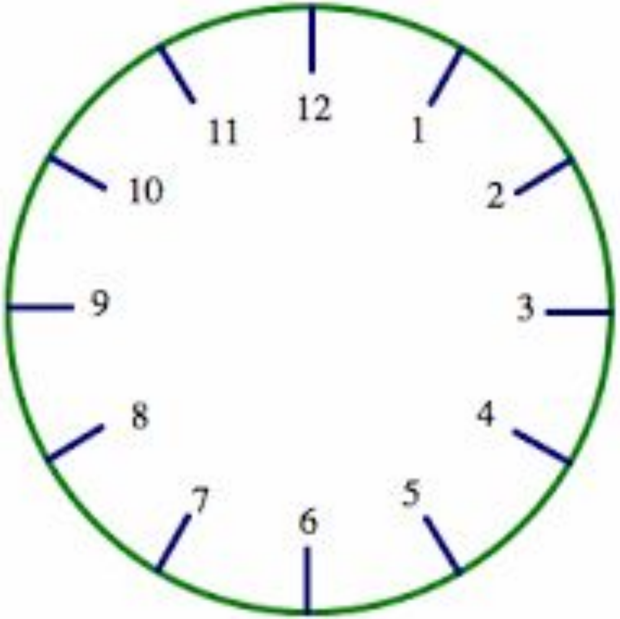
Functions in Math

- Simply put, a function is a (mathematical) operation ...
- ... one input equals to one output
- f(x) where x is the input value
- Example:
  - our function is “Doubling” →
  - $f(x) = 2x$ →
  - Take an input, then double it (or multiply by 2)
  - For x=**4 (i.e. input is 4)**, then the **output is 8**
- But then a funny thing happens ...

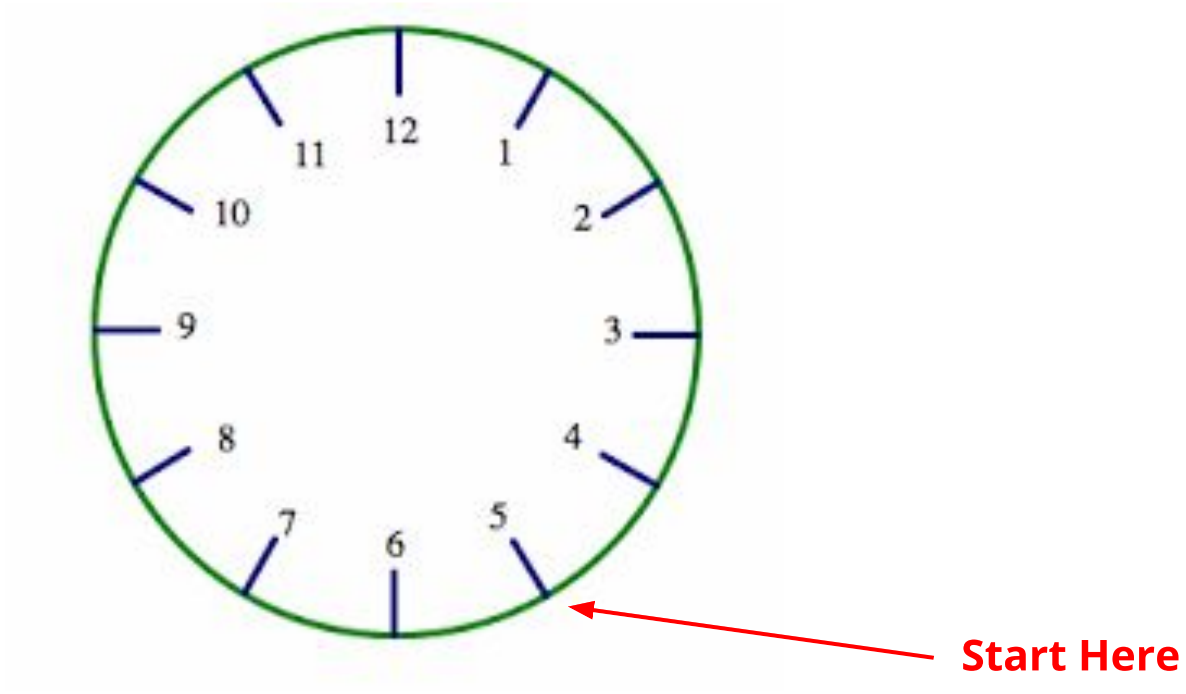
Functions in Math

- But then a funny thing happens ...
- ... our function is still “Doubling” →
- So what if I give you the output only? Can you figure out the input?
- OF COURSE ... we’ll just reverse the function
- Example:
    - our function is “Doubling” →
    - $f(x) = 2x$ →
    - If the output is **44**, then the input is ...
    - **22** ;-)
- Most functions in math are Two-way Functions (reversible)
- But then ...

Modular (or clock) Arithmetics


12
1
2
3
4
5
6
7
8
9
10
11

Start at 5, then jump 11 units ...



If (2+3=5) and (2+11=1),
then (5+11=<u>4</u>)

$$2+3 = 5 \pmod{12}$$
$$2+11 = 1 \pmod{12}$$

$$5+11 = 4 \pmod{12}$$

Let's calculate 11x9 (mod 13)=?

Let’s calculate 11x9 (mod 13)=?

First, let’s use “regular” math: 11x9=99

Then, let’s divide: 99÷13 = 7, with remainder 8

…

So

11x9 = 8 (mod 13)

In-Class Exercise (you can use calculators only)

for x =	1	2	3	4	5	6
3^x						
$3^x \pmod 7$						

Homework for Next Class (can use calculator

for x =	1	2	3	4	5	6
3^x	**3**	**9**	**27**	**81**	**243**	**729**
$3^x \pmod 7$	**3**	**2**	**6**	**4**	**5**	**1**

Let's consider this special one-way function …

$Y^x \pmod{P}$ … with $Y < P$ as two prime numbers

Y and P are NOT secrets and can be shared

Our function is $Y^x \pmod{P}$ … with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Our function is Y^x (mod P) ... with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. 3)

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. 6)

Our function is $Y^x \pmod{P}$ ... with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. 3)

Plug 3 as X into our function to get ***α***, so $7^3 \pmod{11} \rightarrow 343 \pmod{11} = 2$

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. 6)

Plug 6 as X into our function to get **β**, so $7^6 \pmod{11} \rightarrow 117{,}649 \pmod{11} = 4$

Our function is Y^X (mod P) ... with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. 3)

Plug 3 as X into our function to get ***α***, so 7^3 (mod 11) → 343 (mod 11) = 2

Send ***α*** (or 2) to Bob

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. 6)

Plug 6 as X into our function to get **β**, so 7^6 (mod 11) → 117,649 (mod 11) = 4

Send **β** (or 4) to Alice

Our function is Y^x (mod P) … with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. 3)

Plug 3 as X into our function to get α, so 7^3 (mod 11) → 343 (mod 11) = 2

Send α to Bob

Plug β into β^A (mod 11) → 4^3 (mod 11) → 64 (mod 11) = **9**

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. 6)

Plug 6 as X into our function to get β, so 7^6 (mod 11) → 117,649 (mod 11) = 4

Send β to Alice

Plug α into α^B (mod 11) → 2^6 (mod 11) → 64 (mod 11) = **9**

Our function is Y^X (mod P) ... with $Y<P$

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. 3)

Plug 3 as X into our function to get ***α***, so 7^3 (mod 11) → 343 (mod 11) = 2

Send ***α*** to Bob

Plug **β** into β^A (mod 11) → 4^3 (mod 11) → 64 (mod 11) = **9**

How cool! Alice has the same KEY as Bob without exchange of the actual key!

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. 6)

Plug 6 as X into our function to get **β**, so 7^6 (mod 11) → 117,649 (mod 11) = 4

Send **β** to Alice

Plug ***α*** into α^B (mod 11) → 2^6 (mod 11) → 64 (mod 11) = **9**

How cool! Bob has the same KEY as Alice without exchange of the actual key!

If you are Eve (snooper), can you figure out the key?

Alice

Agree & share on Y & P (e.g. Y=7 & P=11)

Alice picks a secret number **A** (e.g. [redacted])

Plug [redacted] as X into our function to get ***α***, so 7[redacted] (mod 11) = 2

Send ***α*** to Bob

Plug **β** into β^A (mod 11) → 4[redacted] (mod 11) = [redacted]

What is the KEY? Eve knows the function, Y, P [7^x (mod 11)] and both ***α*** and **β**, but neither A nor B!

Bob

Agree & share on Y & P (e.g. Y=7 & P=11)

Bob picks a secret number **B** (e.g. [redacted])

Plug [redacted] as X into our function to get **β**, so 7[redacted] (mod 11) = 4

Send **β** to Alice

Plug ***α*** into α^B (mod 11) → 2[redacted] (mod 11) = [redacted]

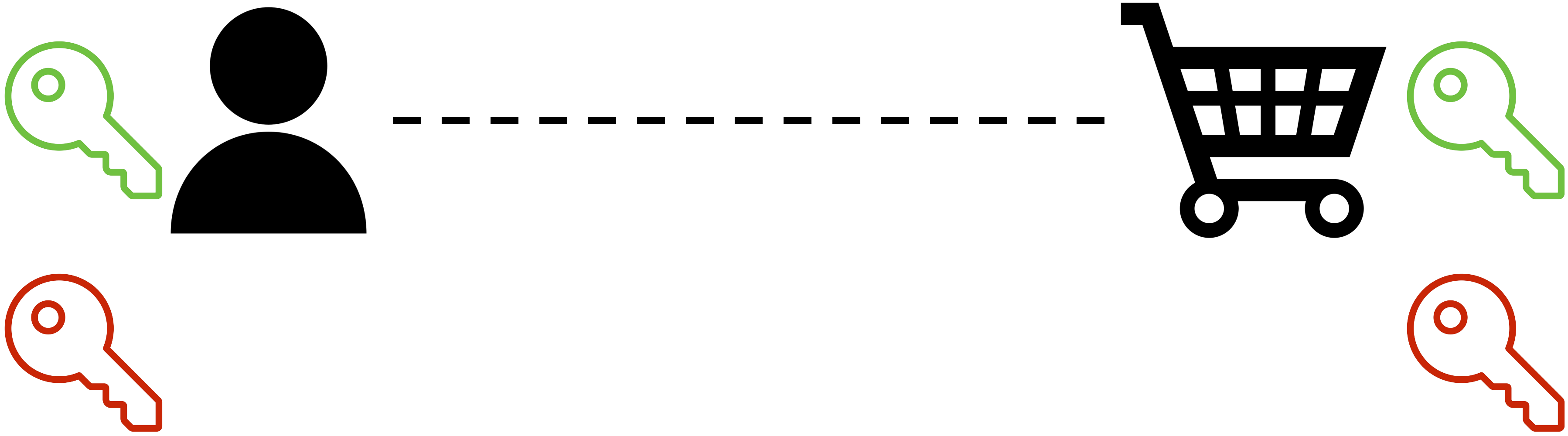
What is the KEY? Even knows the function, Y, P [7^x (mod 11)] and both ***α*** and **β**, but neither A nor B!

We'll come back to one-way functions later on,

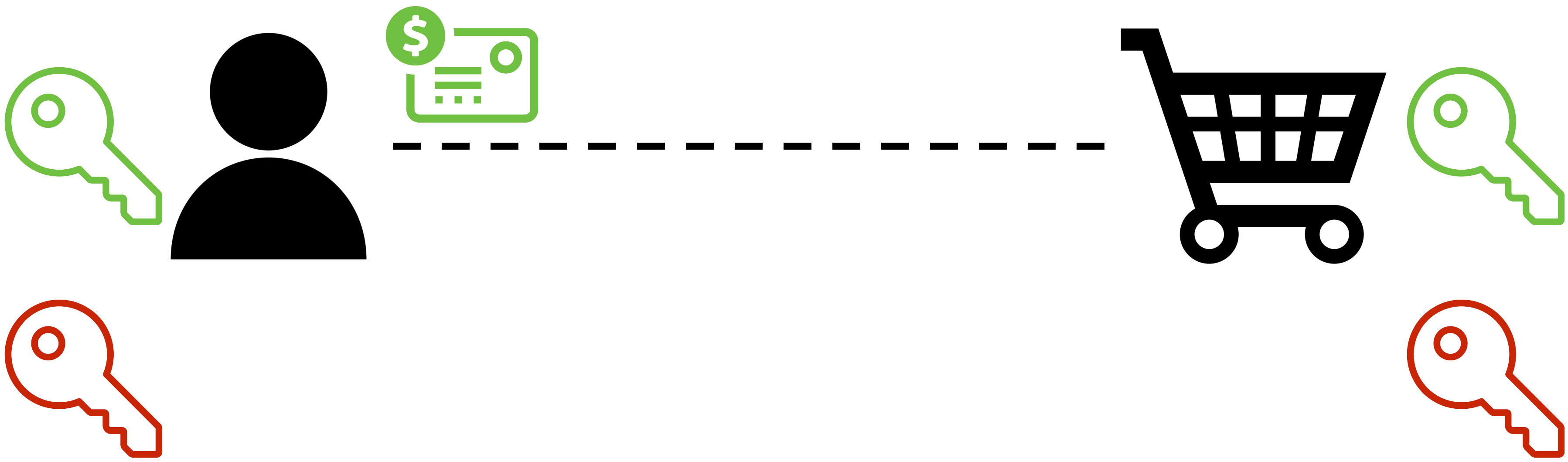
… so stay tuned ;-)

Now back to our Public-Private Key Pair model

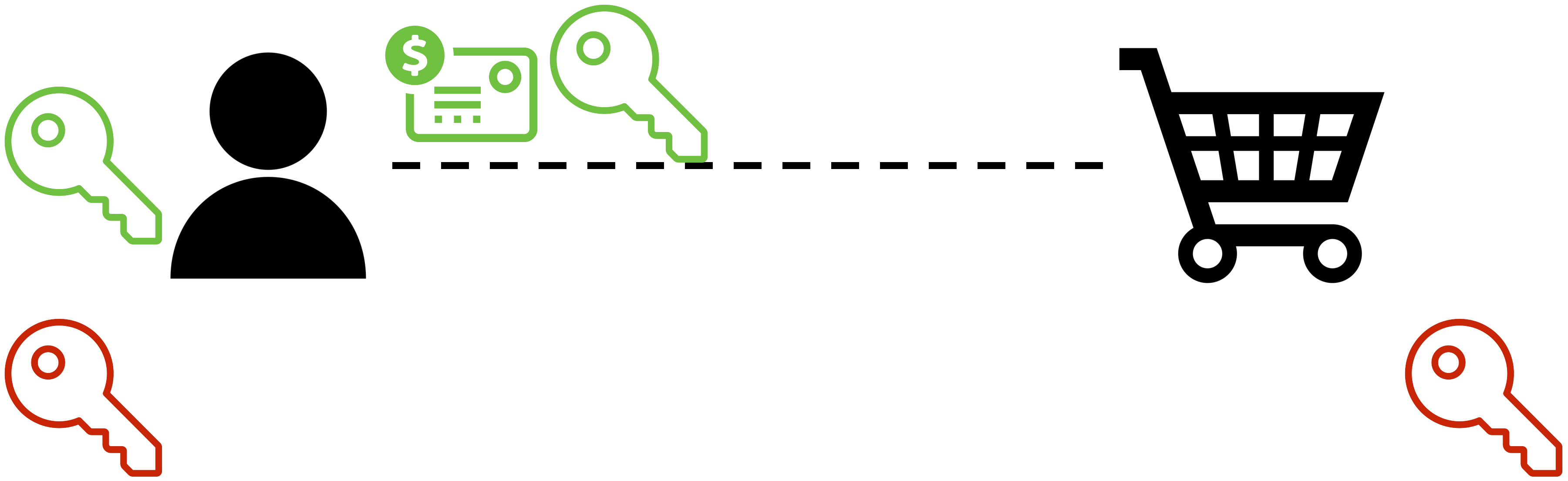
Public Key Encryption



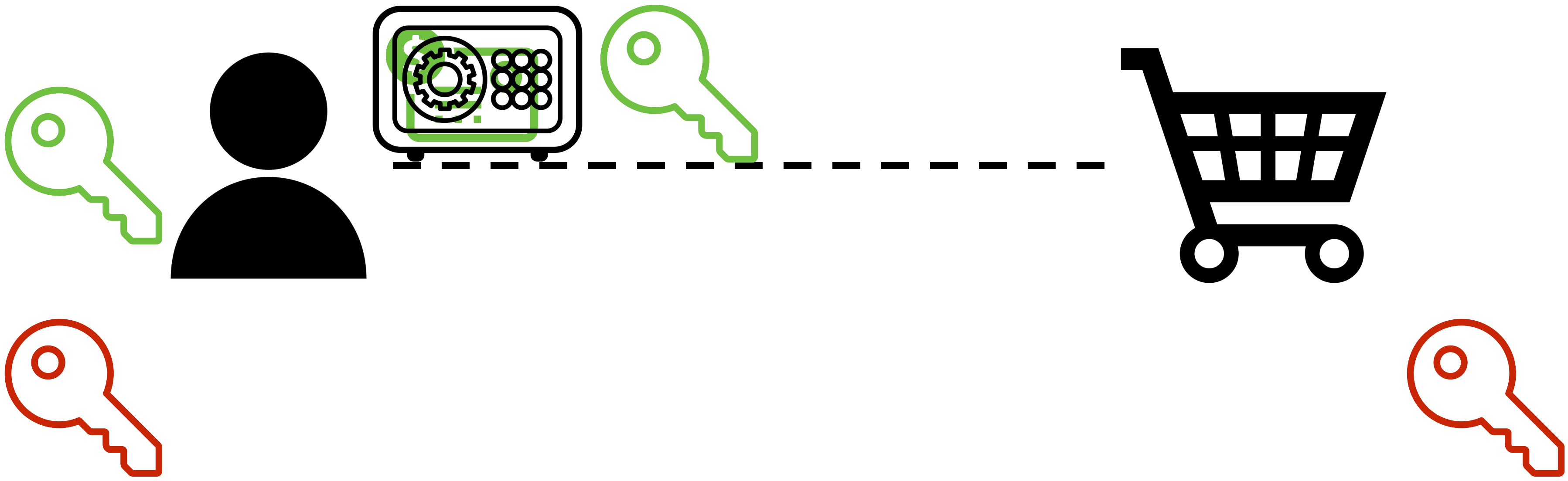
Public Key Encryption



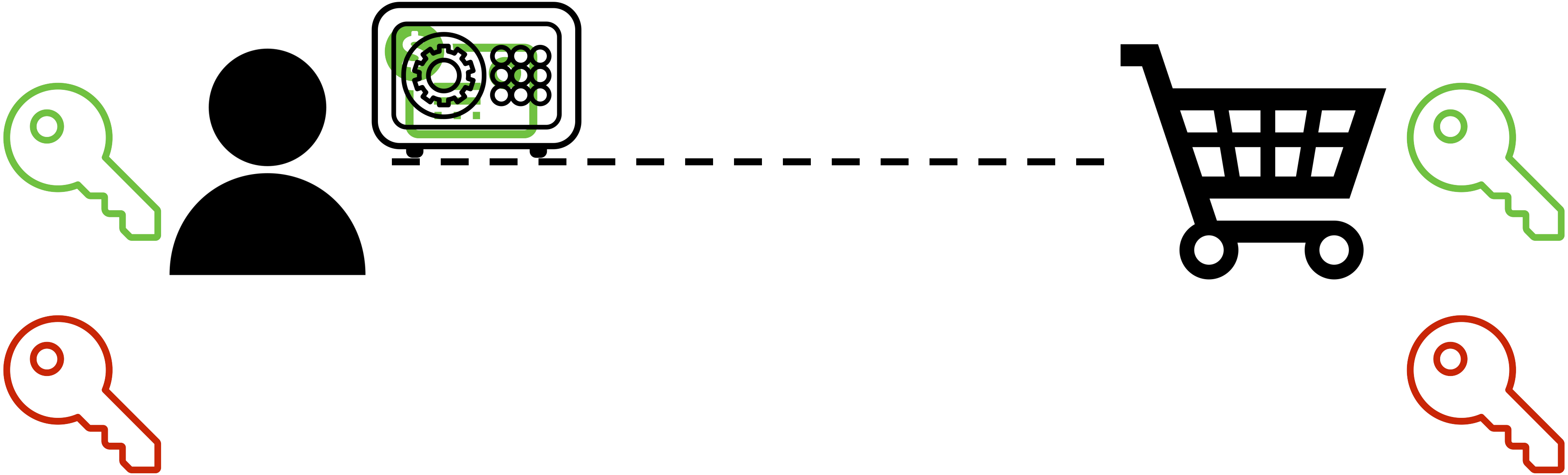
Public Key Encryption



Public Key Encryption



Public Key Encryption



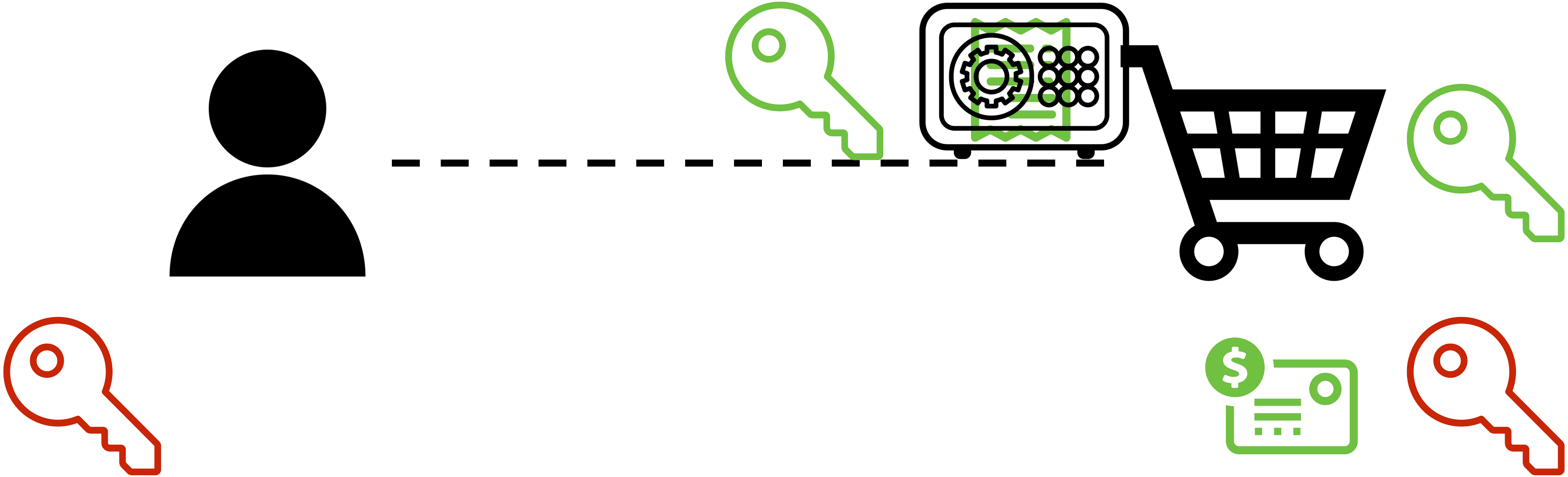
Public Key Encryption

Public Key Encryption

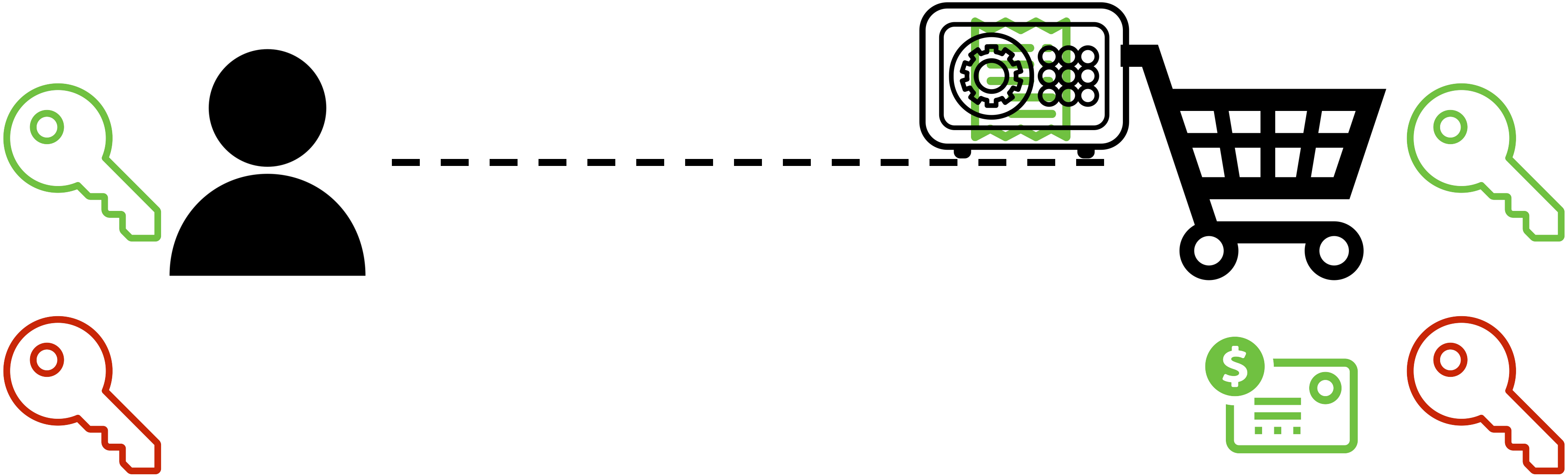
Public Key Encryption

Public Key Encryption

Public Key Encryption



Public Key Encryption

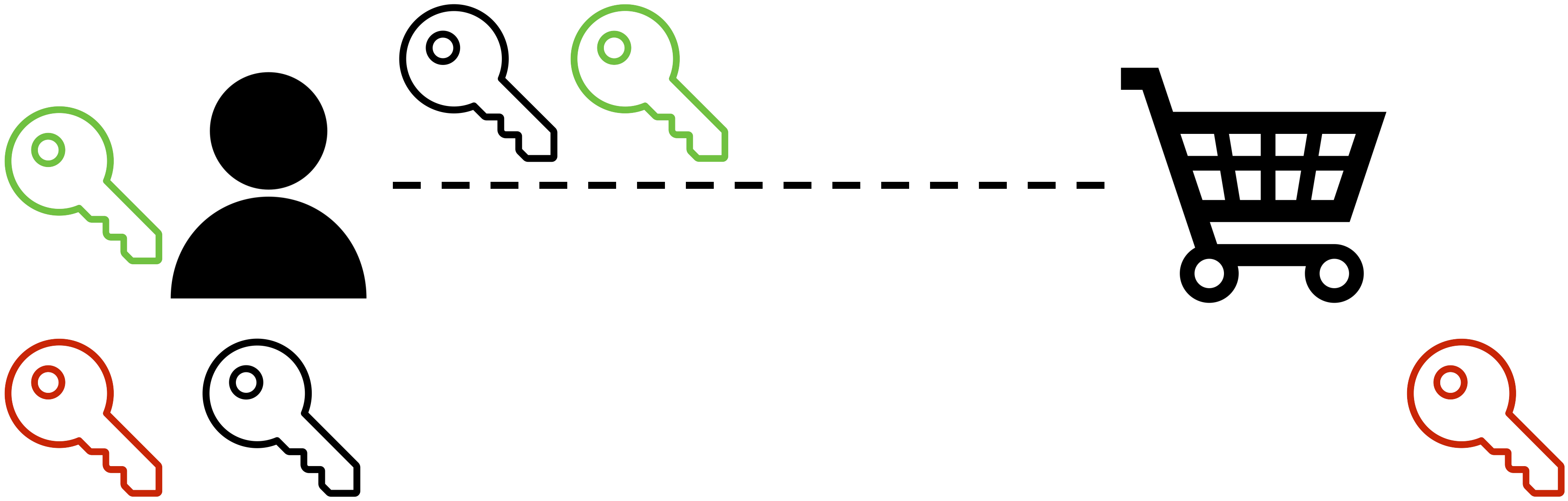


Public Key Encryption

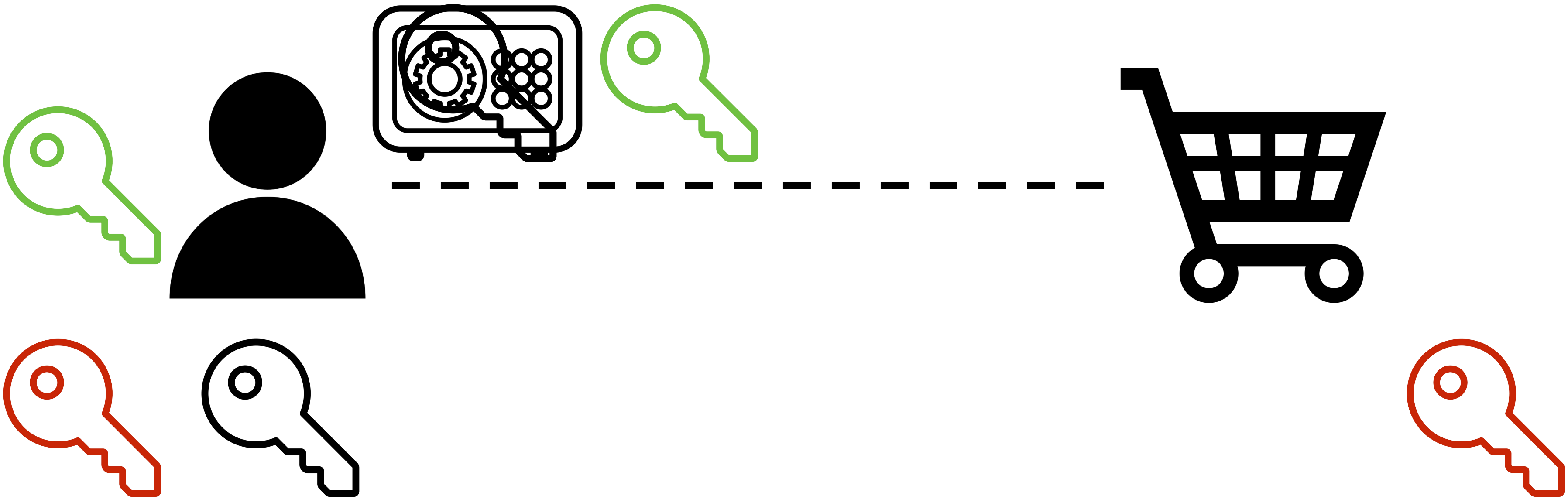
Public Key Encryption

Public Key Encryption: Ease Computational Reqs.

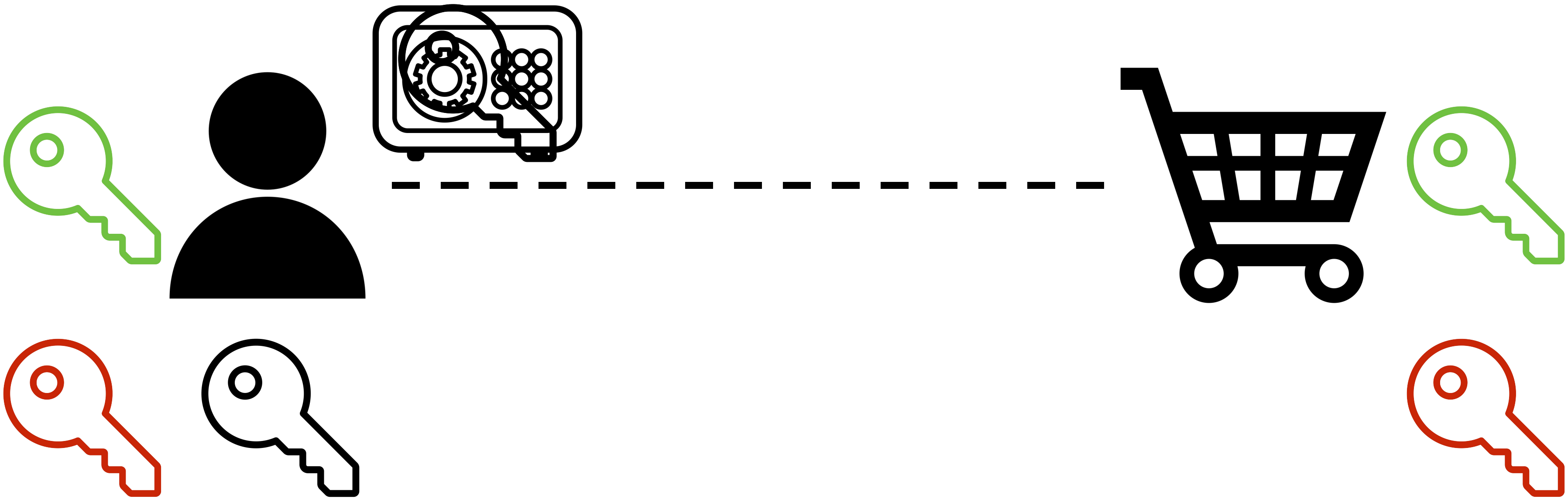
Public Key Encryption



Public Key Encryption



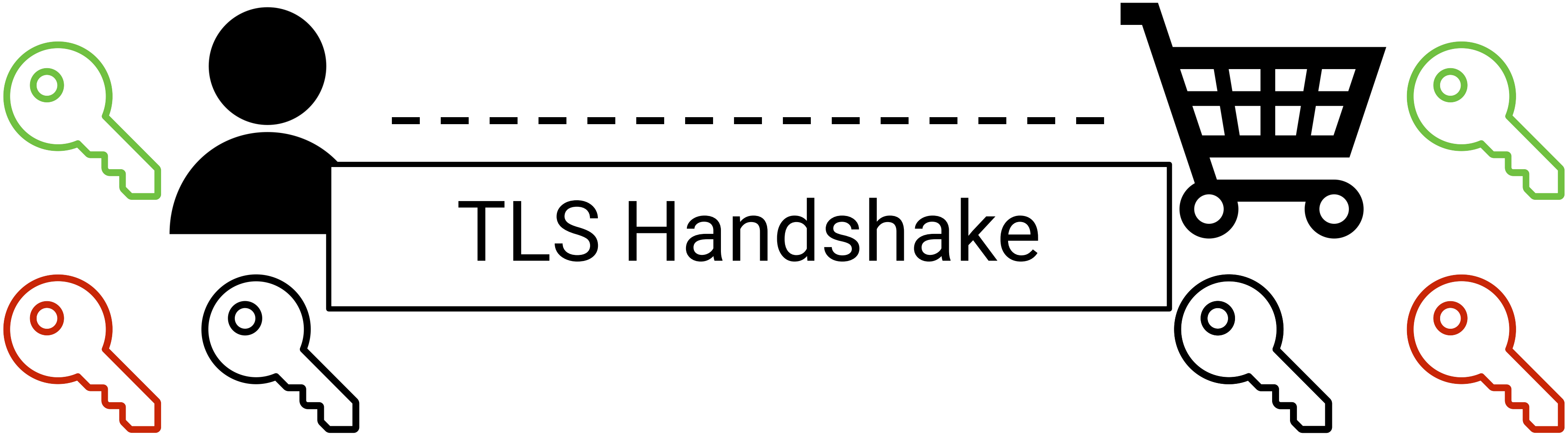
Public Key Encryption



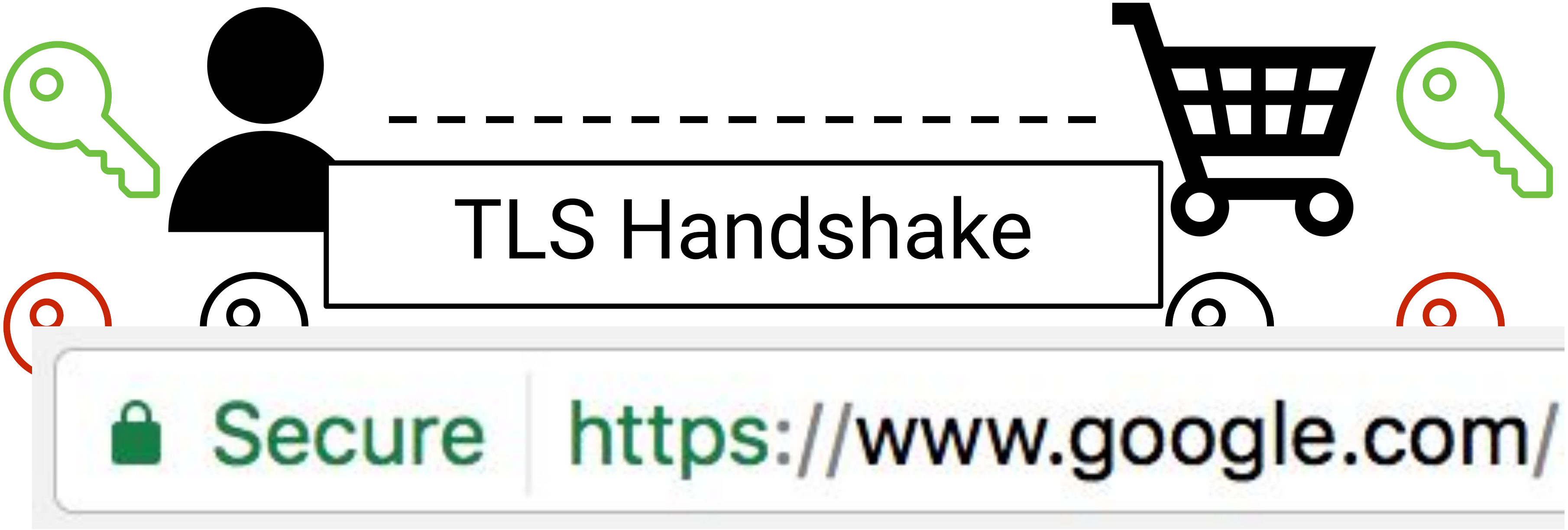
Public Key Encryption

Public Key Encryption

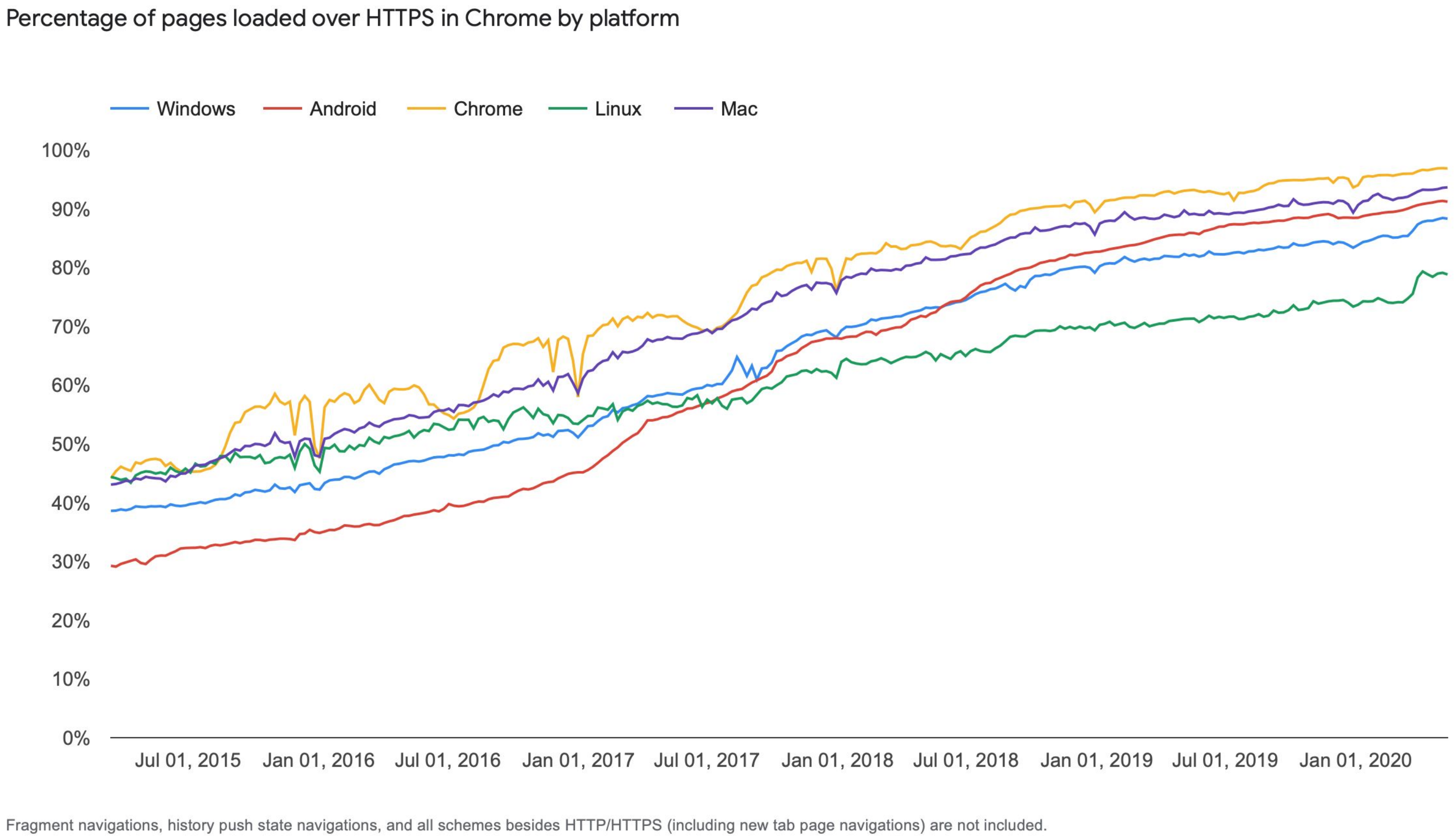
Public Key Encryption



Public Key Encryption



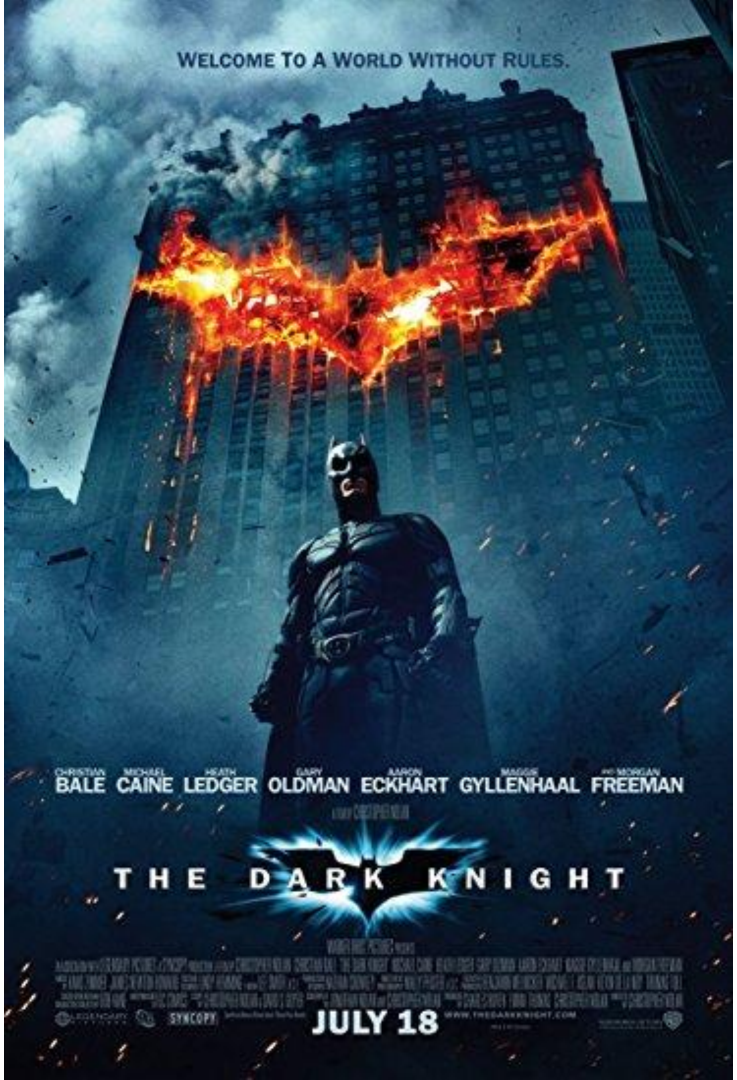
HTTPS Adoption (link)

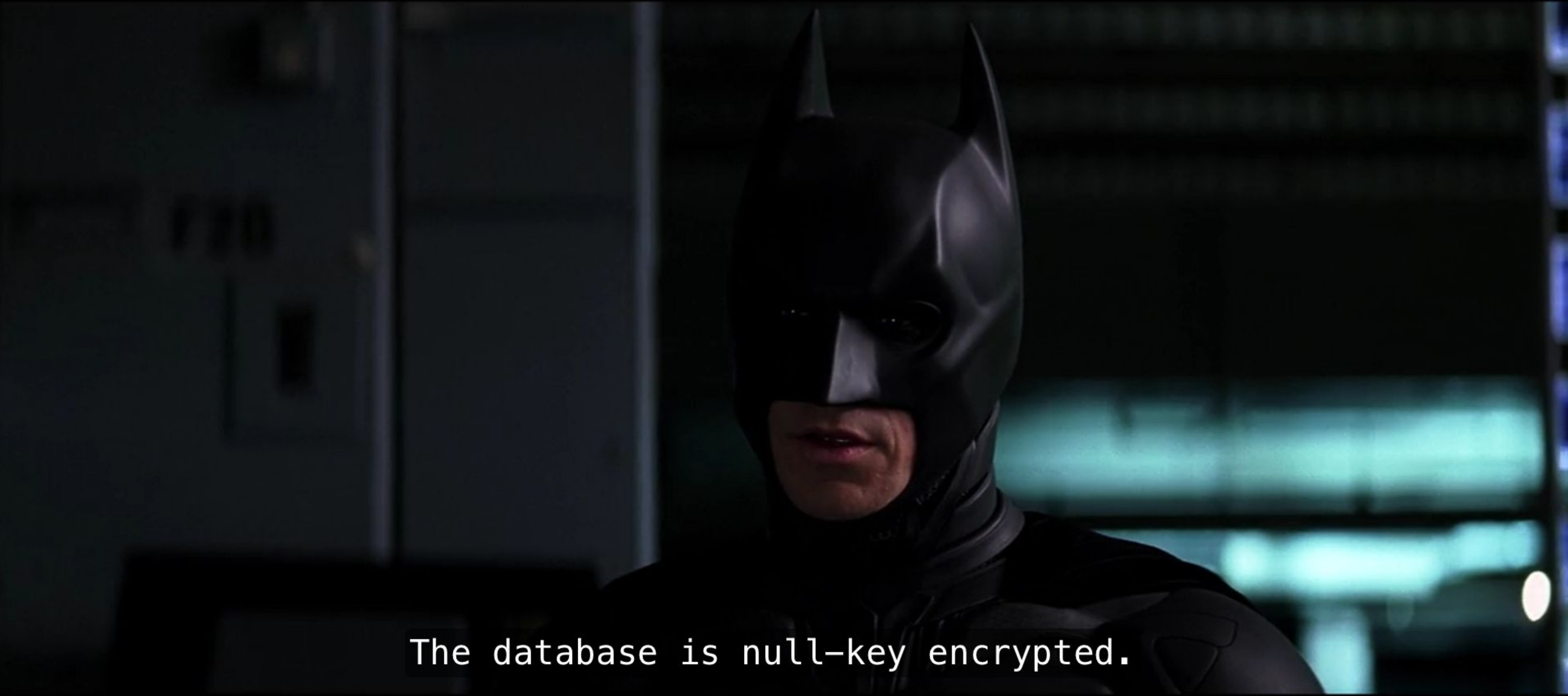


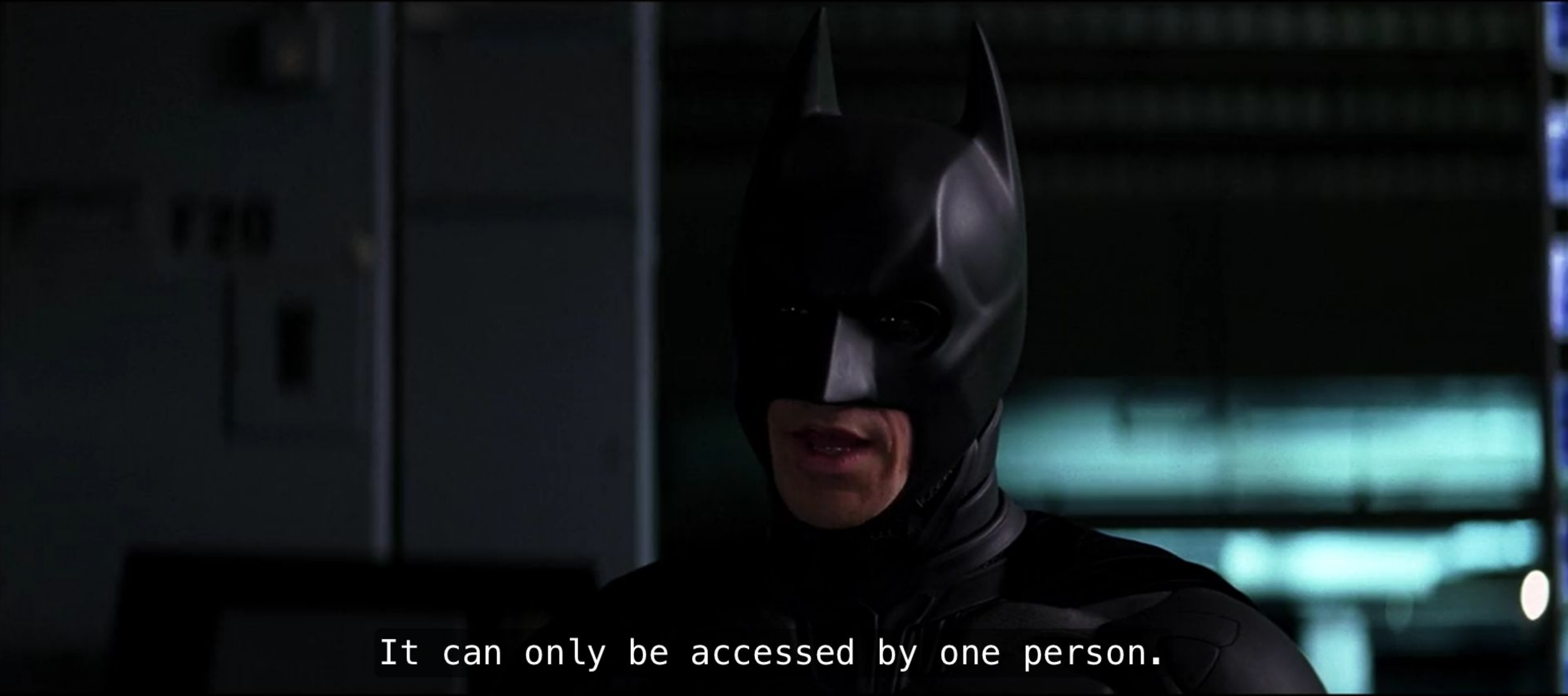
A little fun learning … with movies!


ALL PAPER
TRASH

One of the greatest movies of all time ...
if not THE greatest is ...

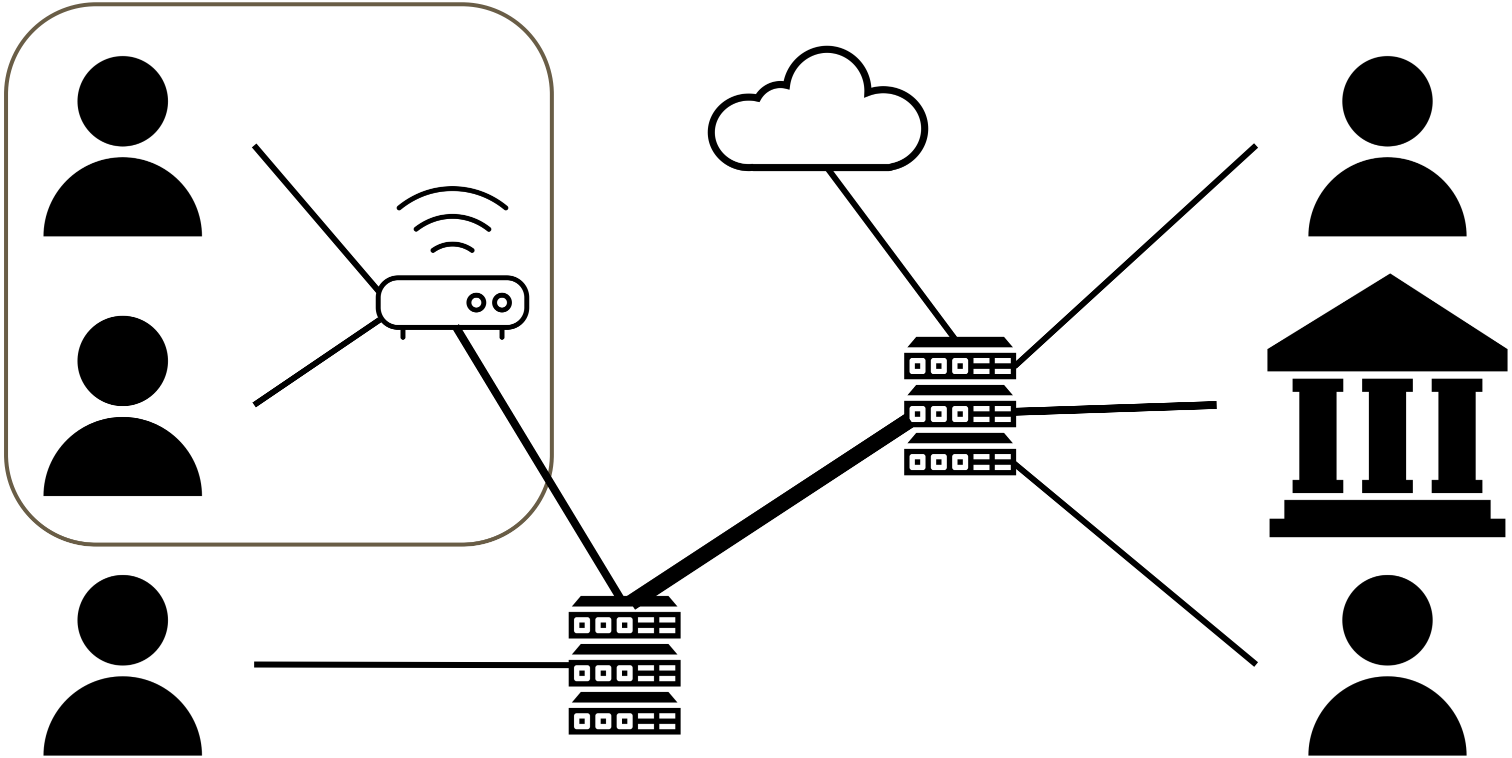

WELCOME TO A WORLD WITHOUT RULES.
CHRISTIAN BALE
MICHAEL CAINE
HEATH LEDGER
GARY OLDMAN
AARON ECKHART
MAGGIE GYLLENHAAL
AND MORGAN FREEMAN
THE DARK KNIGHT
WWW.THEDARKKNIGHT.COM
JULY 18


The database is null-key encrypted.

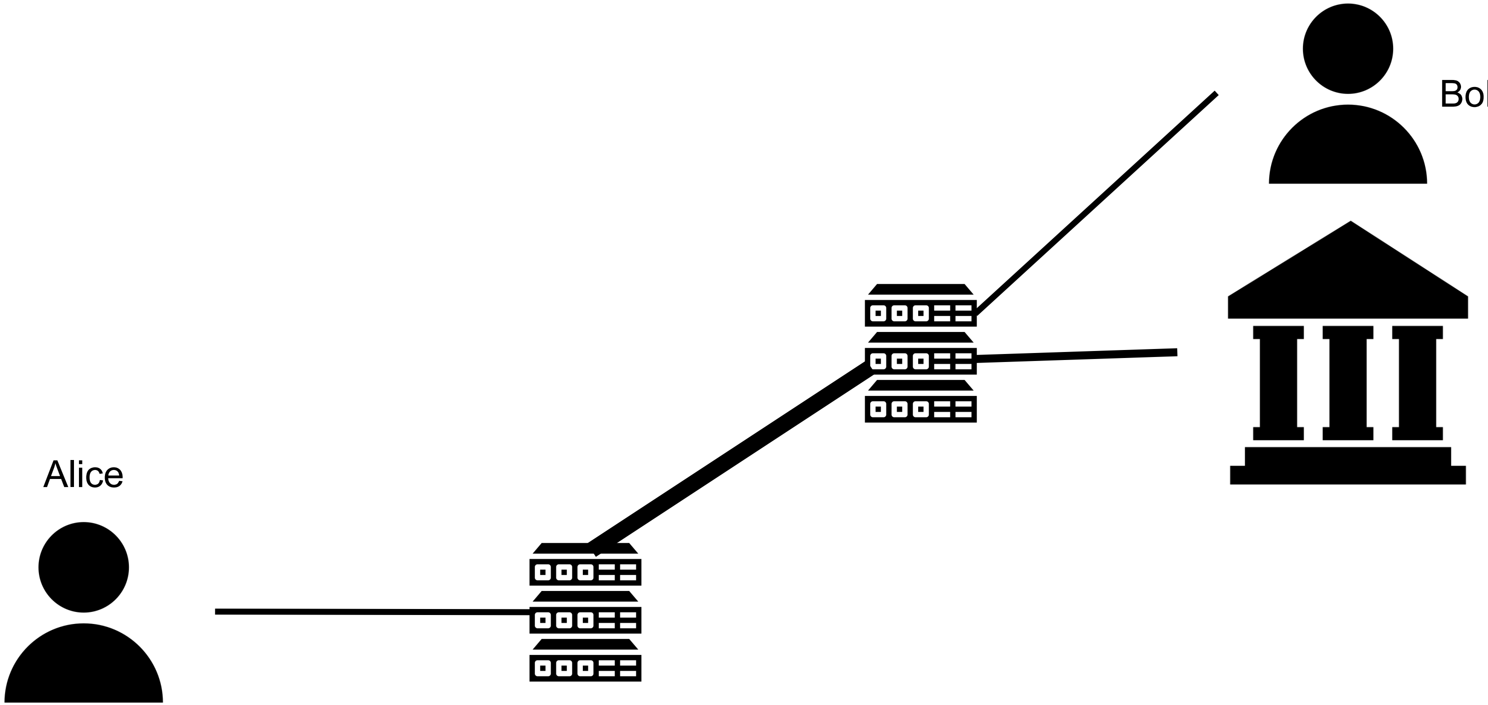

It can only be accessed by one person.

Aren’t movies fun?!
Back to our impostor …

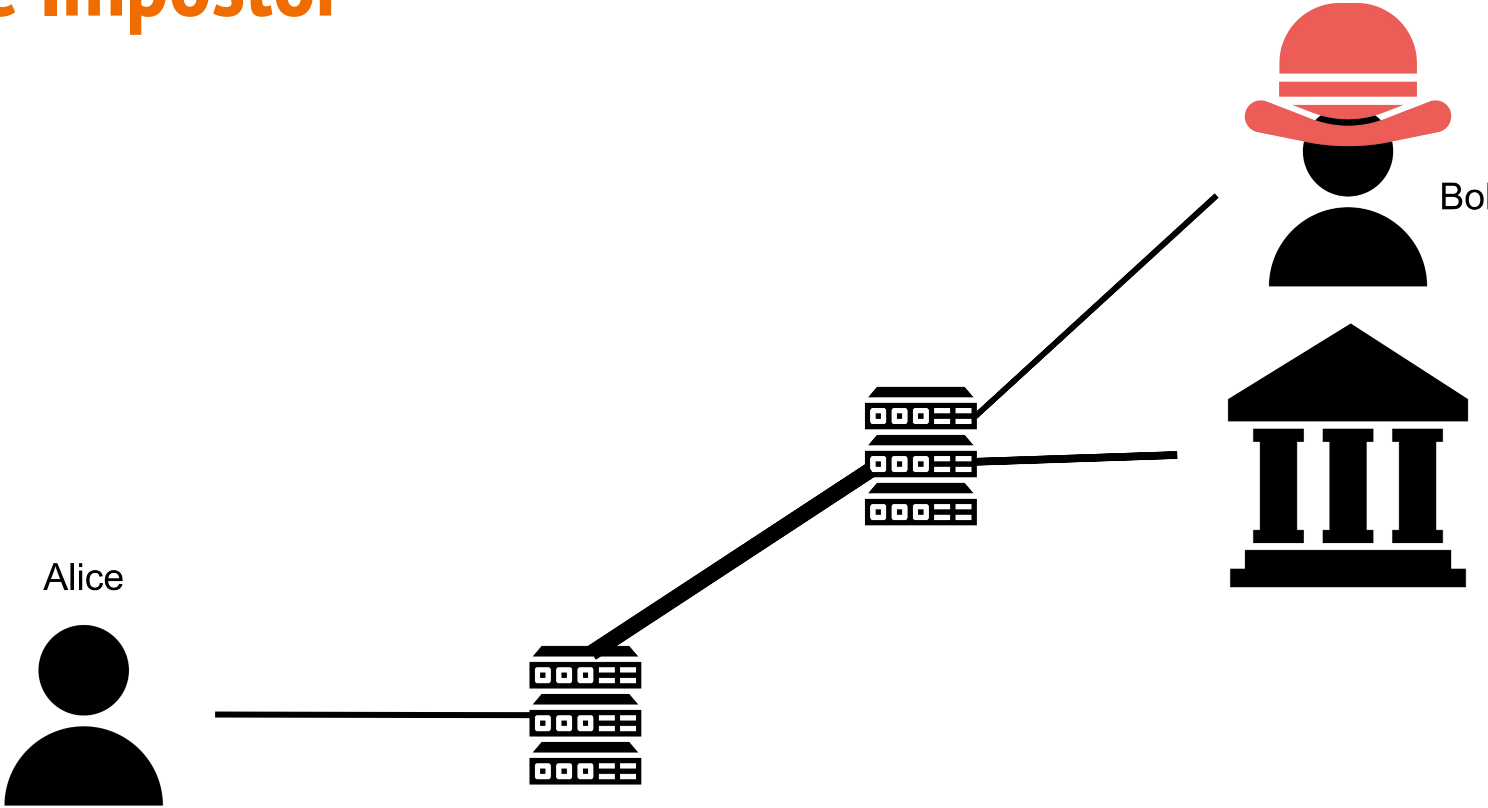
The Impostor (or Impersonator)



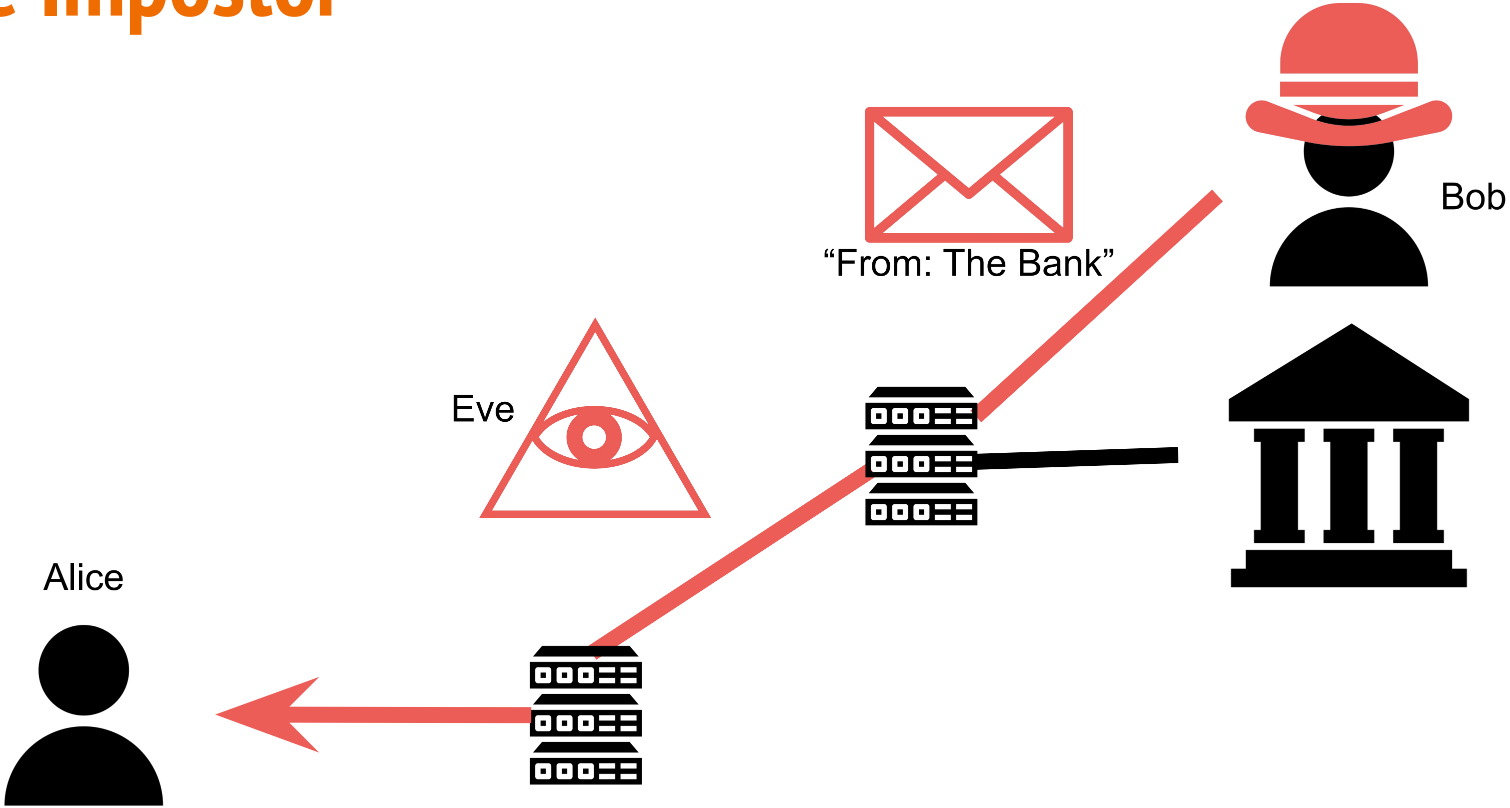
The Impostor



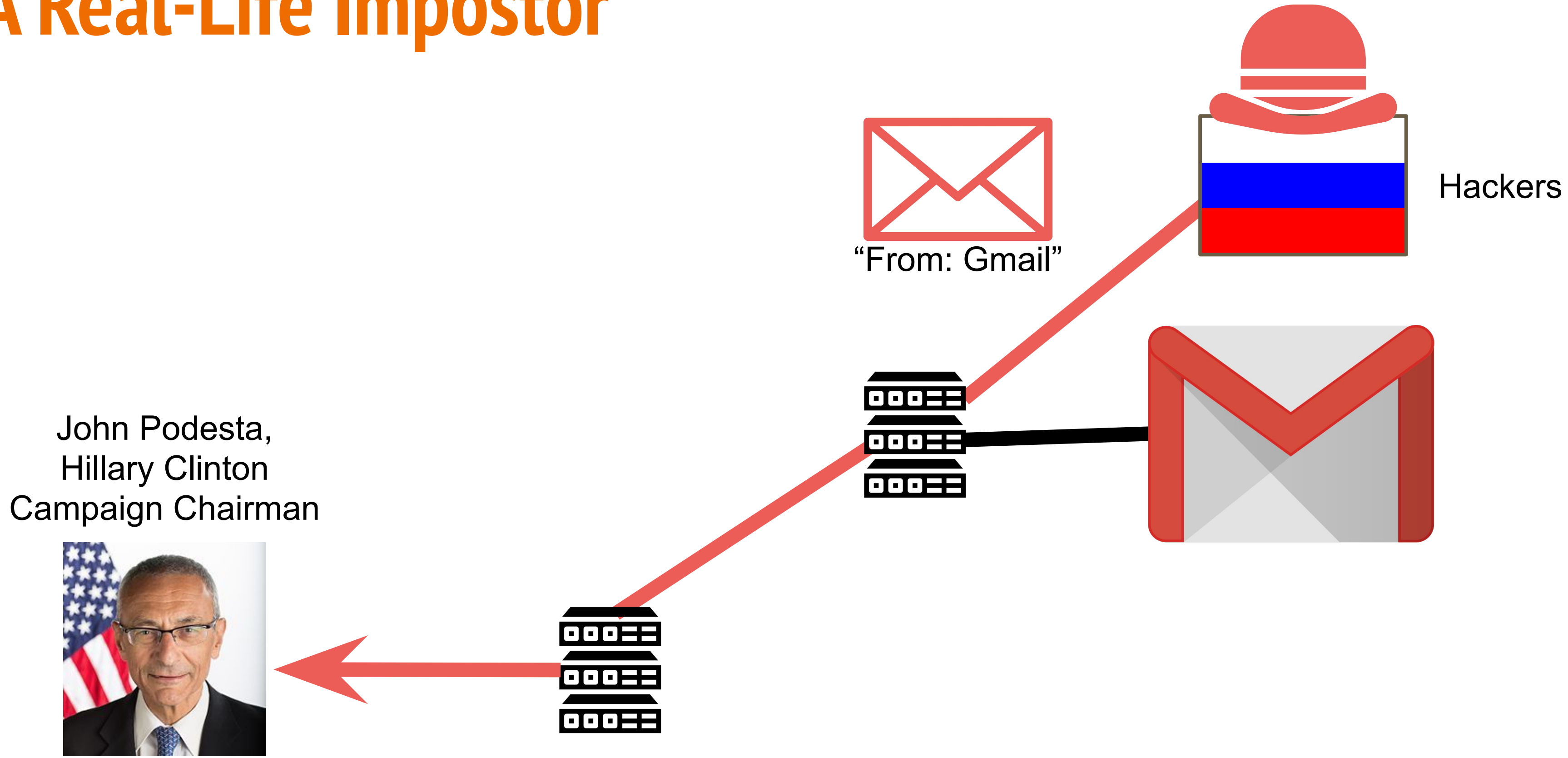
The Impostor



The Impostor



A Real-Life Impostor

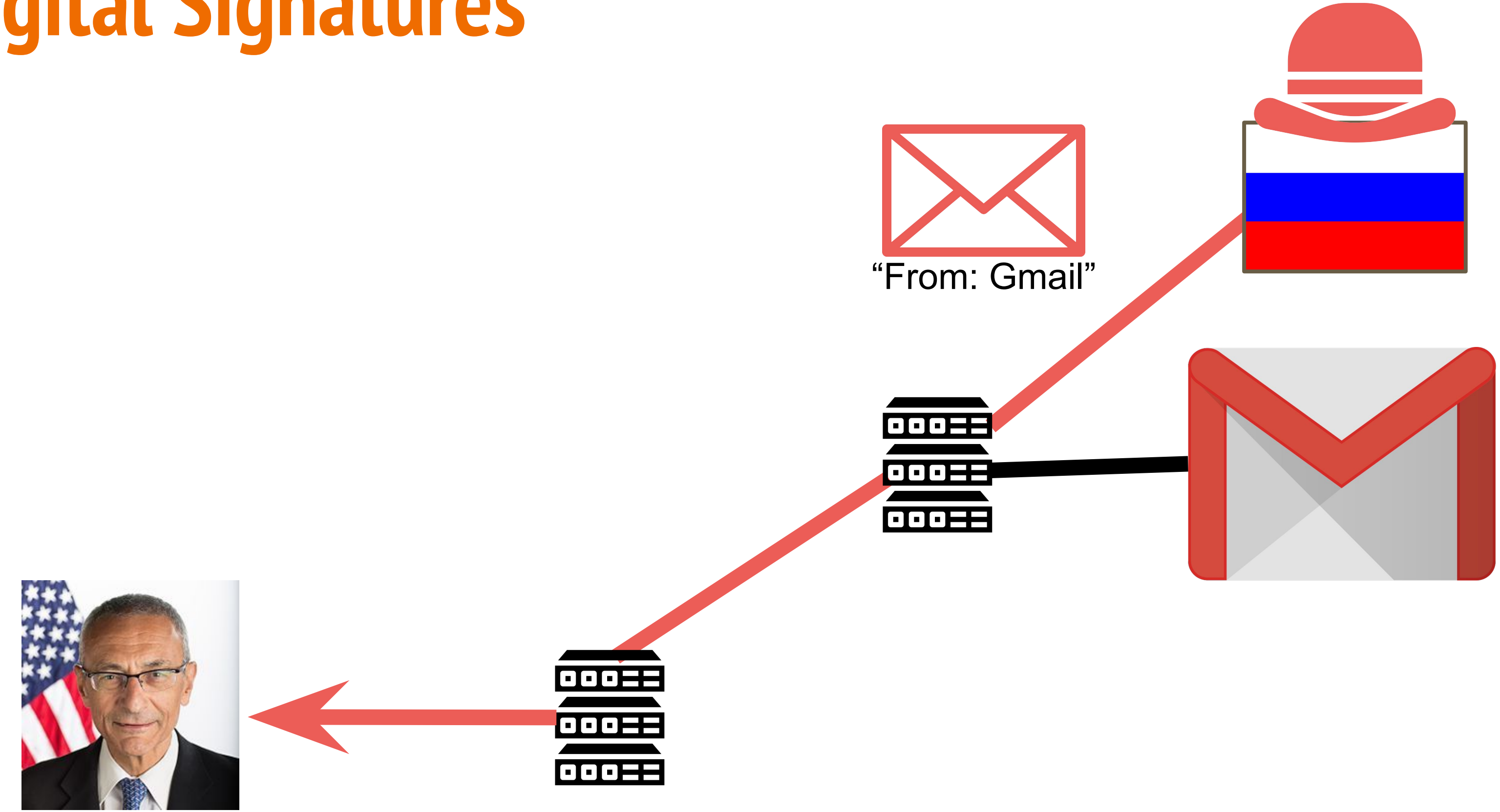


Images: Wikimedia

> Someone just used your password to try to sign in to your Google
Account
> john.podesta@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>
> CHANGE PASSWORD <https://bit.ly/1PibSU0>
>
> Best,
> The Gmail Team
> You received this mandatory email service announcement to update you
about
> important changes to your Google product or account.
>

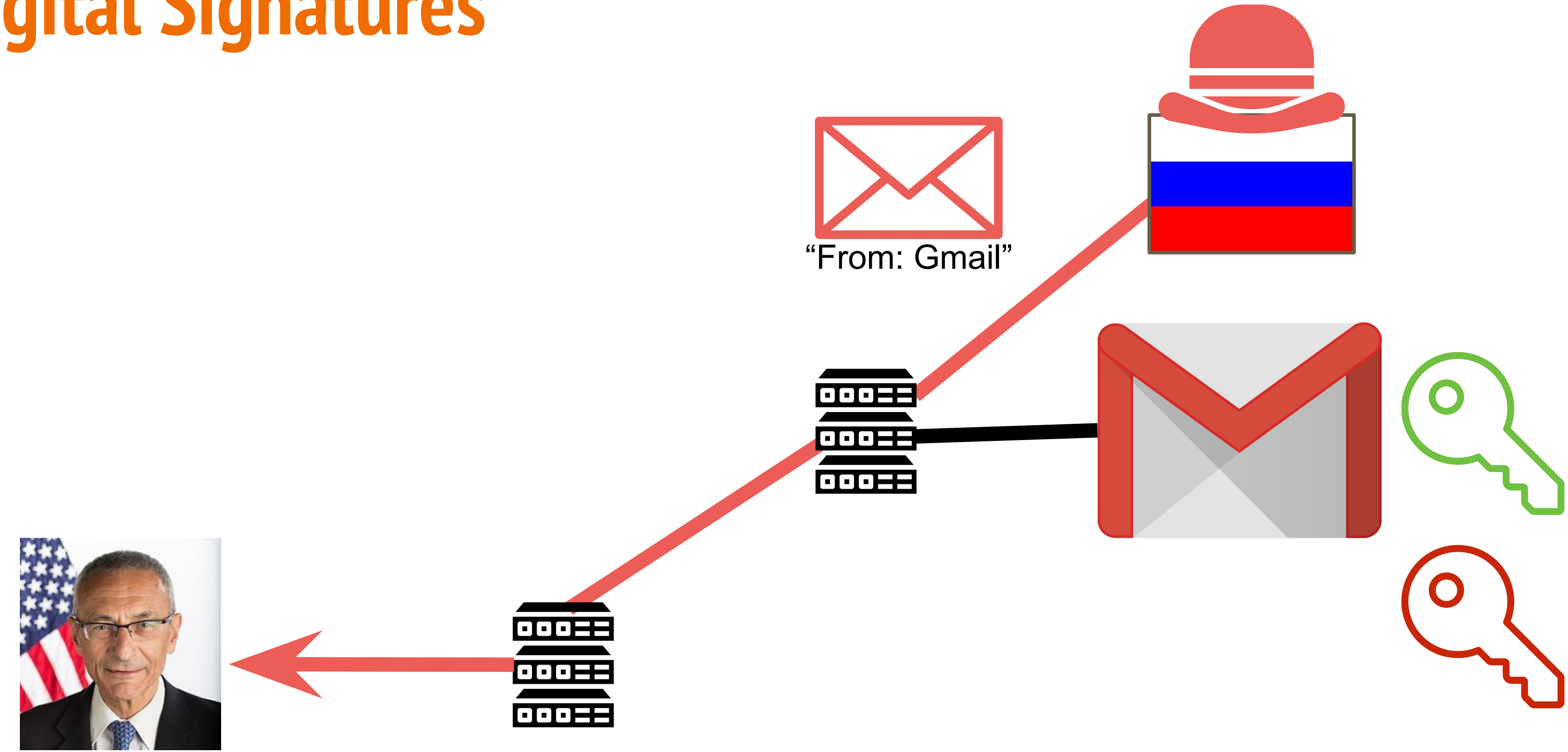
> Someone just used your password to try to sign in to your Google
Account
> john.podesta@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>
> CHANGE PASSWORD <https://bit.ly/1PibSU0>
>
> Best,
> The Gmail Team
> You received this mandatory email service announcement to update you
about
> important changes to your Google product or account.
>

Digital Signatures



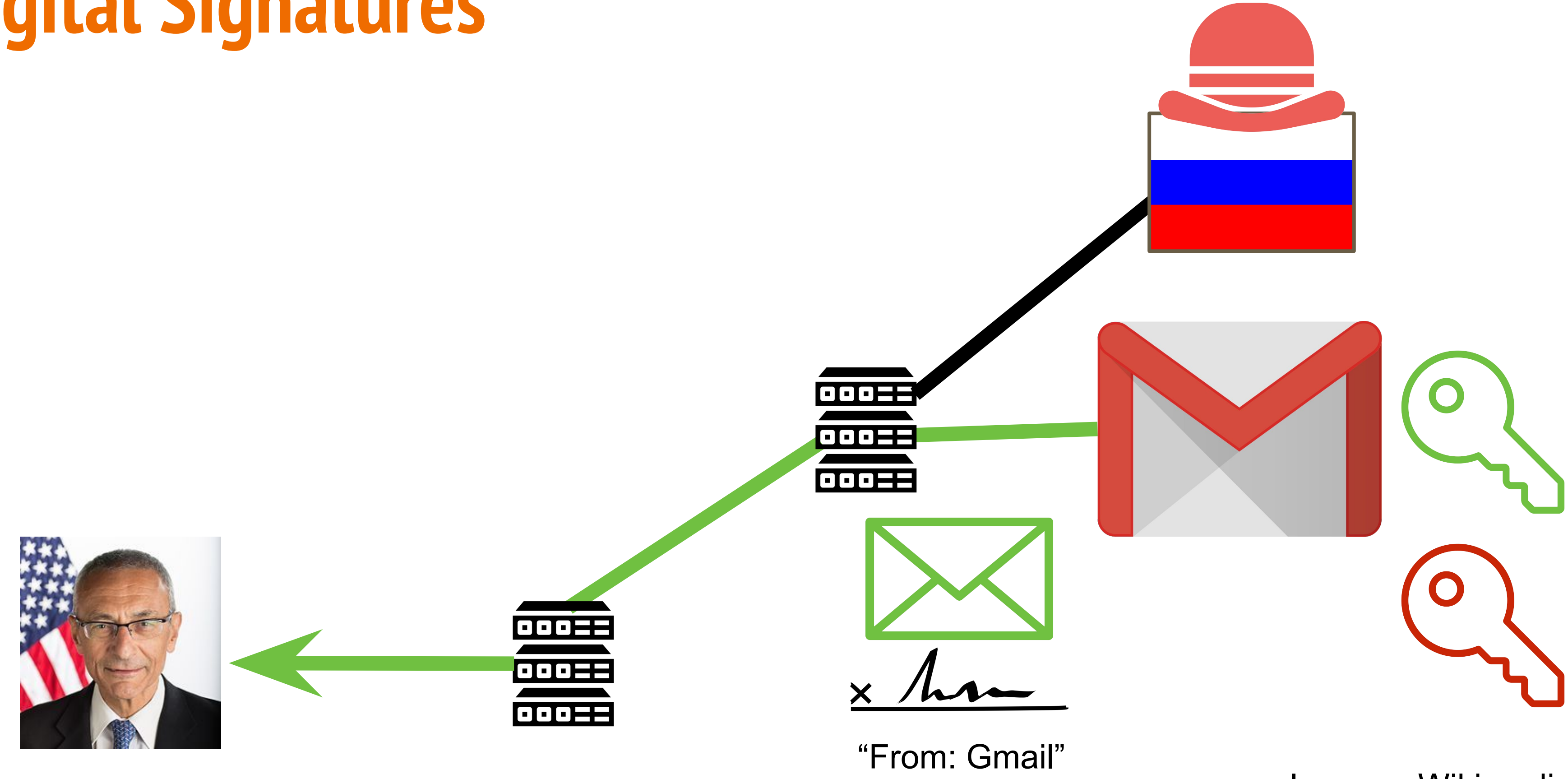
Images: Wikimedia

Digital Signatures



Images: Wikimedia

Digital Signatures



Images: Wikimedia

Digital Signatures

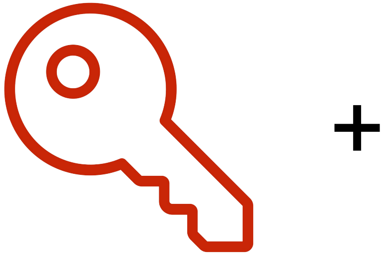
Sending Side

Digital Signatures

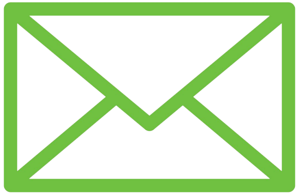
Private Key

Sending Side

Digital Signatures



Private Key



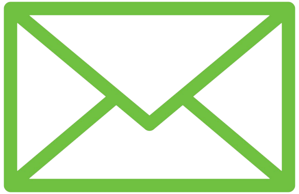
Message

Sending Side

Digital Signatures

Private Key

+

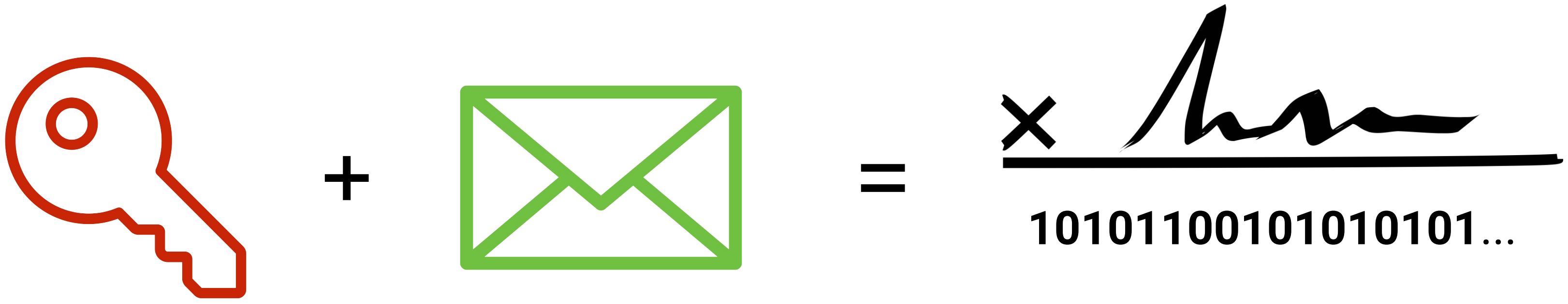


Message

=

Sending Side

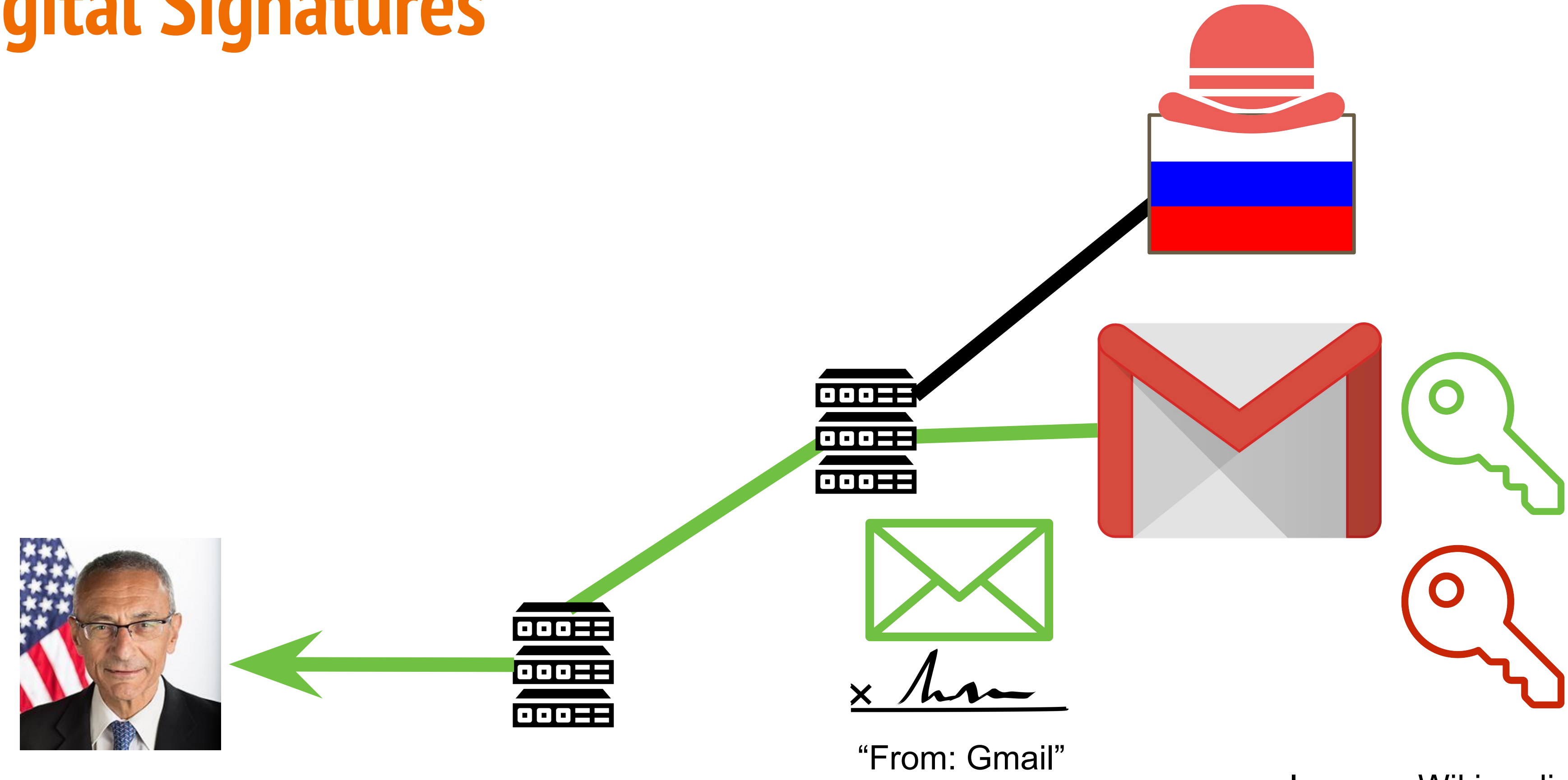
Digital Signatures



Private Key | Message | Unique Digital Signature

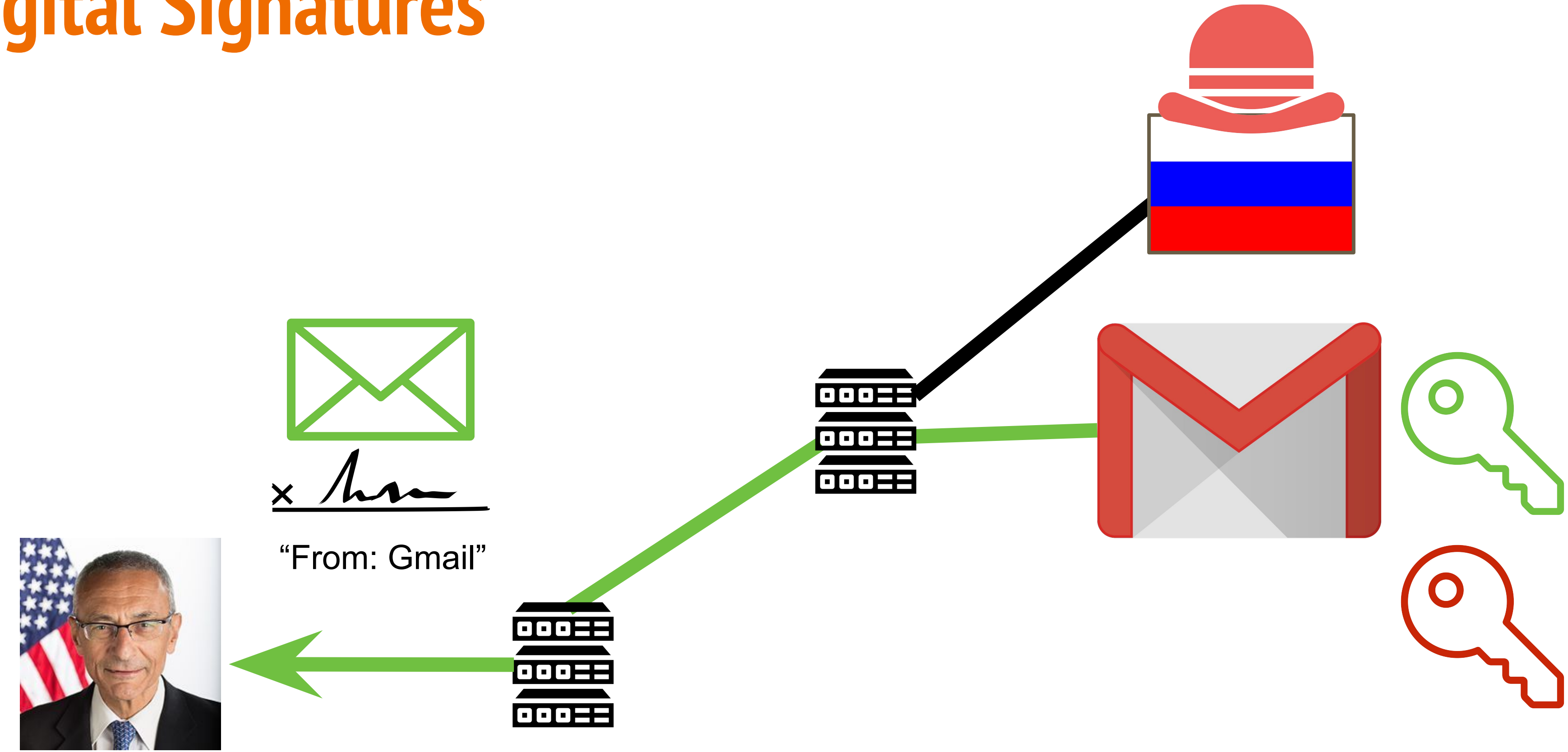
Sending Side

Digital Signatures



Images: Wikimedia

Digital Signatures



Images: Wikimedia

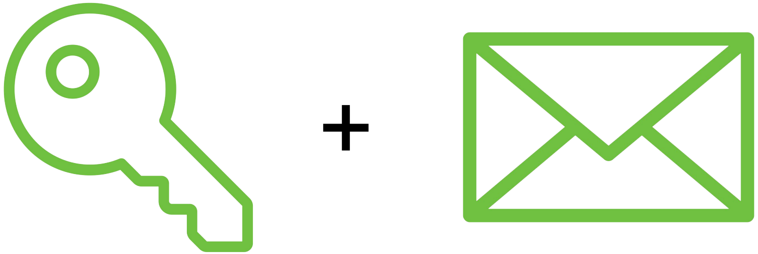
Receiving Side

Digital Signatures

Public Key
of Sender

Receiving Side

Digital Signatures

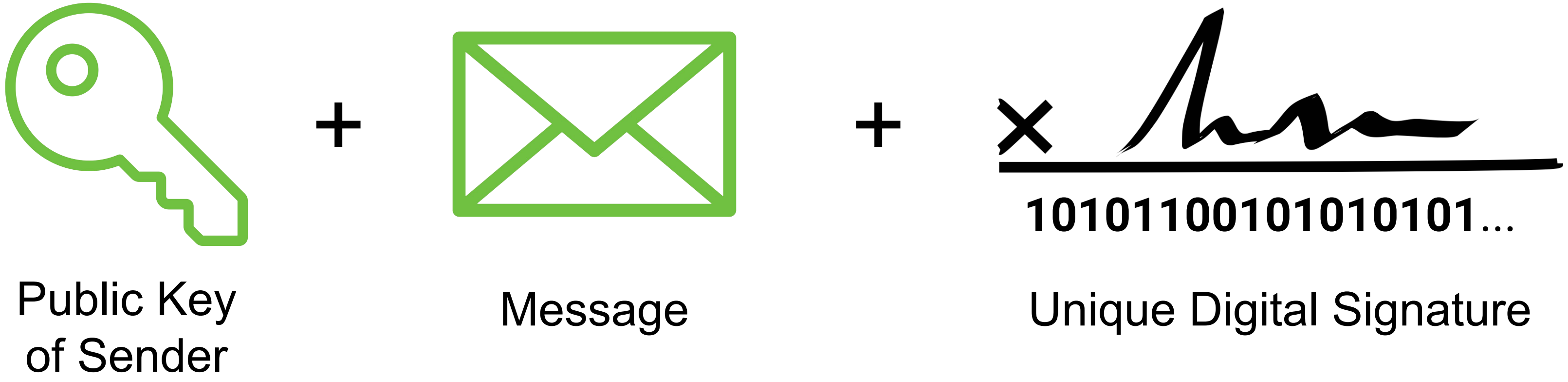


Public Key
of Sender

Message

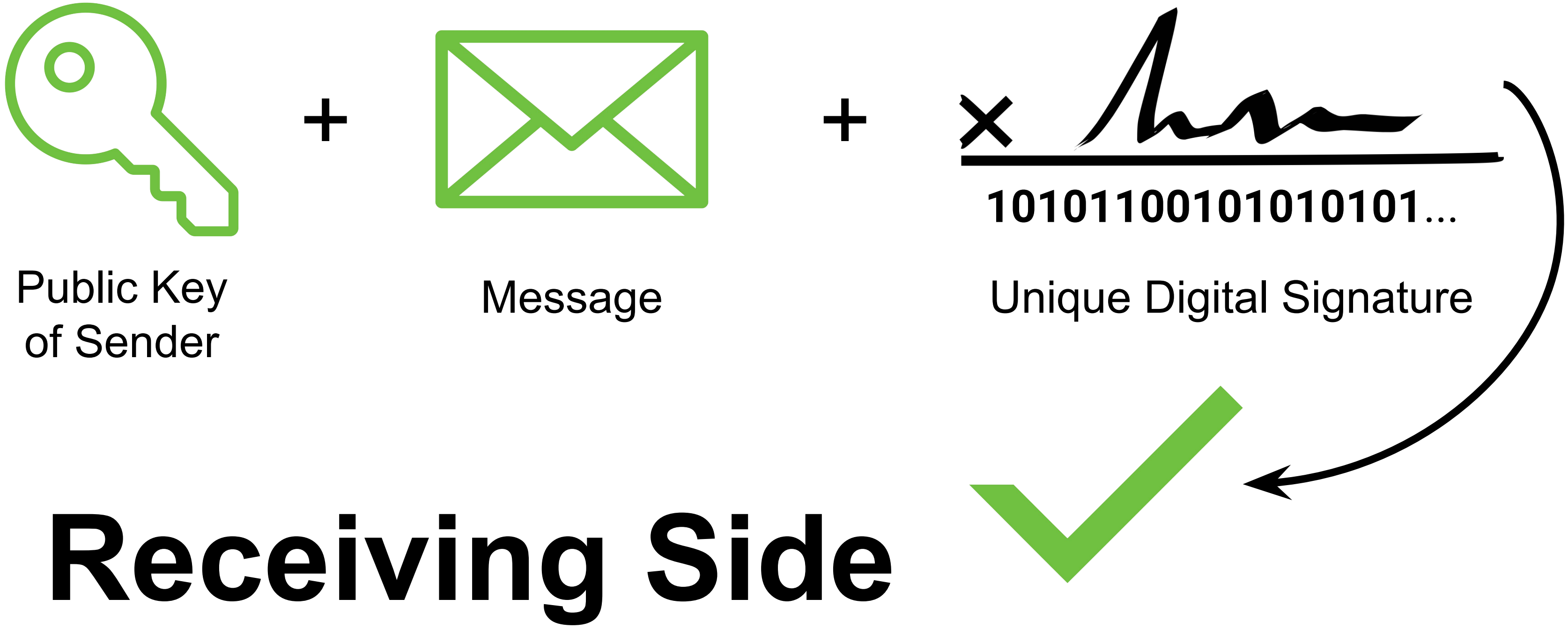
Receiving Side

Digital Signatures

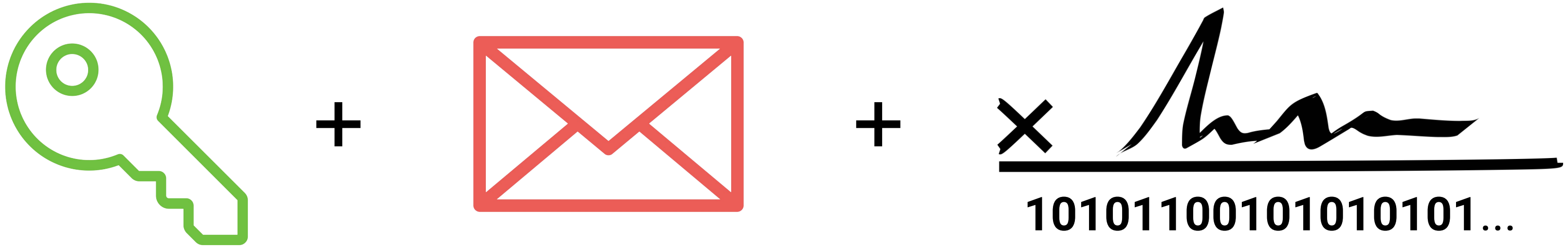


Receiving Side

Digital Signatures



Digital Signatures



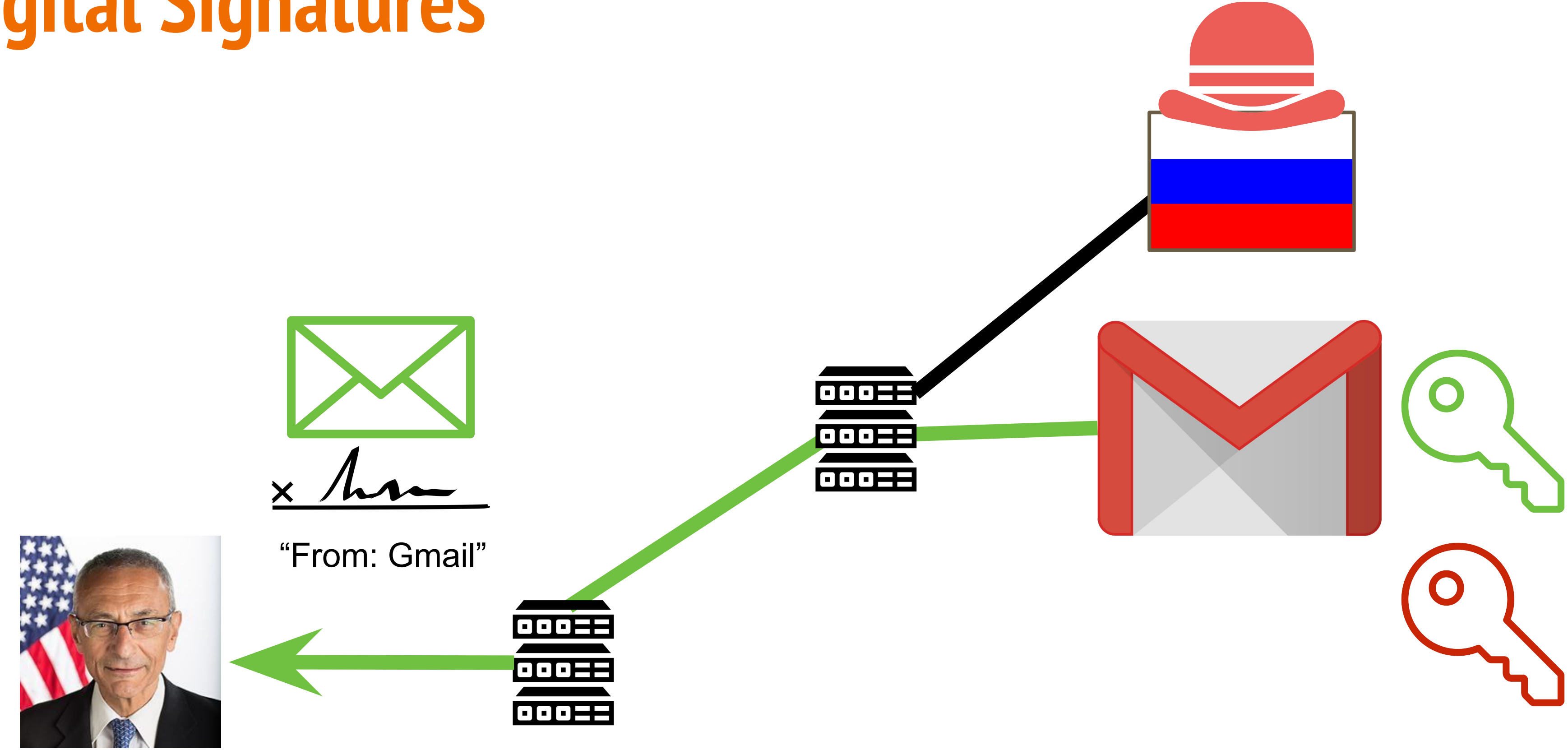
Public Key of Sender + Fraudulent Message + 10101100101010101...

Unique Digital Signature

Receiving Side

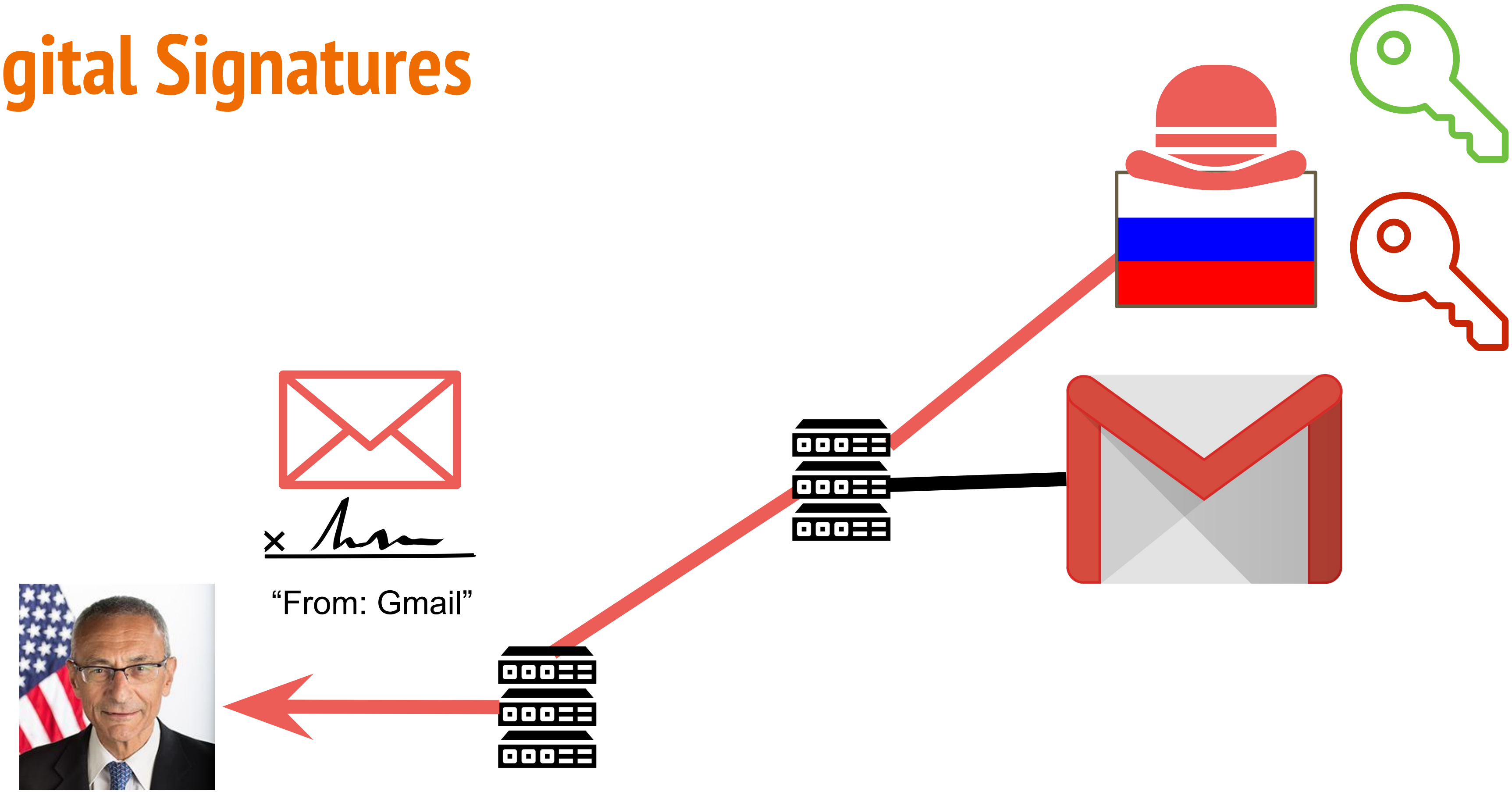
Digital Signatures

Digital Signatures

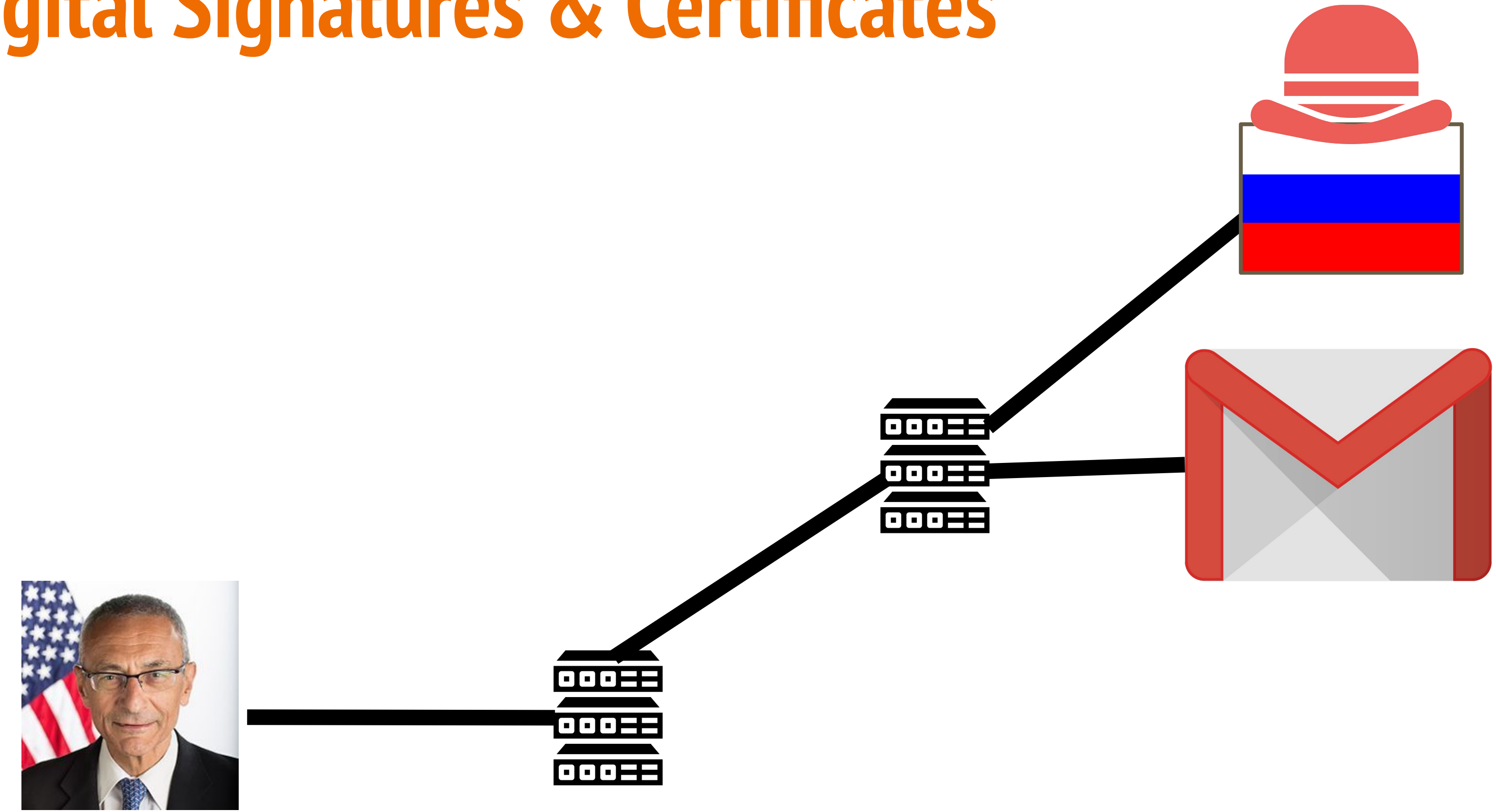


Images: Wikimedia

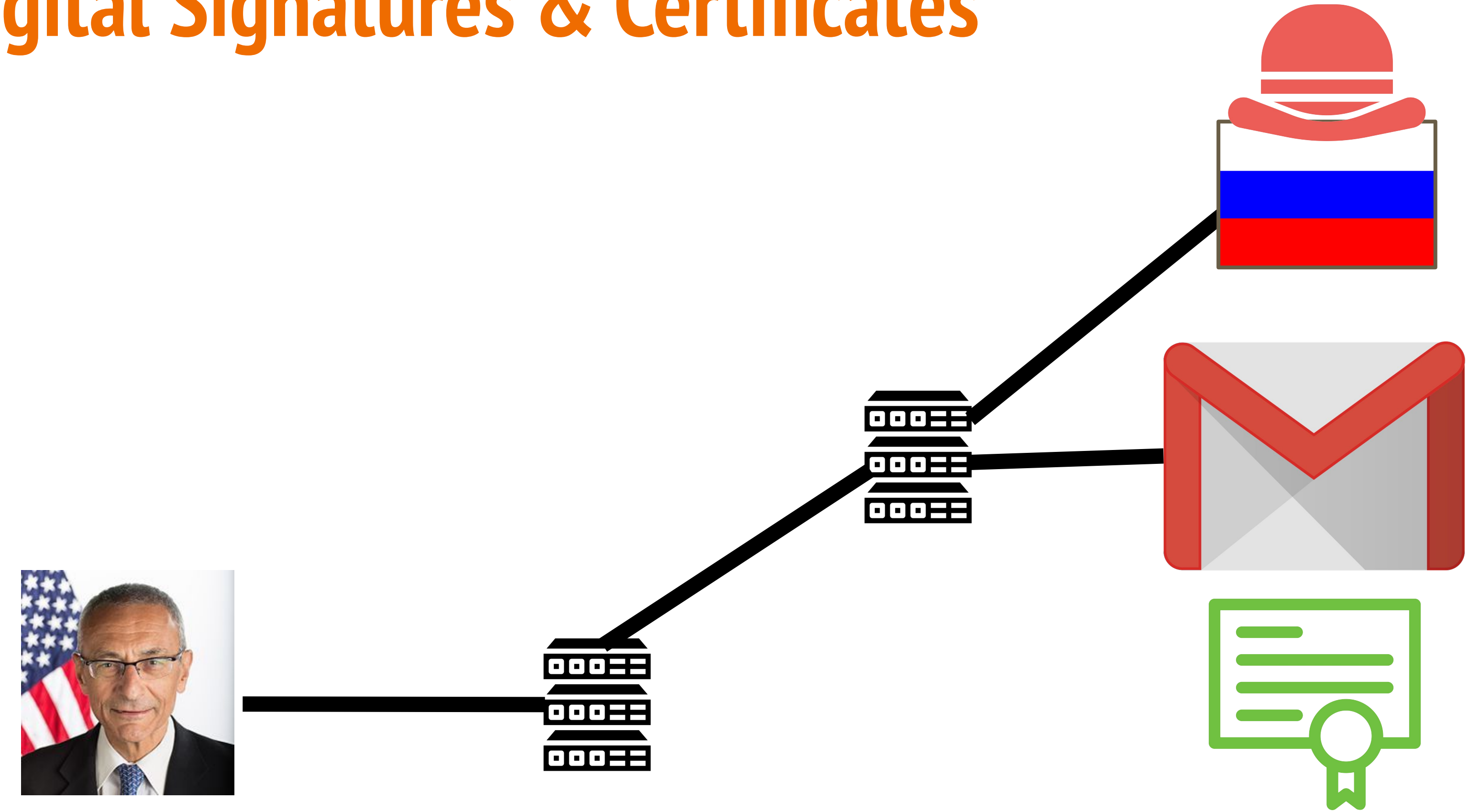
Digital Signatures



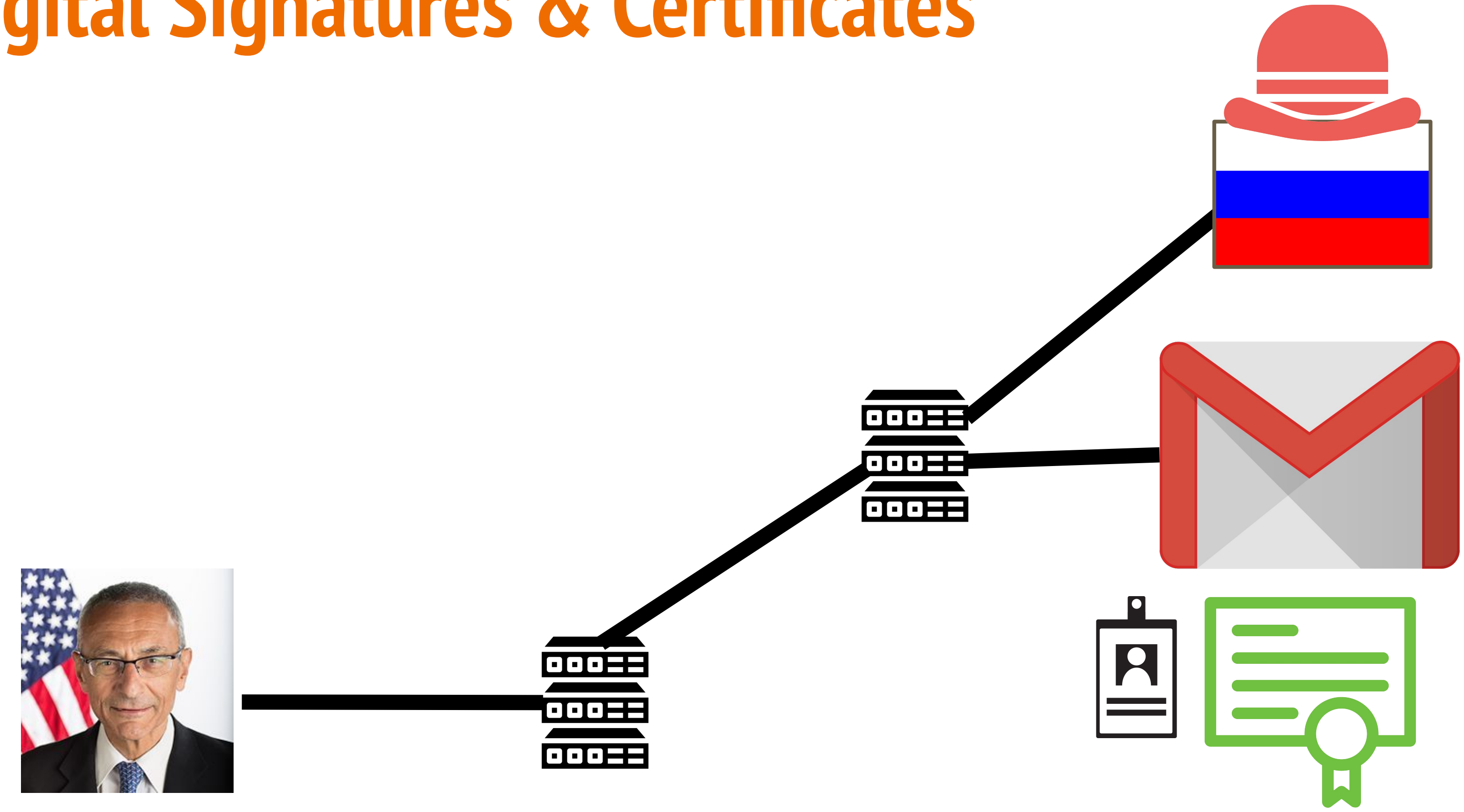
Digital Signatures & Certificates



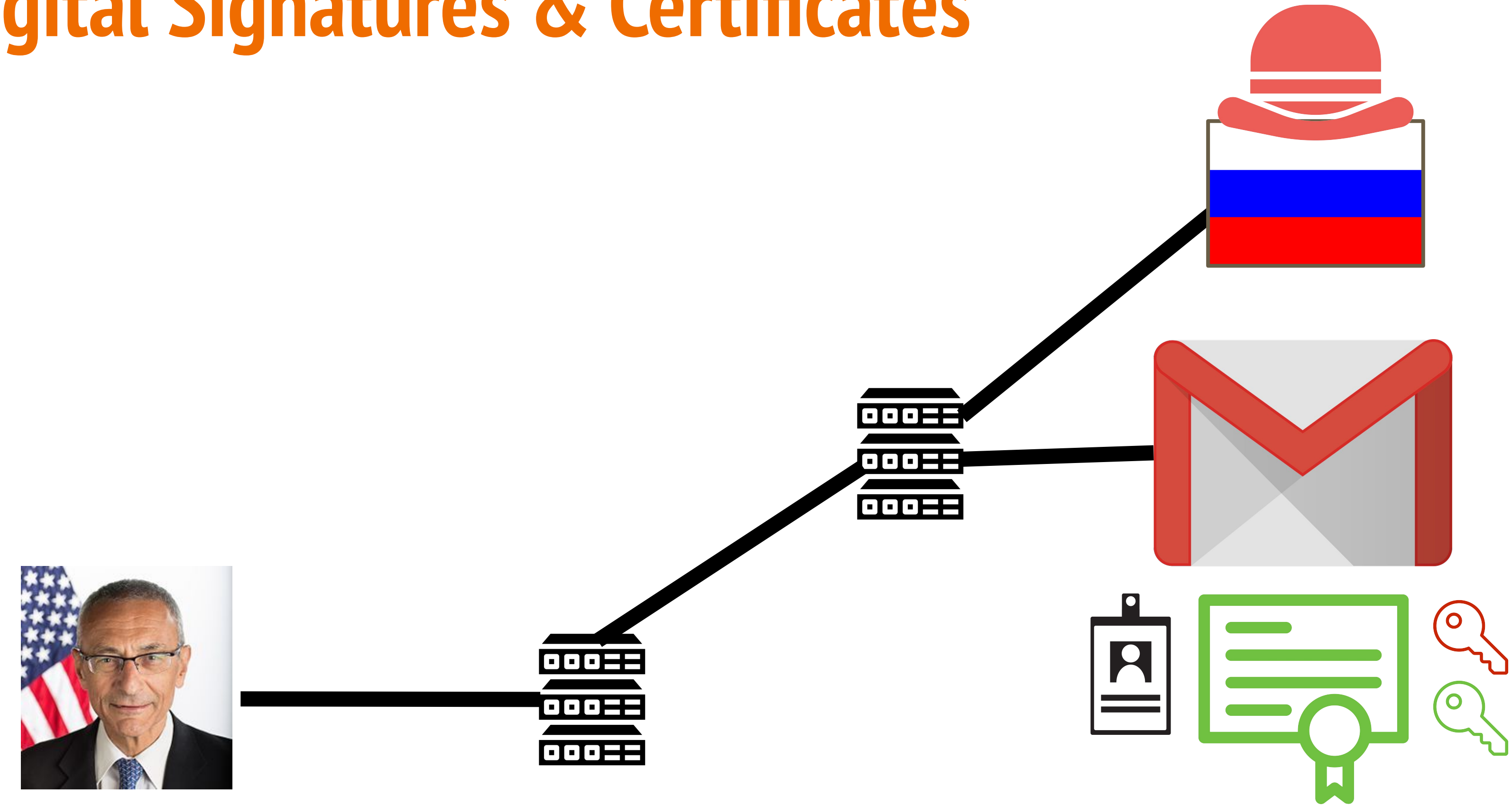
Digital Signatures & Certificates



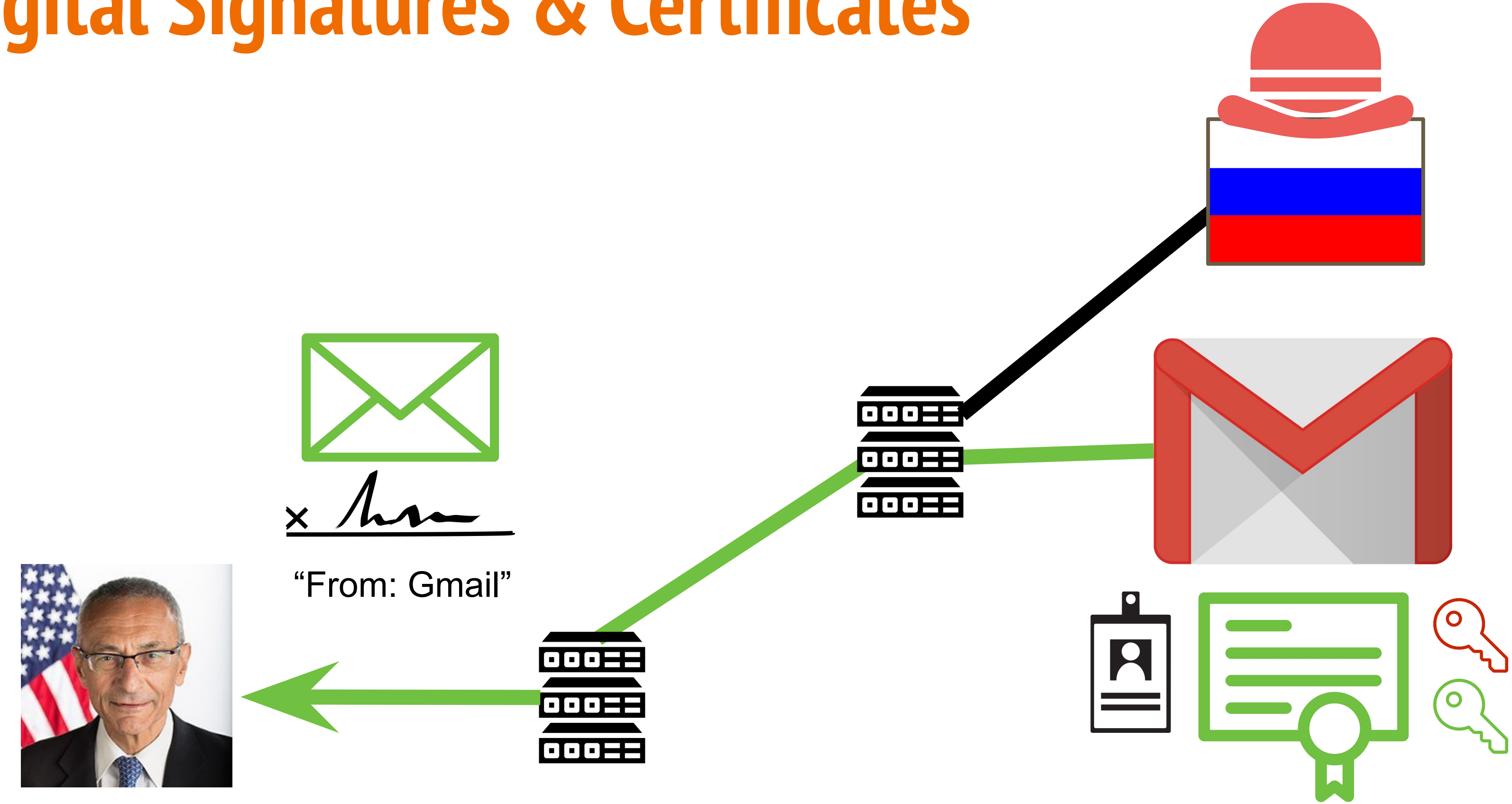
Digital Signatures & Certificates



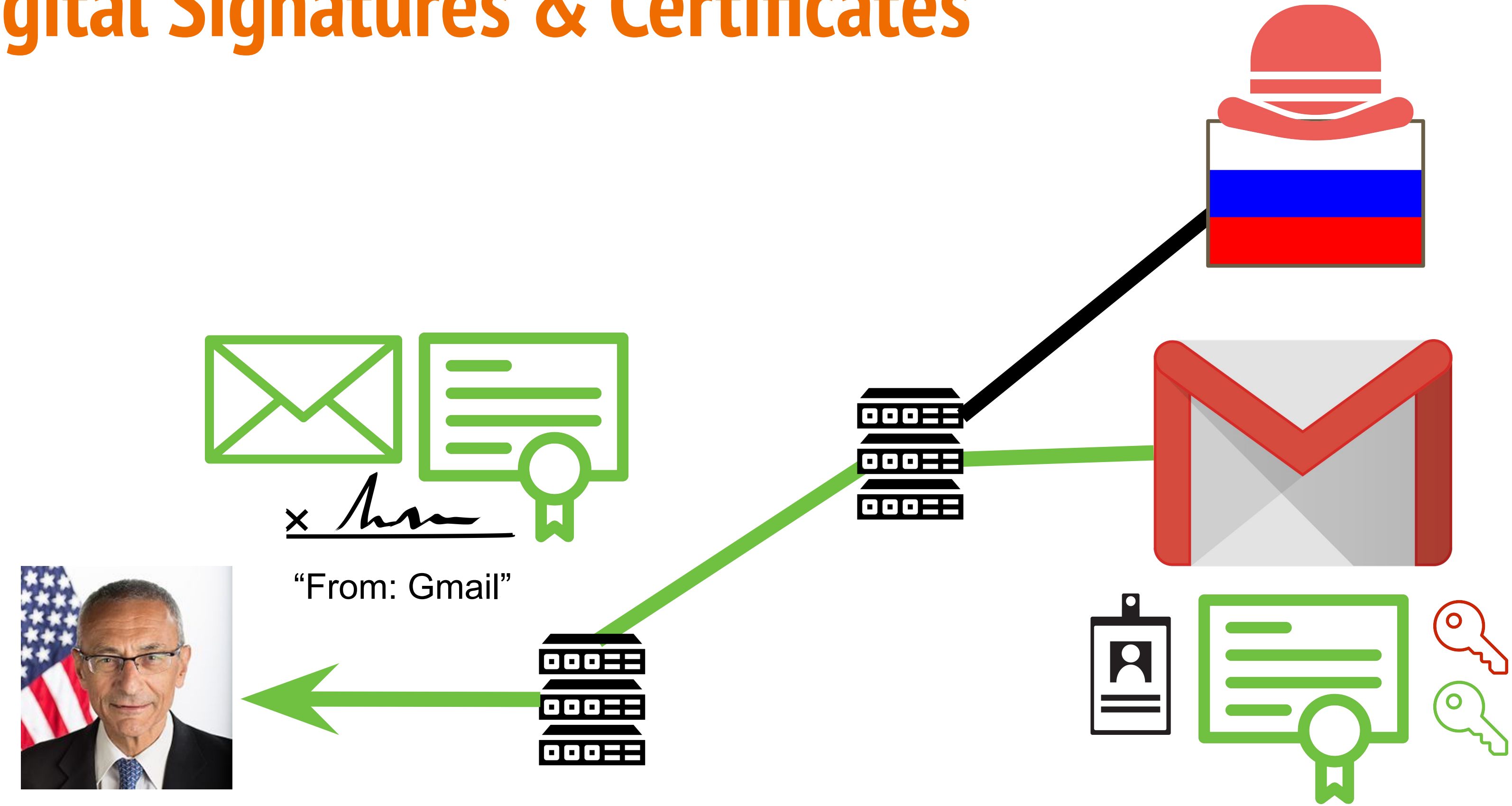
Digital Signatures & Certificates



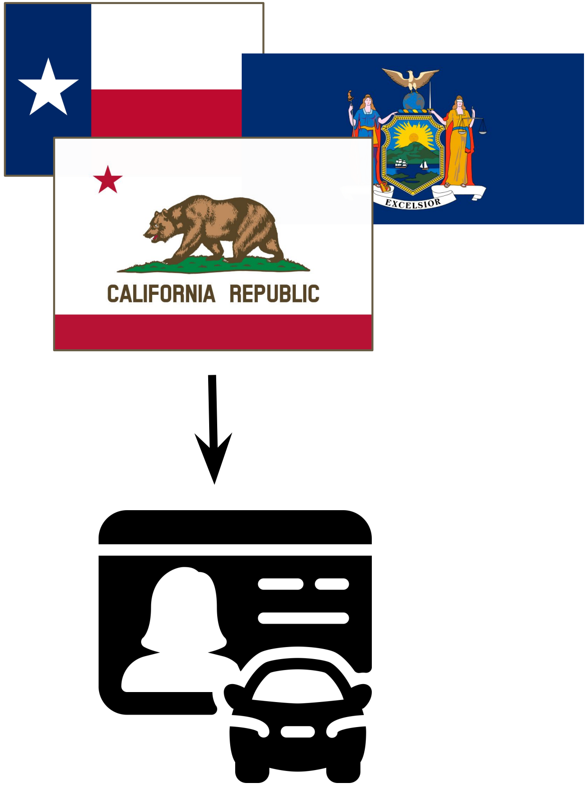
Digital Signatures & Certificates



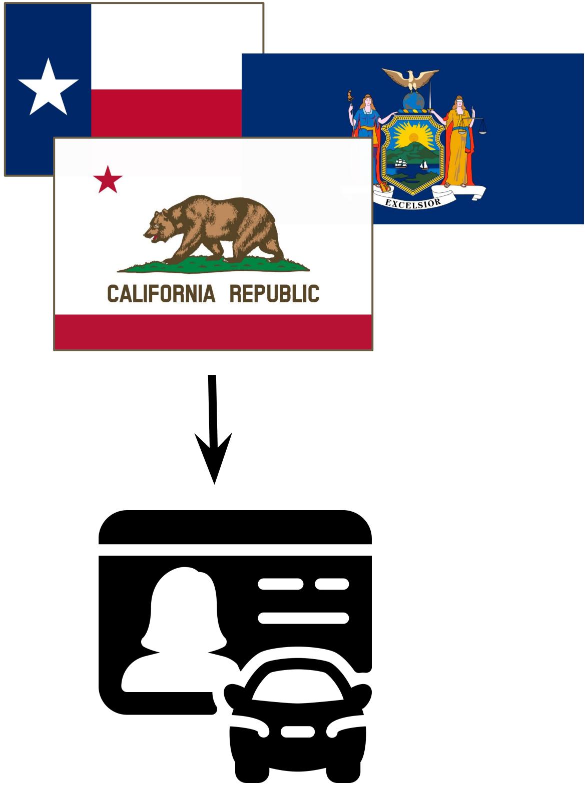
Digital Signatures & Certificates



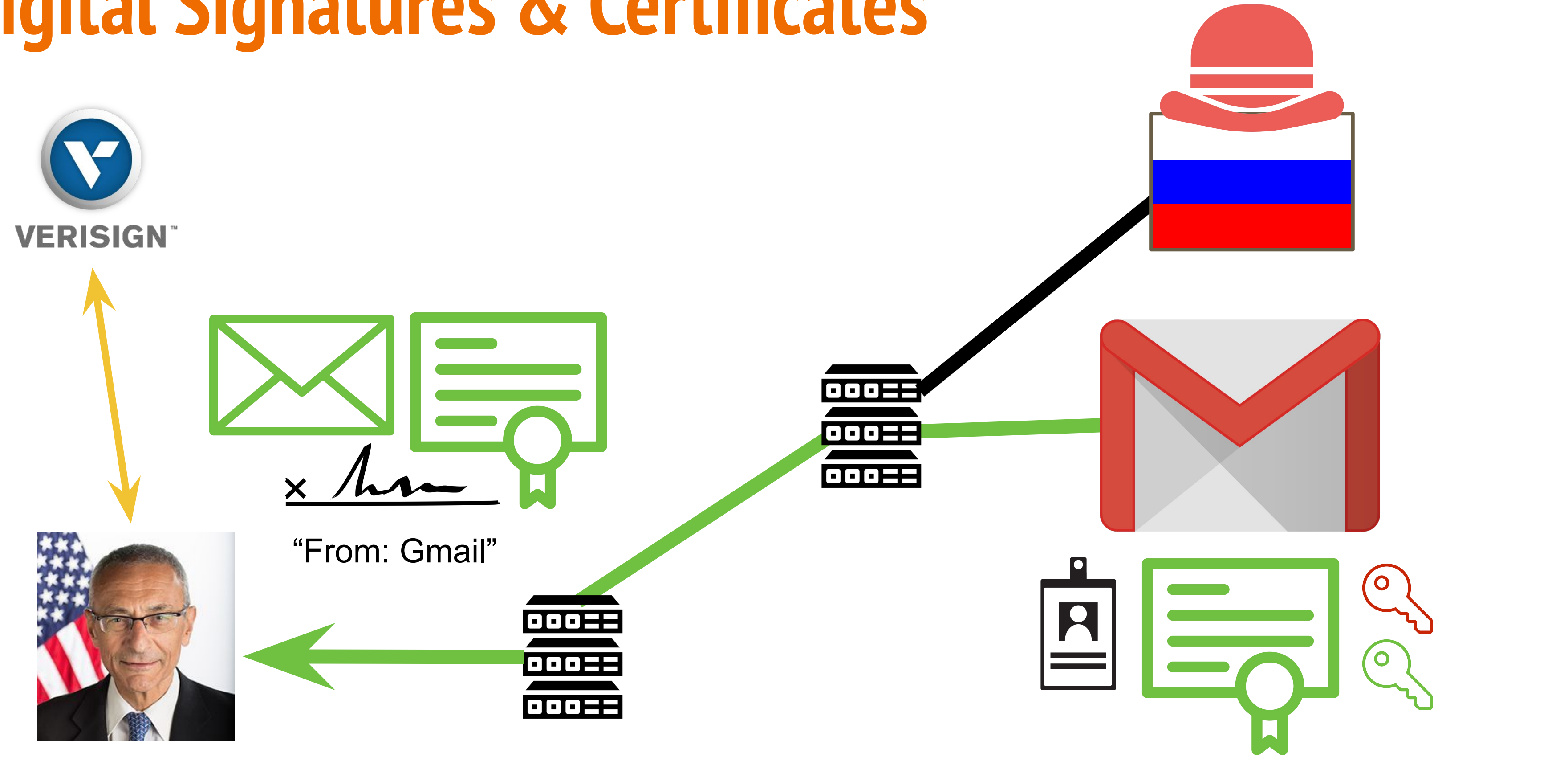
Digital Signatures & Certificates: An Analogy



Certificates Authorities

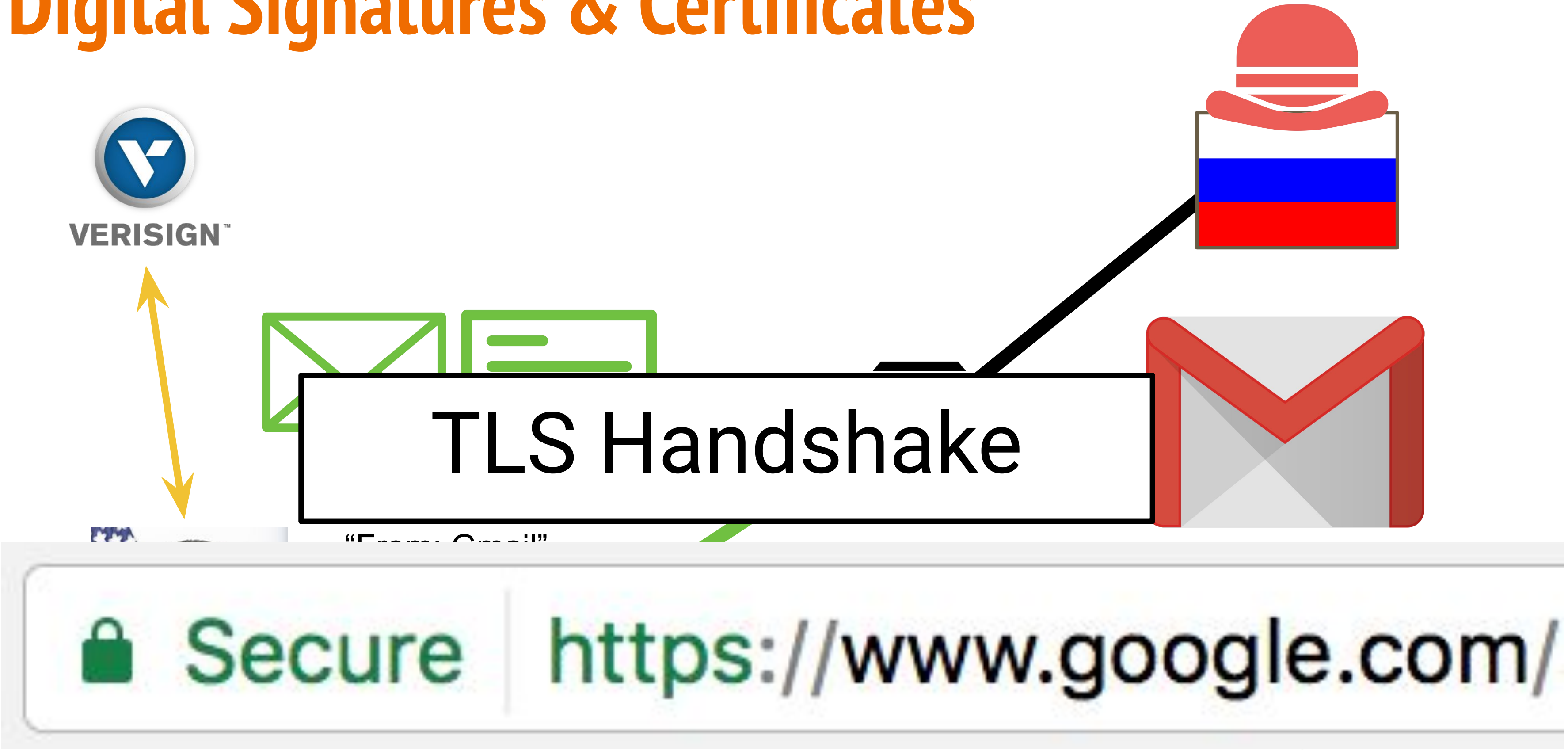


Digital Signatures & Certificates



Images: Wikimedia

Digital Signatures & Certificates



Images: Wikimedia

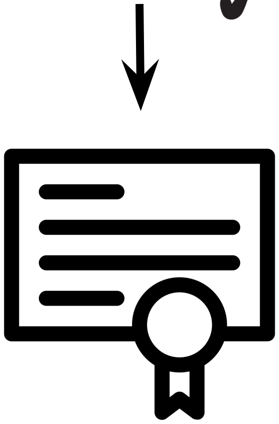
Digital Signatures & Certificates

?

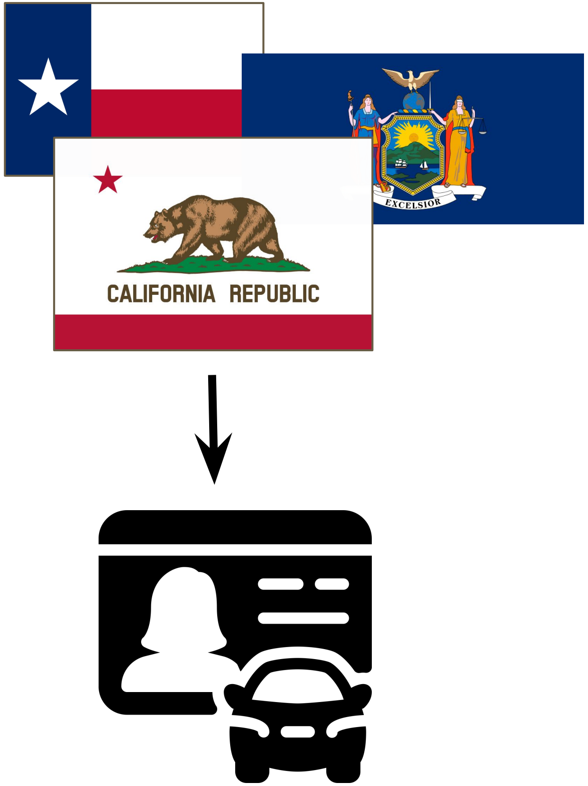
GlobalSign®
GMO INTERNET GROUP

Go Daddy®

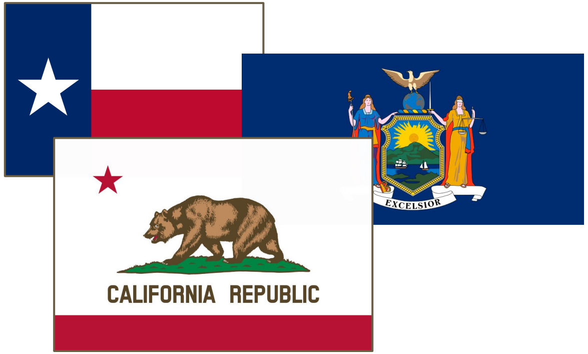
VERISIGN™

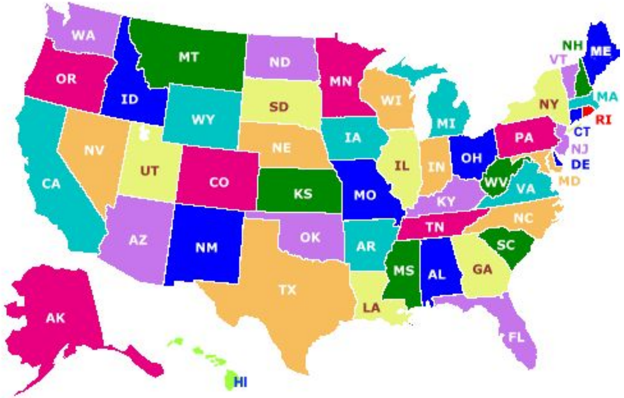


Digital Signatures & Certificates

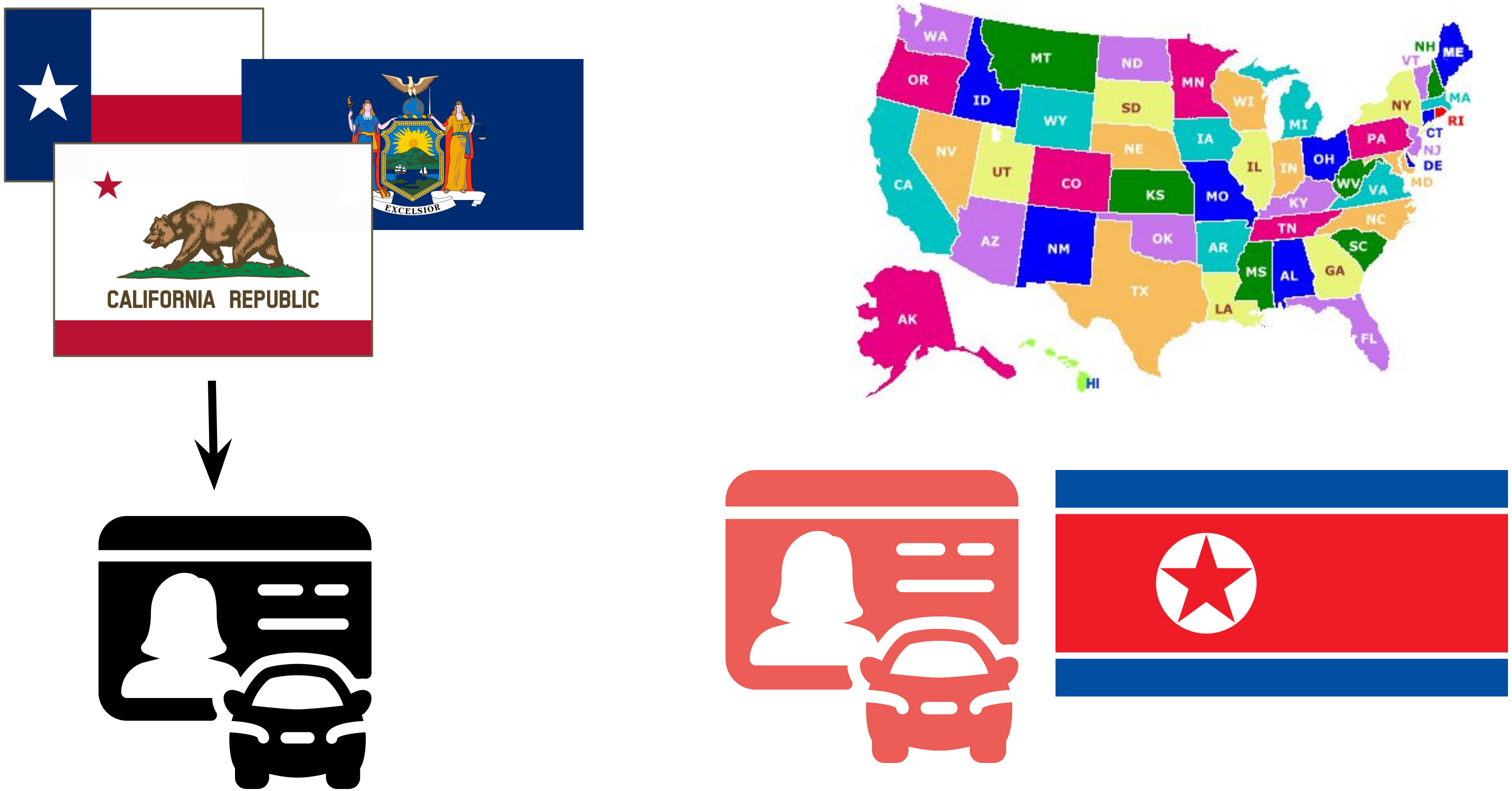


Digital Signatures & Certificates: A Bar Analogy

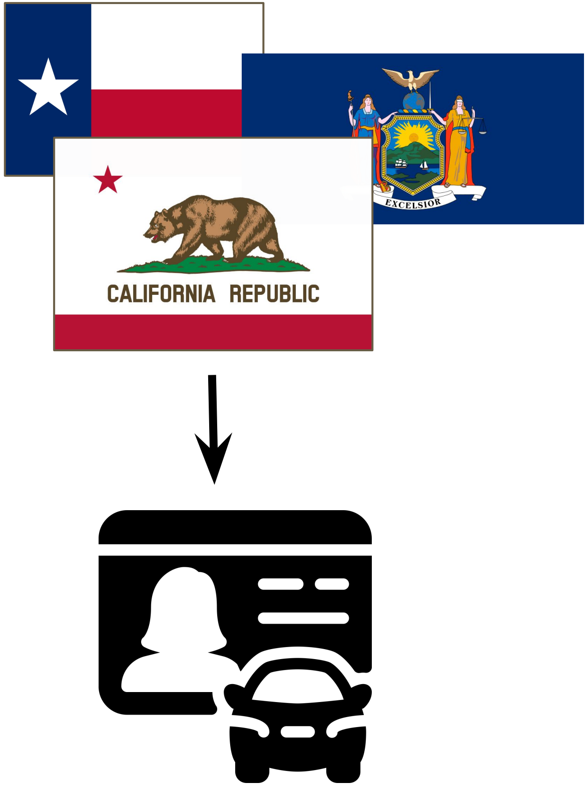


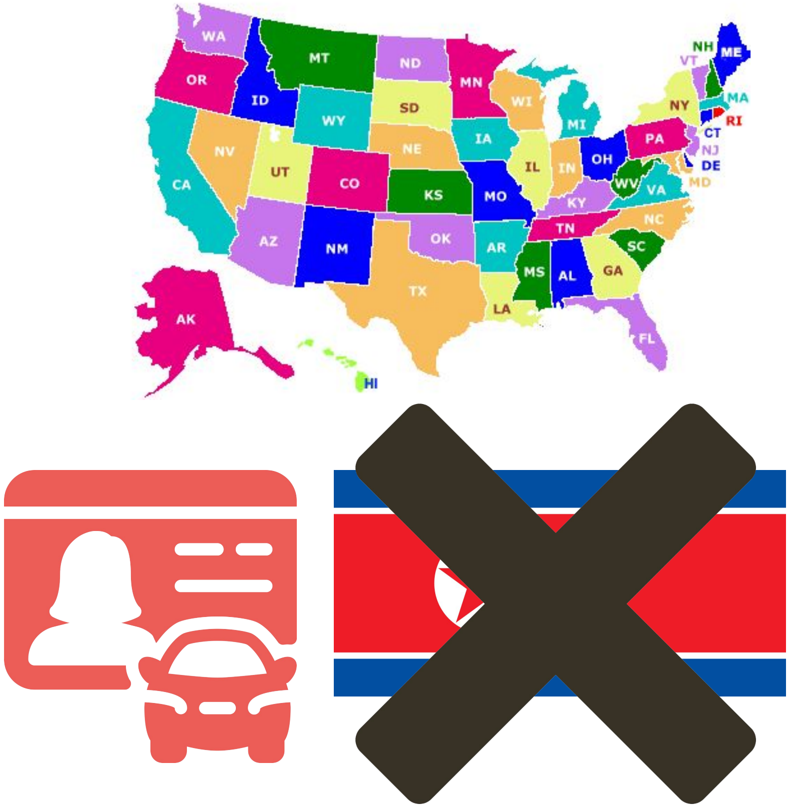


Digital Signatures & Certificates



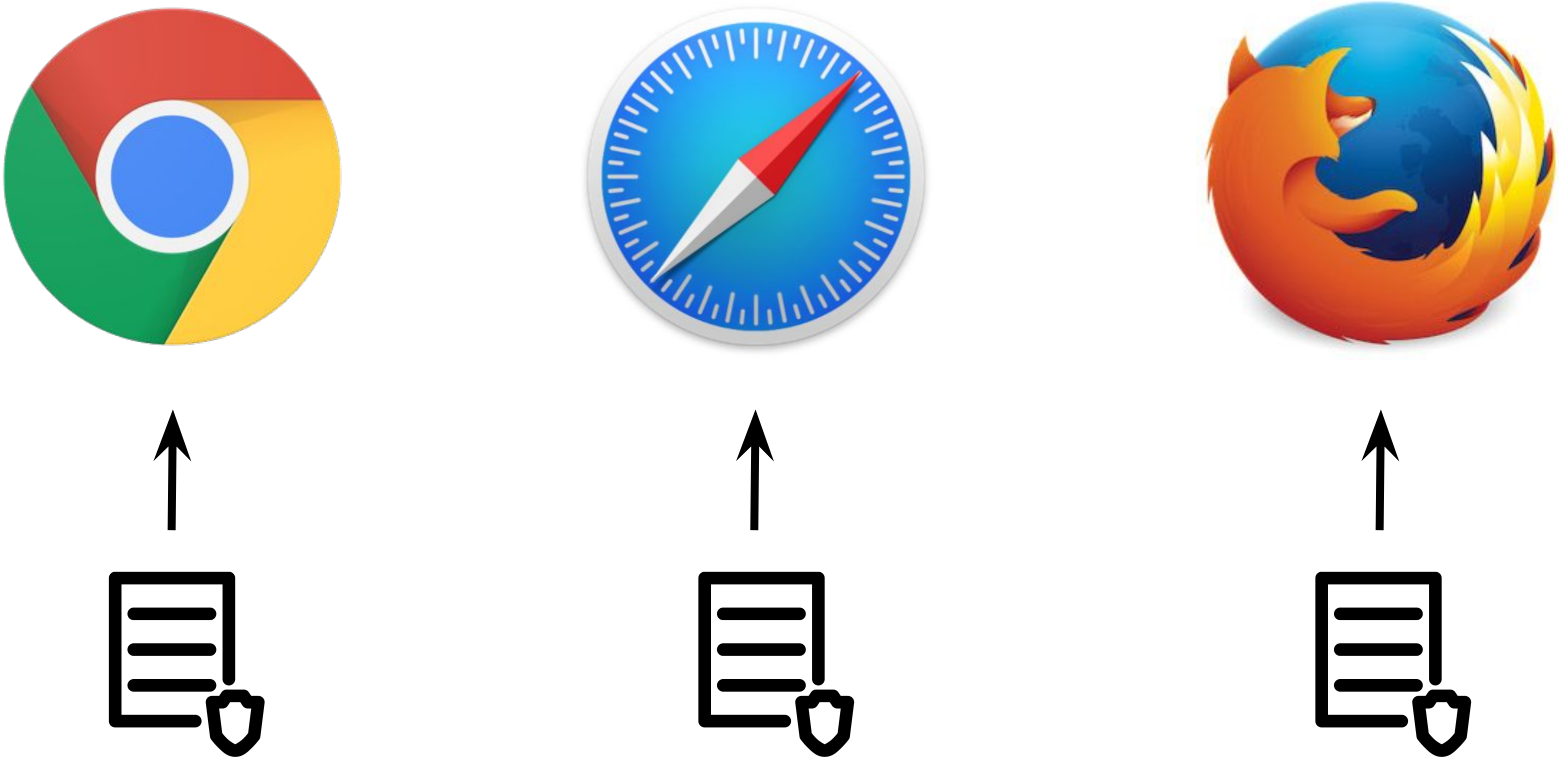
Digital Signatures & Certificates



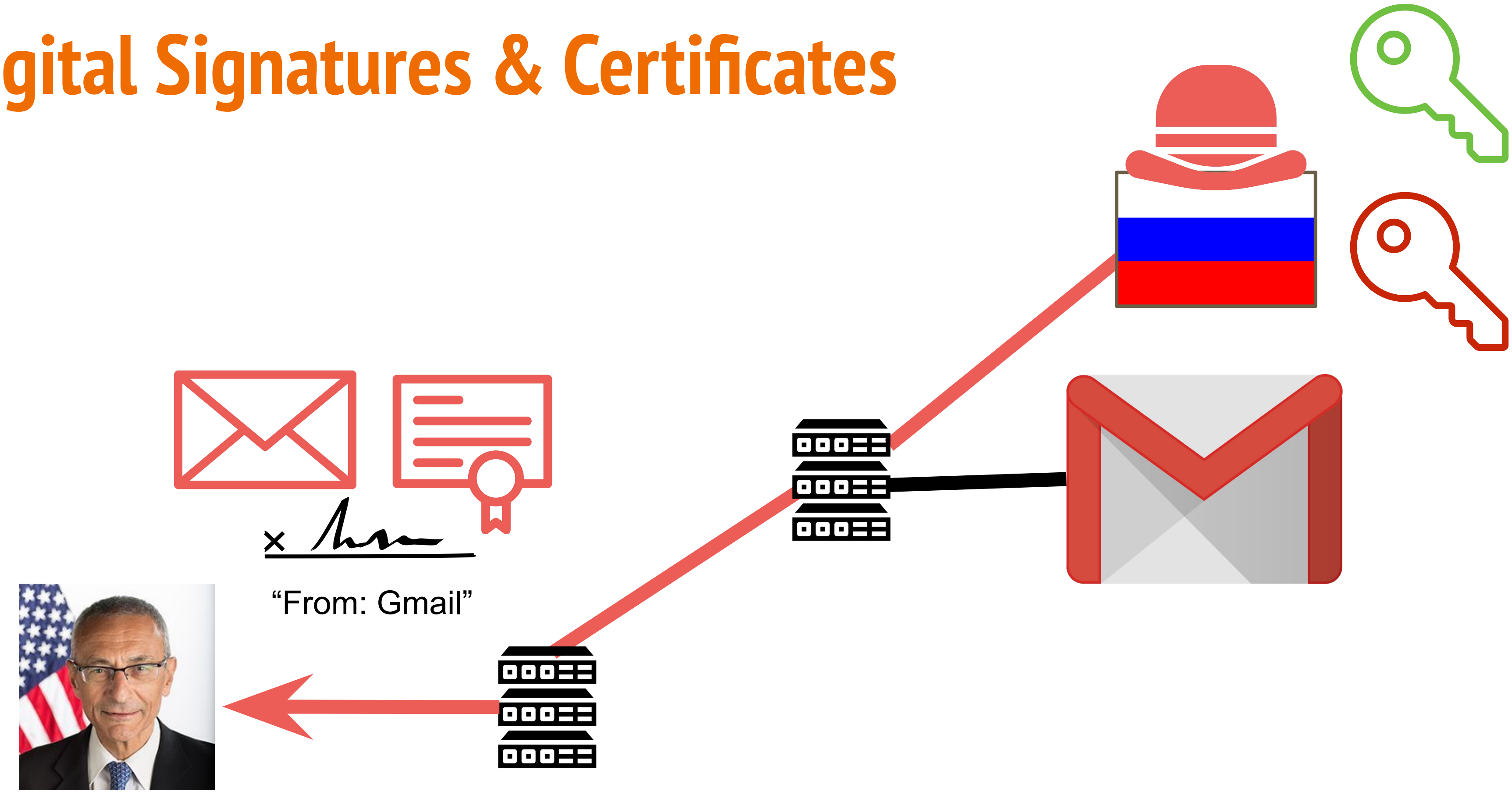


Digital Signatures & Certificates

Digital Signatures & Certificates

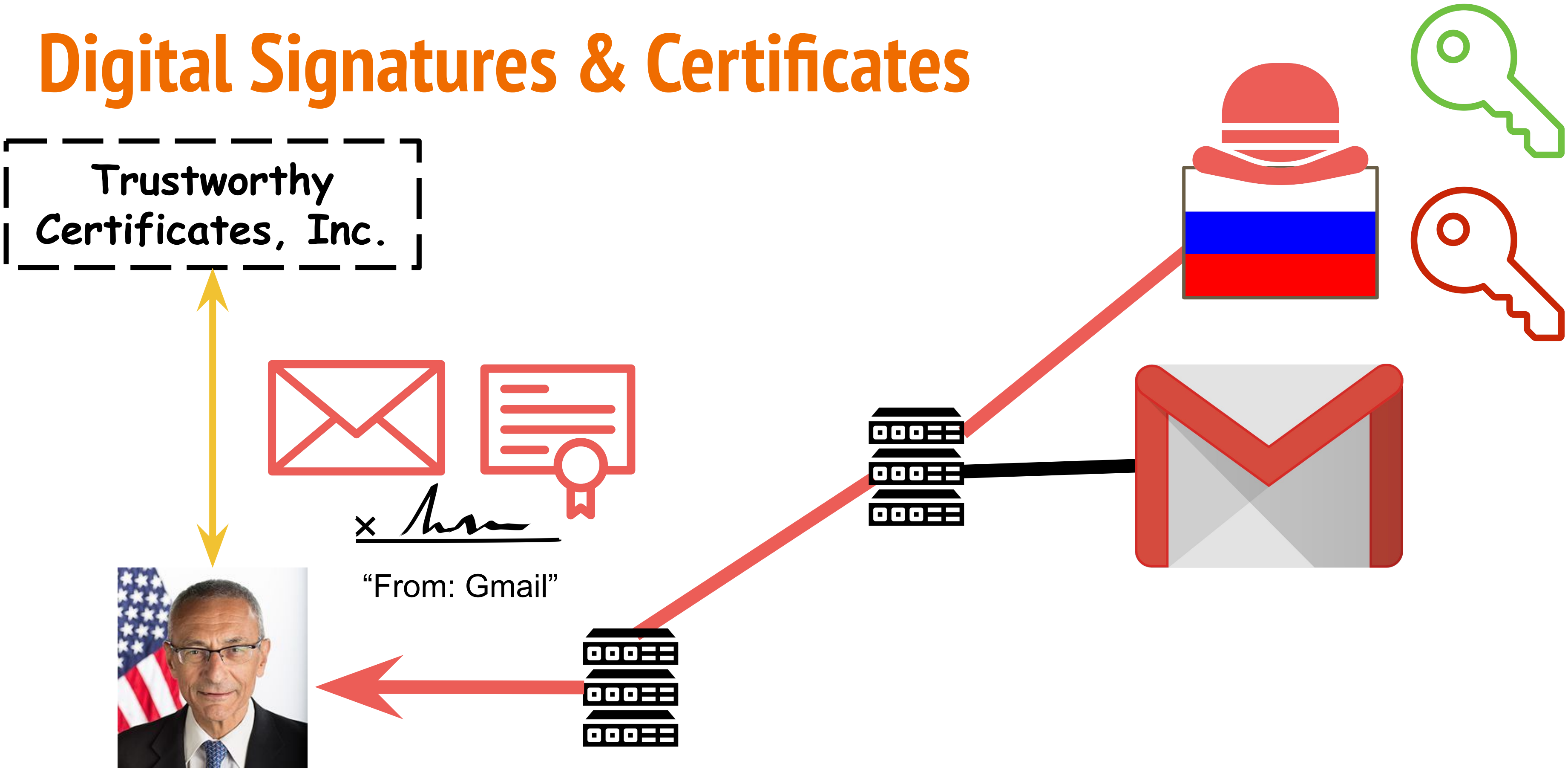


Digital Signatures & Certificates



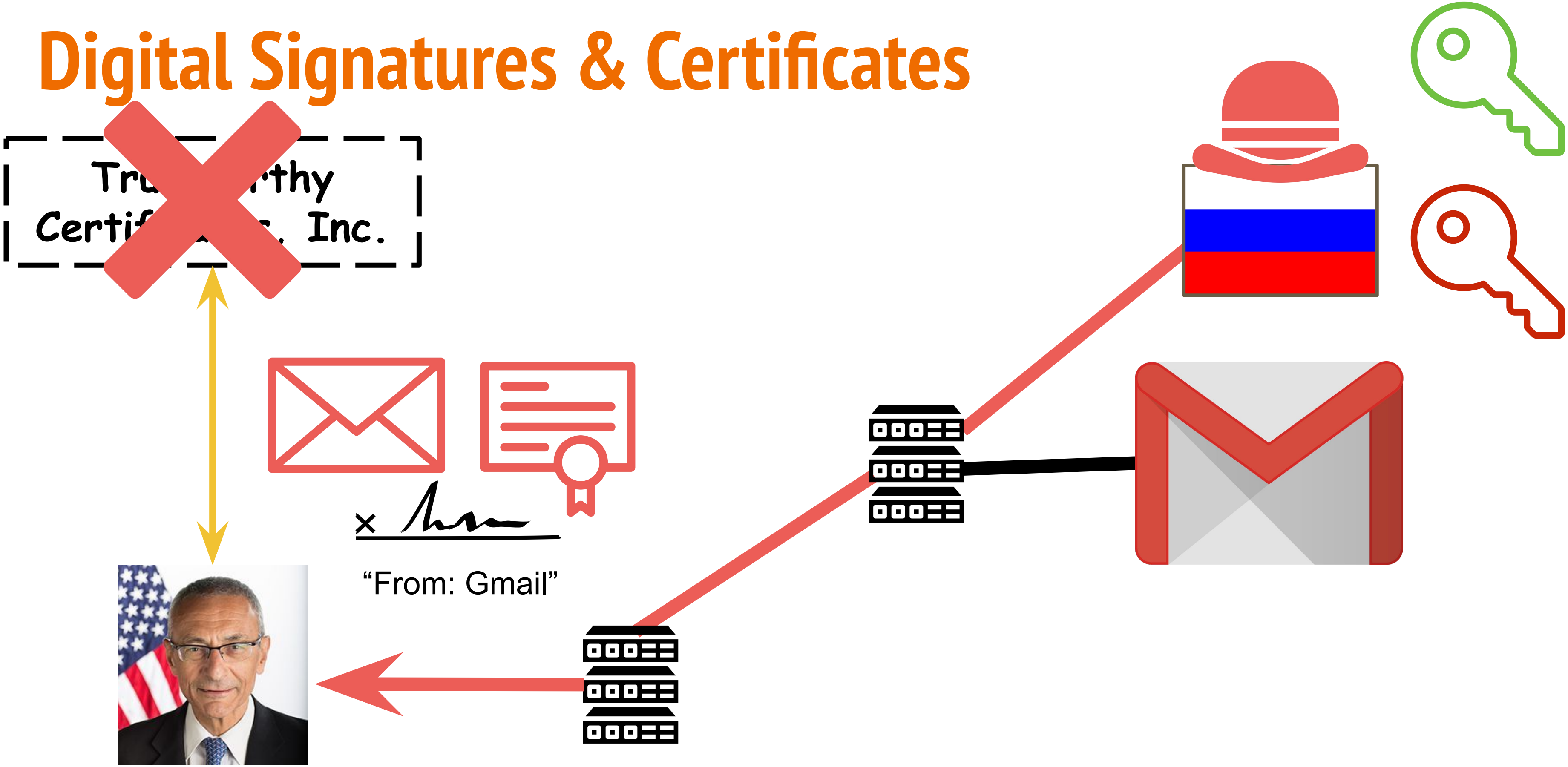
Images: Wikimedia

Digital Signatures & Certificates



Images: Wikimedia

Digital Signatures & Certificates



Images: Wikimedia

One more thing … best password?

GMw89#hUPn_d>k

horse_correct_bat

One more thing … best password?

`GMw89#hUPn_d>k`

`horse_correct_bat`

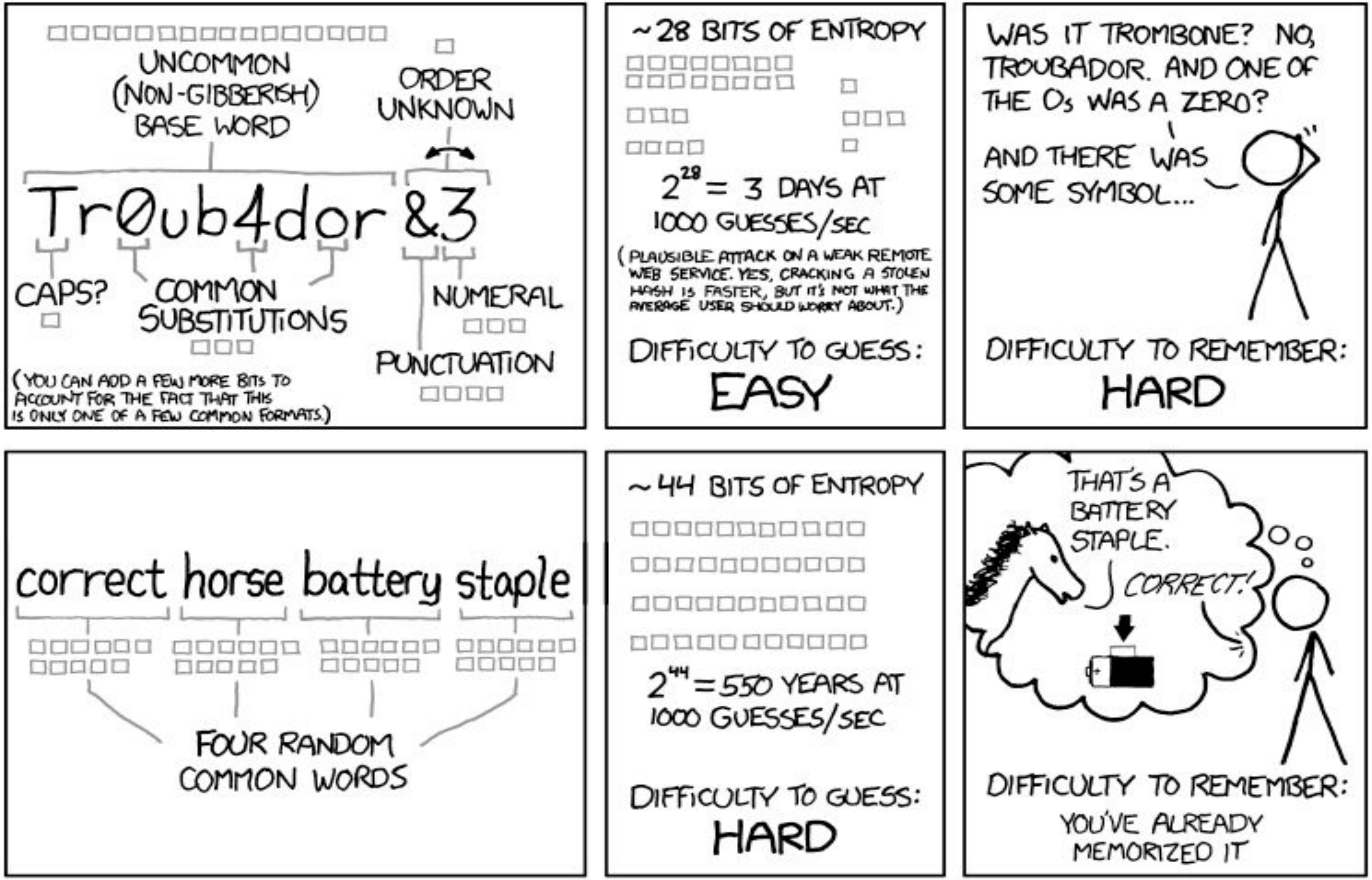

AJUSTE
2 3 4 5 6 7 8 9
A Z E R T Y U I O P
Q S D F G H J K L M
W X C V B N

One more thing ... best password?

GMw89#hUPn_d>k (72.0 bits of entropy)

horse_correct_bat (74.3 bits of entropy)

Horse_correct_bat (82.6 bits of entropy)


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT

III. Too Big to Fail

What happens when the attacker is someone we're supposed to trust?

Things We've Been Trusting

- The banks with which we store our money.
- The tech companies with which we send messages and share files with friends/family.
- The stores we shop from.
- The list of trusted certificate authorities from our web browser.

Things We've Been Trusting

- **The banks with which we store our money.**
- The tech companies with which we send messages and share files with friends/family.
- The stores we shop from.
- The list of trusted certificate authorities from our web browser.

Can’t Trust the Banks

Can’t Trust the Banks

The New York Times

ASIA PACIFIC

Bangladesh Bank Chief Resigns After Cyber Theft of $81 Million

By **RICK GLADSTONE** MARCH 15, 2016

Can't Trust the Banks

The New York Times

ASIA PACIFIC

Bangladesh Bank Chief Resigns After Cyber Theft of $81 Million

By RICK GLADSTONE MARCH 15, 2016

THE WALL STREET JOURNAL.

Subscribe | Sign In

CENTRAL BANKS

FBI Suspects Insider Involvement in $81 Million Bangladesh Bank Heist

Computer hackers tried to steal nearly $1 billion in a brazen attack

Can’t Trust the Banks

THE WALL STREET JOURNAL.

Subscribe | Sign In

The New York Times

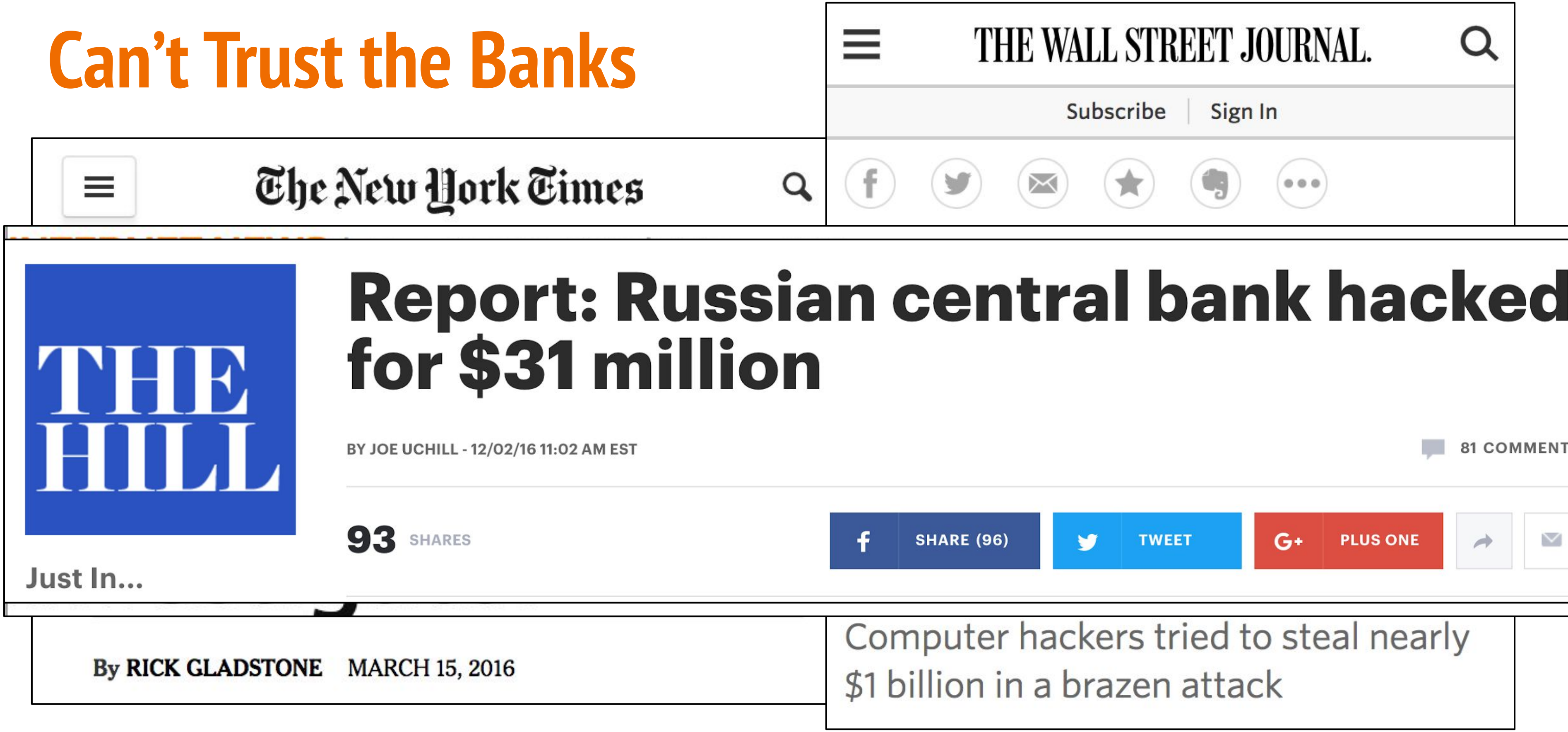
INTERNET NEWS | Mon Dec 12, 2016 | 7:15pm EST

Exclusive: Some Bangladesh Bank officials involved in heist - investigator

By RICK GLADSTONE MARCH 15, 2016

Computer hackers tried to steal nearly $1 billion in a brazen attack

Can't Trust the Banks



THE WALL STREET JOURNAL.

Subscribe | Sign In

The New York Times

THE HILL

Report: Russian central bank hacked for $31 million

BY JOE UCHILL - 12/02/16 11:02 AM EST

81 COMMENT

93 SHARES

SHARE (96) TWEET PLUS ONE

Just In...

By RICK GLADSTONE MARCH 15, 2016

Computer hackers tried to steal nearly $1 billion in a brazen attack

Things We've Been Trusting

- **The banks with which we store our money.**
- The tech companies with which we send messages and share files with friends/family.
- The stores we shop from.
- The list of trusted certificate authorities from our web browser.

Things We've Been Trusting

- The banks with which we store our money.
- **The tech companies with which we send messages and share files with friends/family.**
- The stores we shop from.
- The list of trusted certificate authorities from our web browser.

Can’t Trust the Tech Companies

Can’t Trust the Tech Companies


theguardian

home › US | politics | world | opinion | ≡ all

US national security

Glenn Greenwald on security and liberty

NSA Prism program taps in to user data of Apple, Google and others

- Top-secret Prism program claims direct access to servers of firms including Google, Apple and Facebook
- Companies deny any knowledge of program in operation since 2007

Can’t Trust the Tech Companies

theguardian

home › US | politics | world

US national security

Glenn Greenwald on security and li

NSA Prism program
user data of Apple, G
others

- Top-secret Prism program claims
servers of firms including Google, A
- Companies deny any knowledge of program in operation since 2007

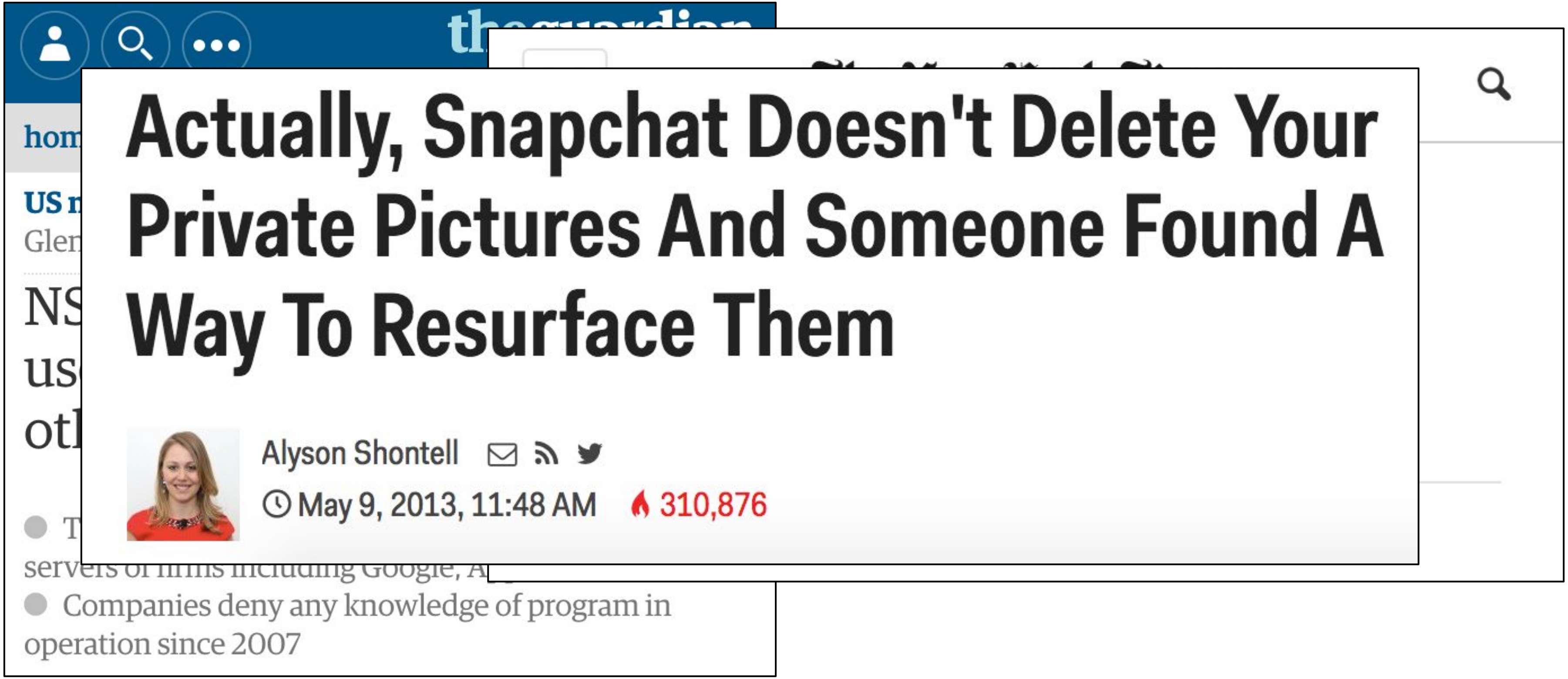
The New York Times

TECHNOLOGY

Yahoo Says 1 Billion User Accounts Were Hacked

By VINDU GOEL and NICOLE PERLROTH DEC. 14, 2016

Can’t Trust the Tech Companies



Things We've Been Trusting

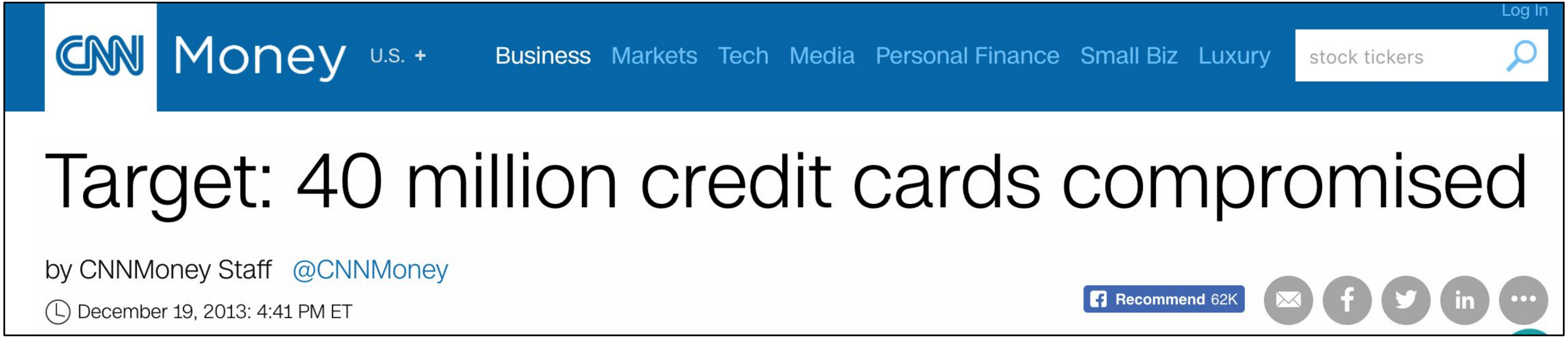
- The banks with which we store our money.
- **The tech companies with which we send messages and share files with friends/family.**
- The stores we shop from.
- The list of trusted certificate authorities from our web browser.

Things We've Been Trusting

- The banks with which we store our money.
- The tech companies with which we send messages and share files with friends/family.
- **The stores we shop from.**
- The list of trusted certificate authorities from our web browser.

Can’t Trust the Stores

Can’t Trust the Stores



Can't Trust the Stores

THE WALL STREET JOURNAL.

Subscribe | Sign In

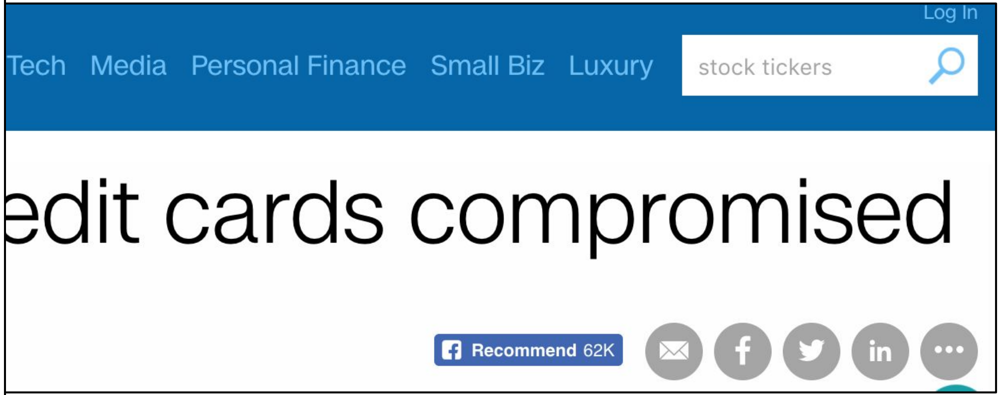
TECH

Home Depot Hackers Exposed 53 Million Email Addresses

Hackers Used Password Stolen From Vendor to Gain Access to Retailer's Systems

By SHELLY BANJO

Updated Nov. 6, 2014 8:03 p.m. ET



Log In

Tech Media Personal Finance Small Biz Luxury

stock tickers

edit cards compromised

Recommend 62K

Can’t Trust the Stores

Things We've Been Trusting

- The banks with which we store our money.
- The tech companies with which we send messages and share files with friends/family.
- **The stores we shop from.**
- The list of trusted certificate authorities from our web browser.

Things We've Been Trusting

- The banks with which we store our money.
- The tech companies with which we send messages and share files with friends/family.
- The stores we shop from.
- **The list of trusted certificate authorities from our web browser.**

Can’t Trust the CAs

Can’t Trust the CAs

The New York Times

Hacking in the Netherlands Took Aim at Internet Giants

By THE ASSOCIATED PRESS SEPT. 5, 2011

AMSTERDAM (AP) — Attackers who hacked into a Dutch Web security firm have issued hundreds of fraudulent security certificates for intelligence agency Web sites, including the C.I.A., as well as for Internet giants like Google, Microsoft and Twitter, the Dutch government said on Monday.

Can’t Trust the CAs


The New York Times

The Hacker News™
Security in a serious way

Chinese Certificate Authority 'mistakenly' gave out SSL Certs for GitHub Domains

Monday, August 29, 2016 Swati Khandelwal

Dutch Web security firm
tes for intelligence
Internet giants like
nt said on Monday.

Can't Trust the CAs

Chinese
'mistake
Certs fo
Monday, Augus

ars TECHNICA
BIZ & IT TECHNICA SCIENCE POLICY CARS GAMING &

RISK ASSESSMENT —

Google Chrome will banish Chinese certificate authority for breach of trust [Updated]

Draconian move follows the issuance of certificates masquerading as Google domains.

DAN GOODIN - 4/1/2015, 11:55 PM

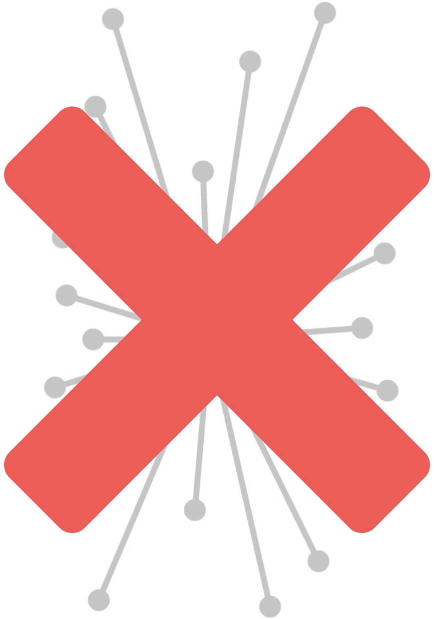
Things We've Been Trusting

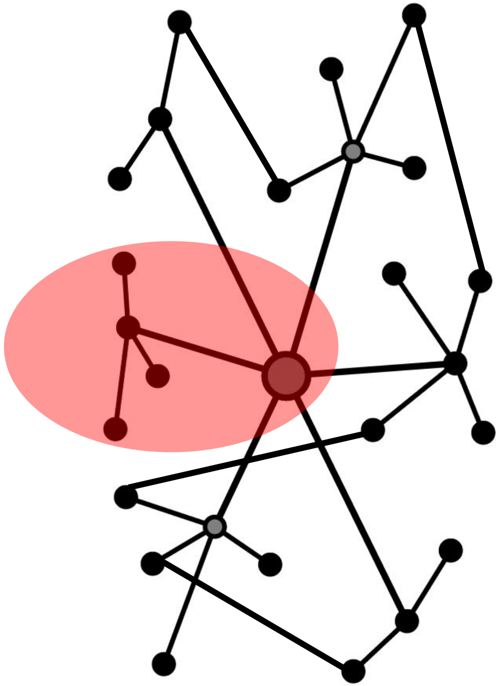
- The banks with which we store our money.
- The tech companies with which we send messages and share files with friends/family.
- The stores we shop from.
- **The list of trusted certificate authorities from our web browser.**

Things We've Been Trusting

- The banks with whic[illegible]store our mone[illegible]
- The tech companies [illegible]we[illegible]ages and share files with friends/family.
- The stores we shop from.
- **The list of trusted [illegible]auth[illegible]m our web browser.**

Decentralized Networks


A


B

Real Decentralized Technologies

Internet

Bitcoin

End of First Session (yay 🥳!)

Thank you for your attention and participation

Get some rest, review the material, and we'll see you in our next class 👋

Blockchain, Cryptocurrencies & Digital Tokens Demystified

Fall 2023 (EMBA)
Columbia Business School

Welcome Back to Session 2 🎉

Curriculum Roadmap

	Nov 4	Nov 18	Dec 2	Dec 9
Morning	Networks & Protocols	Hashing, Hashing Tables & One- Way Functions & a few more tech	Bitcoin + other forms of crypto payments and store of value mechanisms and media	DeFi & Other Applications (Digital Tokens, CBDC, etc.) + Speaker: Future of Finance + Discussion Forum
	Lunch	Lunch	Lunch	Lunch
Afternoon	Encryption & Cryptography (plus some math!)	**Bring it All Together:** Let's build a blockchain & discuss variety of cases	Ethereum & Other Digital Tokens + Speaker: Regulatory & Legal Considerations in Blockchain & Digital Assets	Governance, Marketplaces, NFTs & More; Final Lecture on How the Future May Play Out + Final Presentations

Class Schedule - Nov 4, Nov 18, Dec 2, Dec 9

Class Plan

Nov 4	08:30 am to 6:45 pm (K-440)	Module 1 + 2
Nov 18	08:30 am to 6:45 pm (K-440)	Module 3 + 4
Dec 2	08:30 am to 6:45 pm (K-440)	Midterm Project + 5 & 6 + Guest Speaker
Dec 9	08:30 am to 6:45 pm (K-440)	Module 7 & 8 + Guest Speaker + final presentations

Daily Schedule

8:30-9:45 am	**Lecture**
9:45-10:00 am	*Break*
10:00-11:15 pm	**Lecture**
11:15 am-12:30 pm	Lunch (1h15min) - Kravis 2nd floor (Smith Dining)
12:30-2:00 pm	**Lecture**
2:00-2:15 pm	*Break*
2:15-3:30 pm	**Lecture**
3:30-3:45 pm	*Break*
3:45-5:00 pm	**Lecture**
5:00-5:15 pm	*Break*
5:15-6:45 pm	**Lecture**

Important Admin Items for the Day

- Team formations finalized today, ideally by 3:30 pm and no later than end of day today
- Details on your midterm project
- Thoughts on “Blockchain Killer App” for Sessions 3 and/or 4
- Make sure not to fall behind as Sessions 1 & 2 are foundational
- Watch lecture recordings and email me for office hours
- I REALLY enjoyed our first session, and thank you VERY much for the amazing level of participation and engagement. Let’s hope today would be equally fun, if not more 😃
- ... btw, did you watch The Simpsons episode right after our first class session? It was about blockchain & NFTs!! Check out S35E5.

THE MOST Important Admin Item for the Day

THE MOST Important Admin Item for the Day

Catering today is by **Dinosaur BBQ**:

- Mac & cheese
- Turkey
- Beef brisket
- BBQ Salmon
- Portabella Mushrooms w/ peppers & onions
- Simmered Greens
- Sweet Potatoes

Before we begin, any interesting points or lessons from our first session you'd like to share?

Let's start our Session 2

History of Cryptographically-based e-Currencies:

It’s nothing new:

- Remember Error 402?

History of Cryptographically-based e-Currencies:

It's nothing new:

- Remember Error 402?
- **DigiCash**: proposed in 1983 by David Chaum, set up eCash, launched in 1989, declared bankruptcy in 1998
- **CyberCash**: payment service founded in 1994, IPO in 1996, set up CyberCoin for micro-payments (through NetBill at CMU), went bankrupt in 2001
- Hashcash: proposed in 1997 by Adam Back,
- **BitGold**: proposed by Nick Szabo in 1998 (he coined "Smart Contracts.") Although never implemented, it has many similarities to Bitcoin!
- ... and others (**Hashcash**, **B-Money**, **First Virtual**, etc.)

Why did these early forms of digital currencies fail?

Double-Spending, Trust, and Consensus are amongst the top reasons …

Speaking of consensus …

Byzantine Generals Problem &
the question of Byzantine Fault Tolerance

A seminal CS paper (1982)

The Byzantine Generals Problem

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE
SRI International

Reliable computer systems must handle malfunctioning components that give conflicting information to different parts of the system. This situation can be expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach agreement. It is shown that, using only oral messages, this problem is solvable if and only if more than two-thirds of the generals are loyal; so a single traitor can confound two loyal generals. With unforgeable written messages, the problem is solvable for any number of generals and possible traitors. Applications of the solutions to reliable computer systems are then discussed.

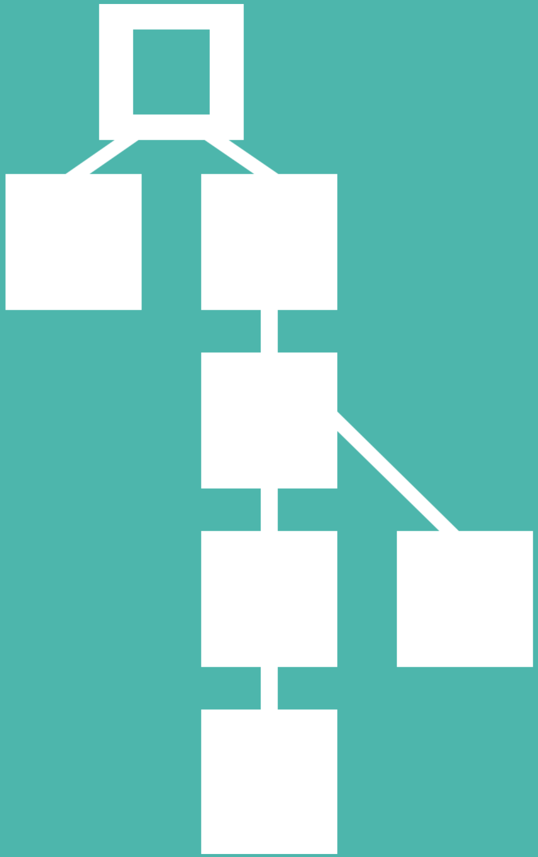
Categories and Subject Descriptors: C.2.4. [**Computer-Communication Networks**]: Distributed Systems—*network operating systems*; D.4.4 [**Operating Systems**]: Communications Management—*network communication*; D.4.5 [**Operating Systems**]: Reliability—*fault tolerance*

ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, Pages 382–401.

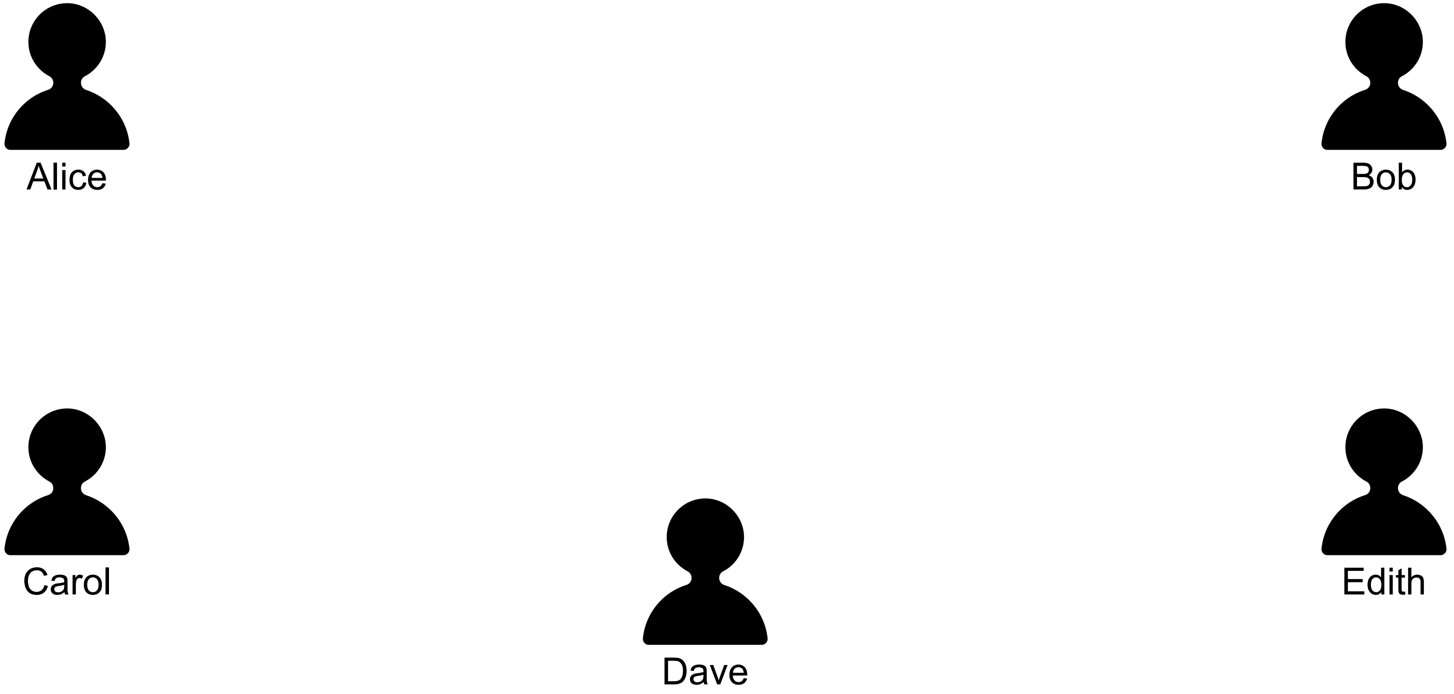
In a distributed network, how many node failures can the system tolerate and still function as intended in delivering consensus?

IV. Building the Blockchain

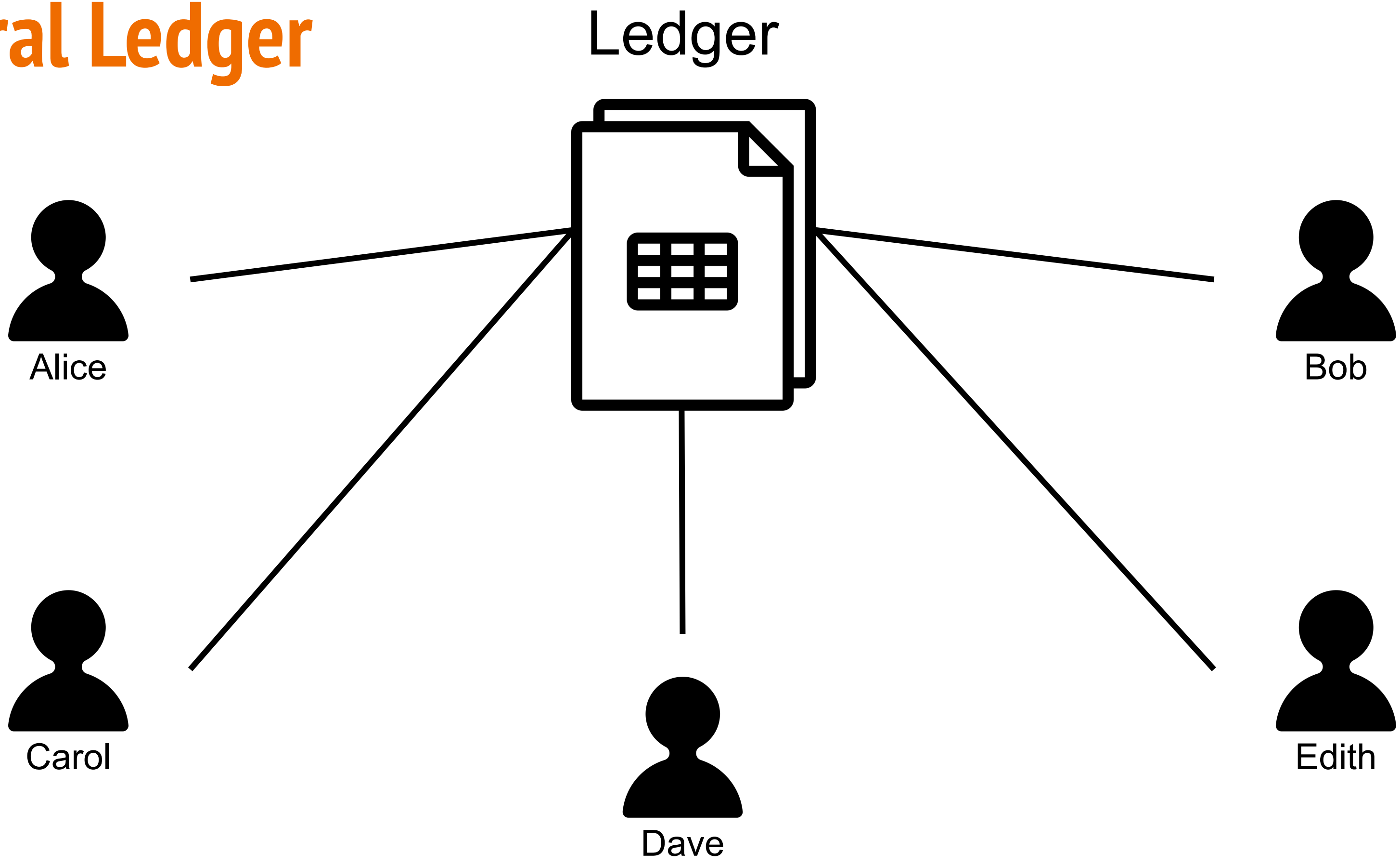
Using cryptography to build decentralized technologies.



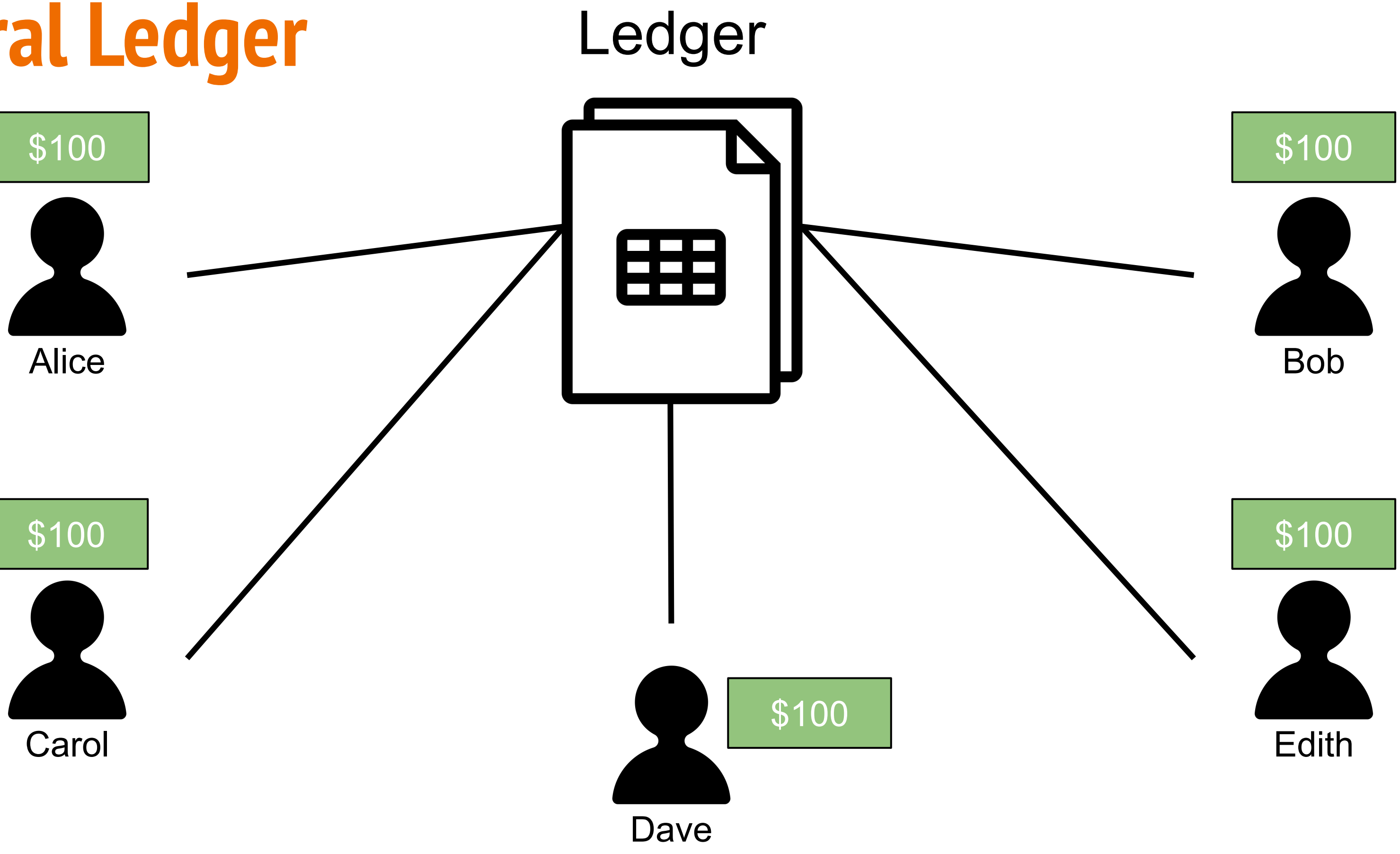
Blockchains



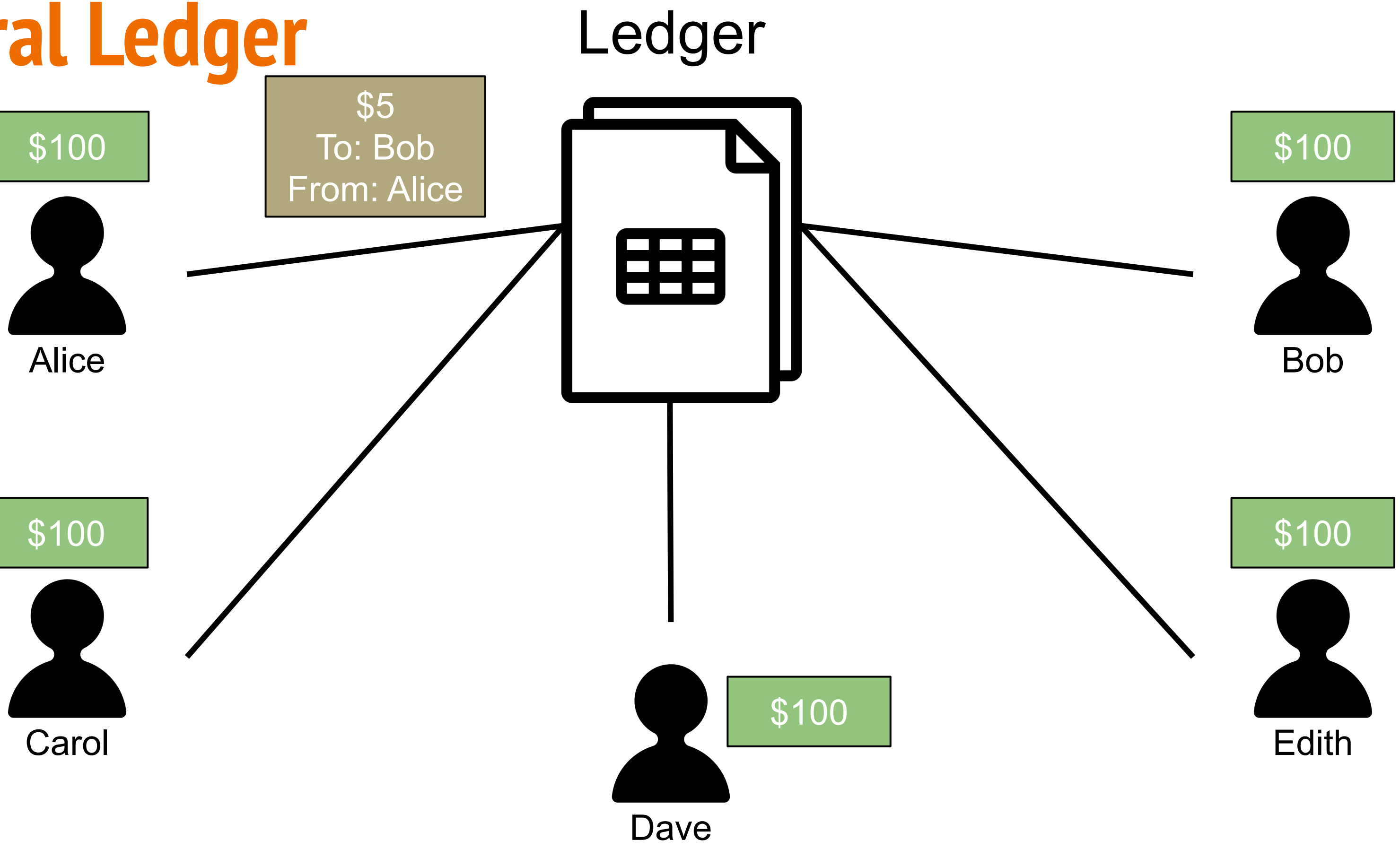
Central Ledger



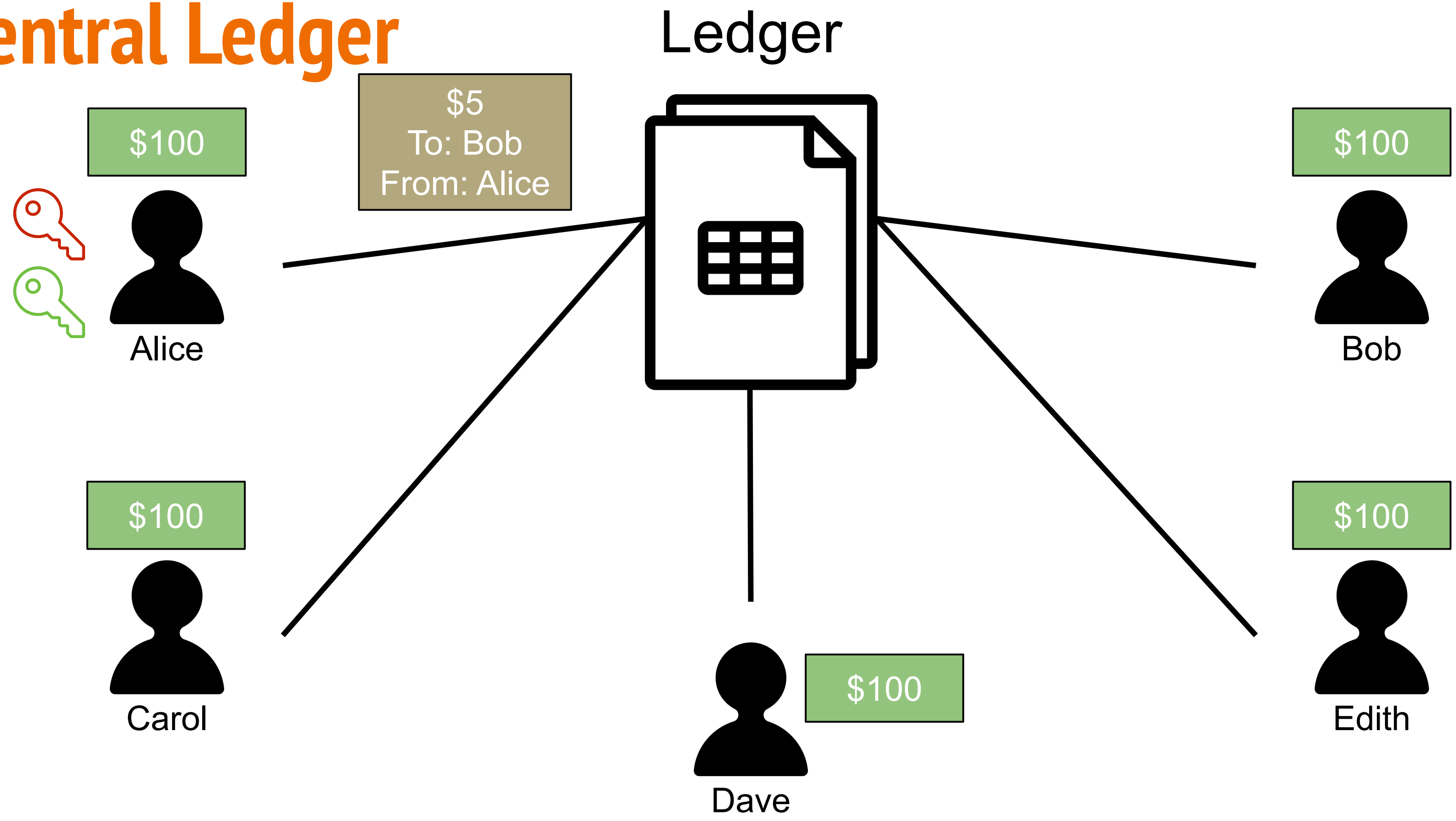
Central Ledger



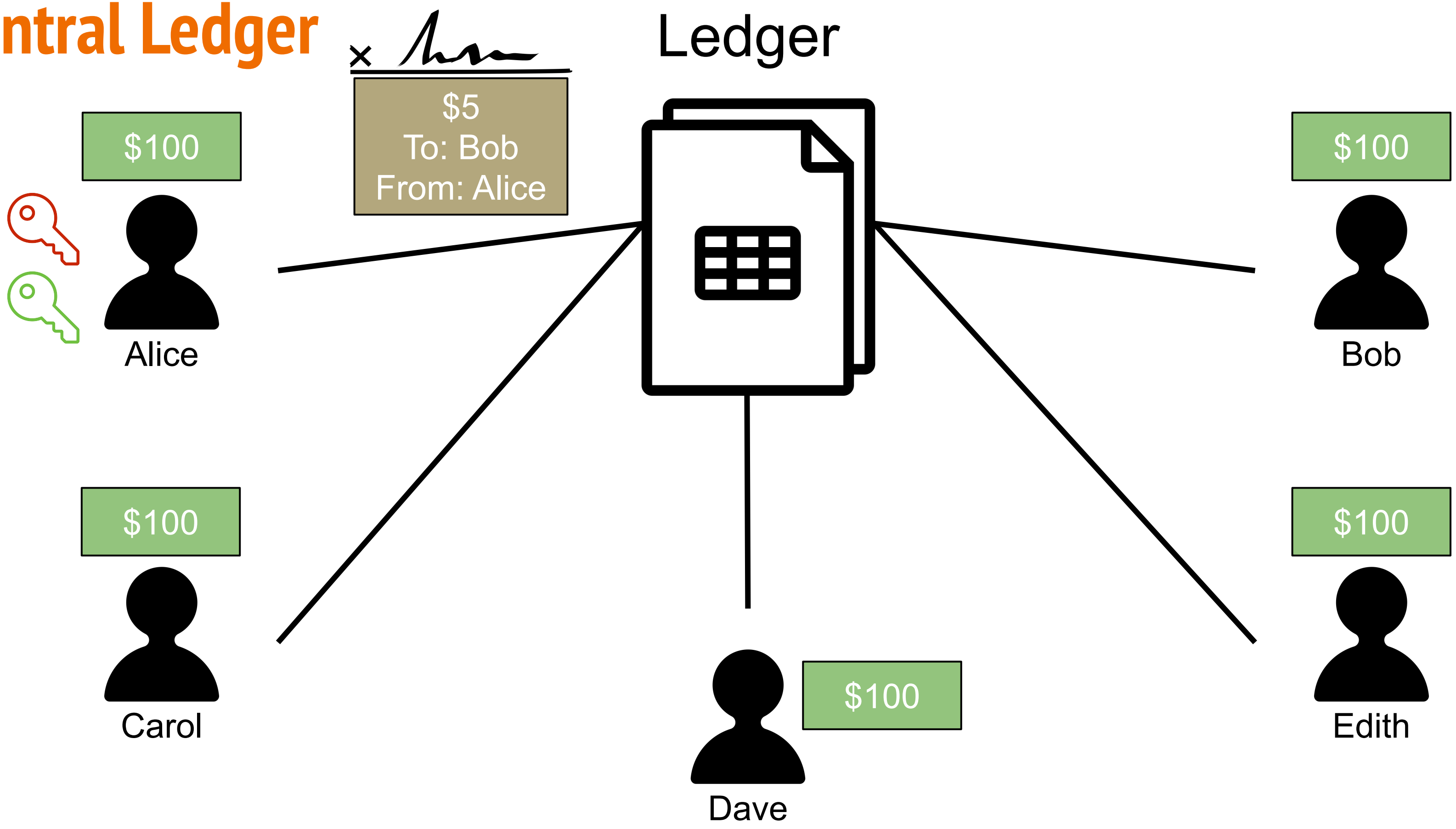
Central Ledger



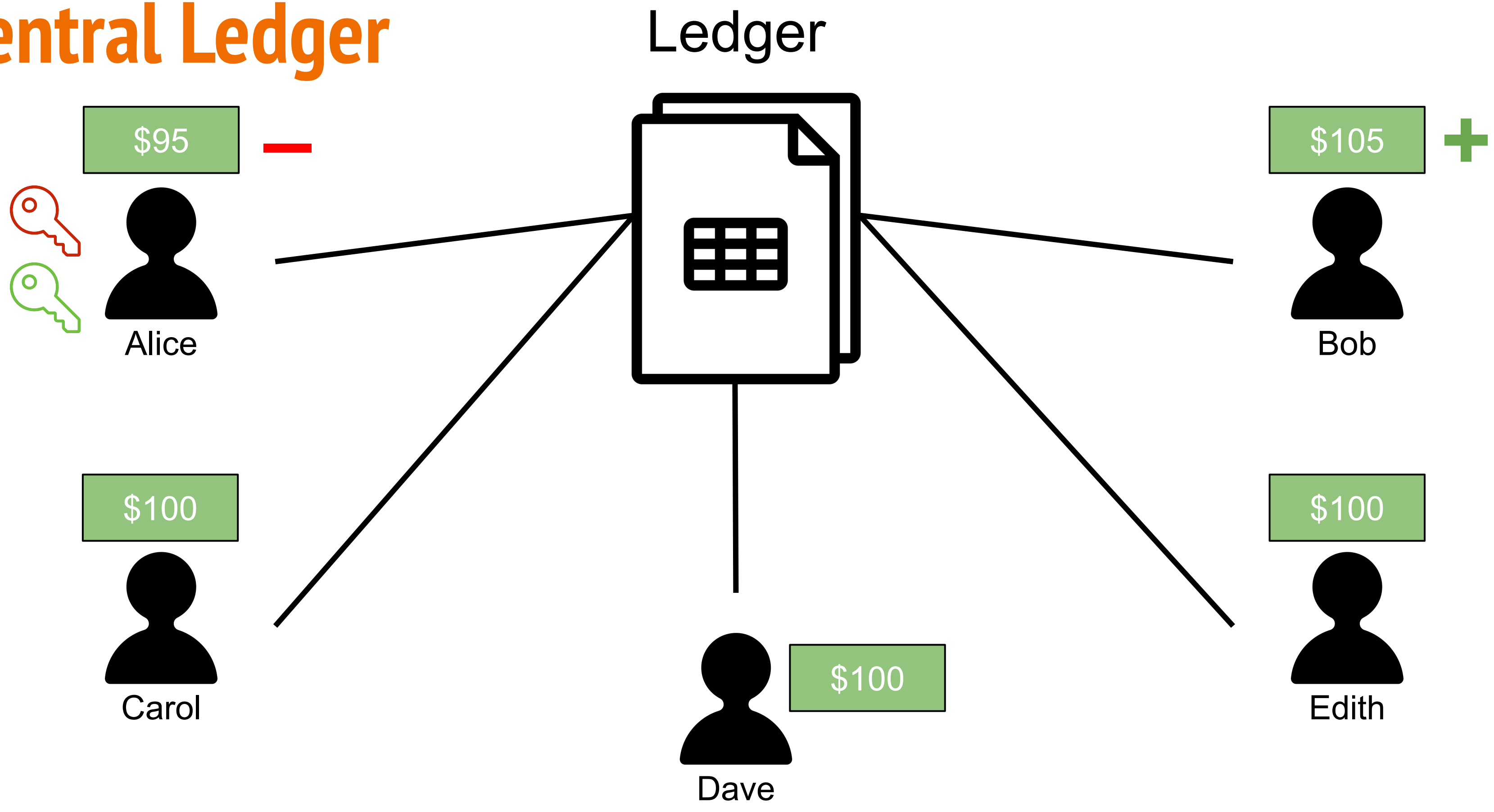
Central Ledger



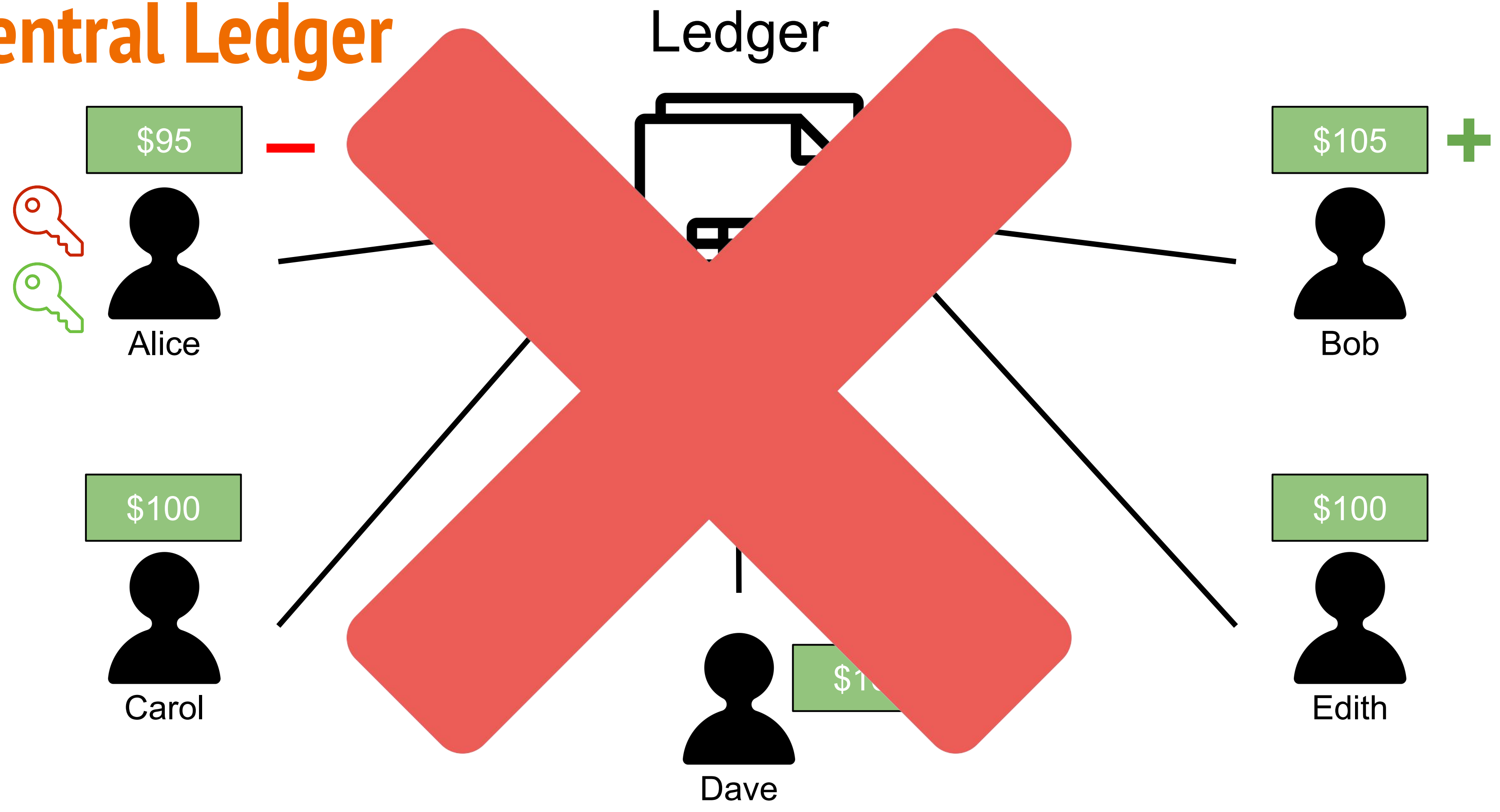
Central Ledger



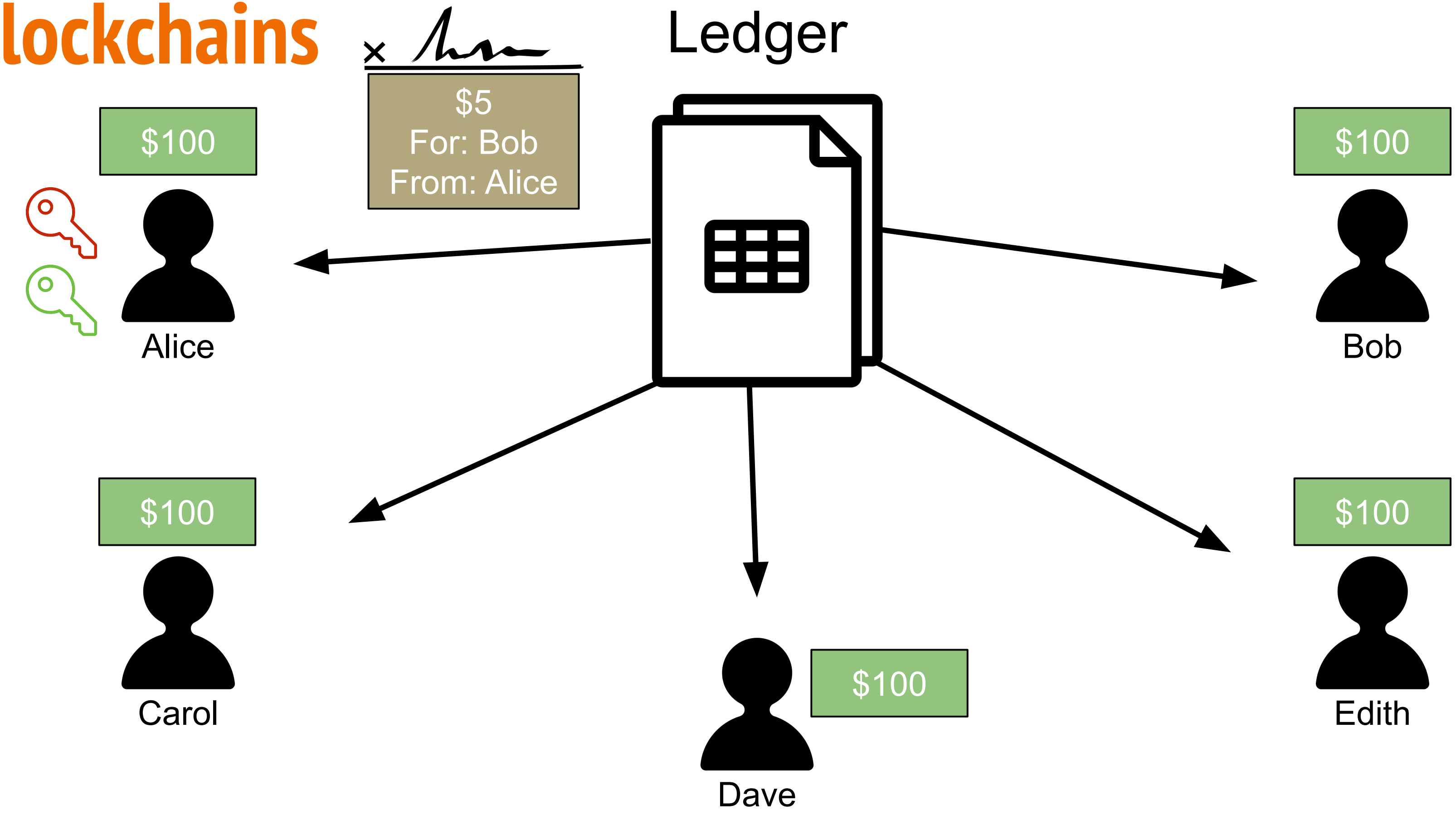
Central Ledger



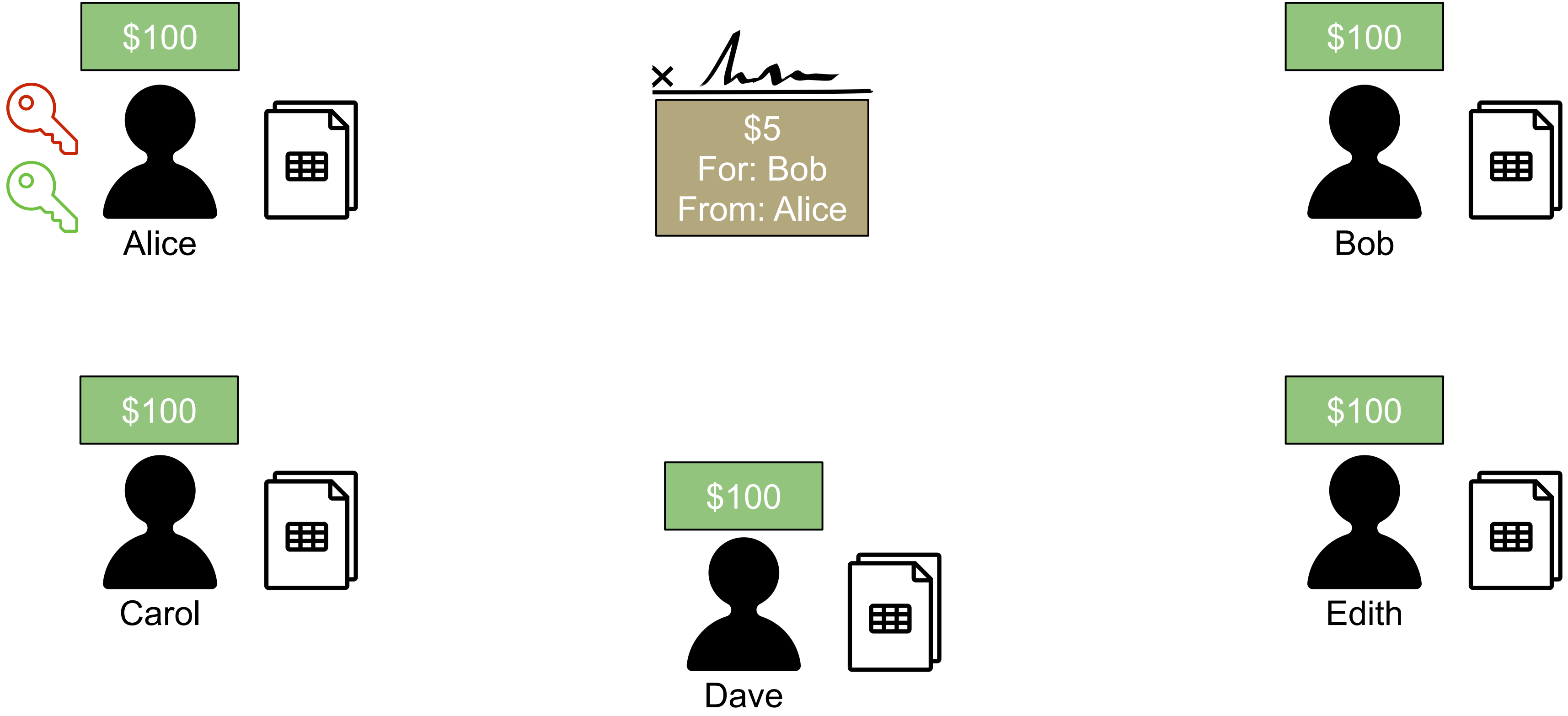
Central Ledger



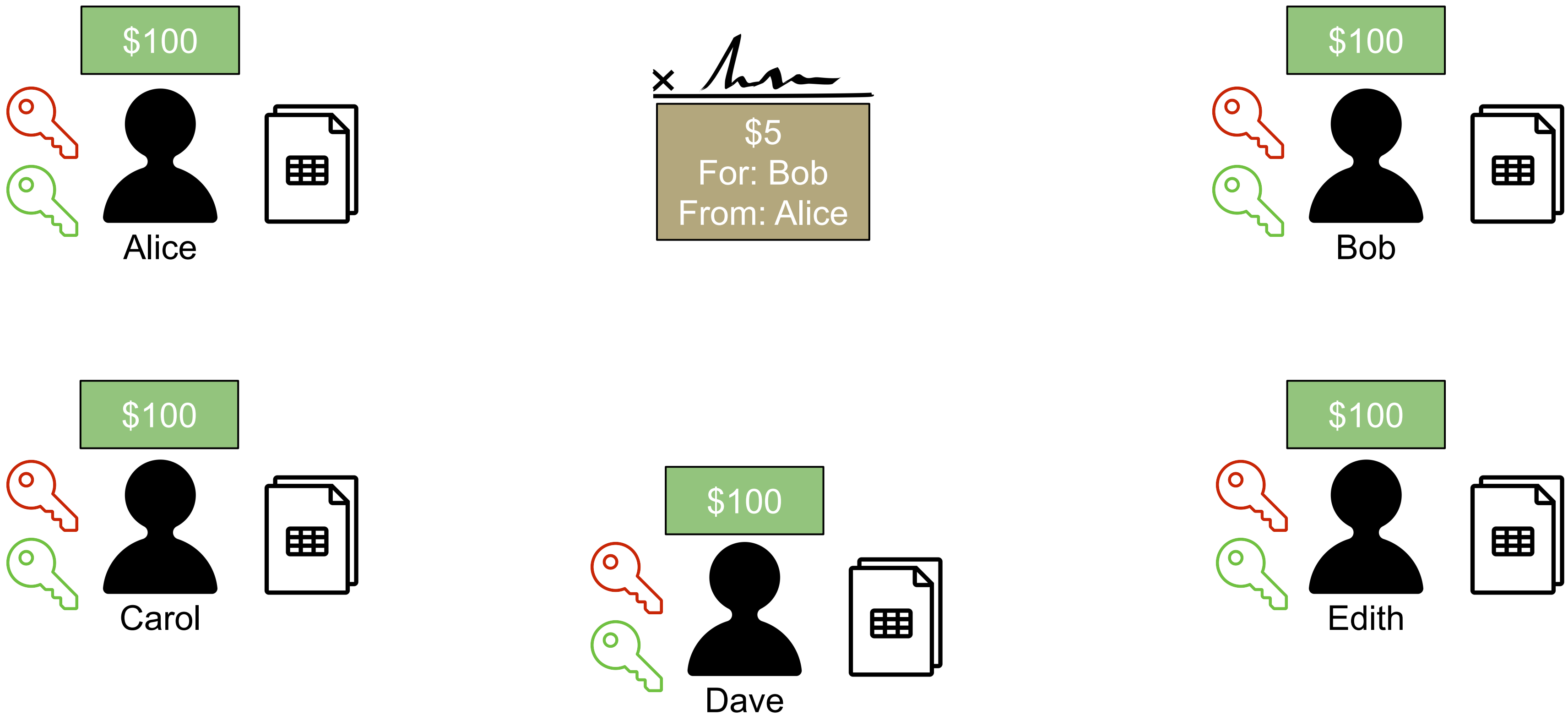
Blockchains



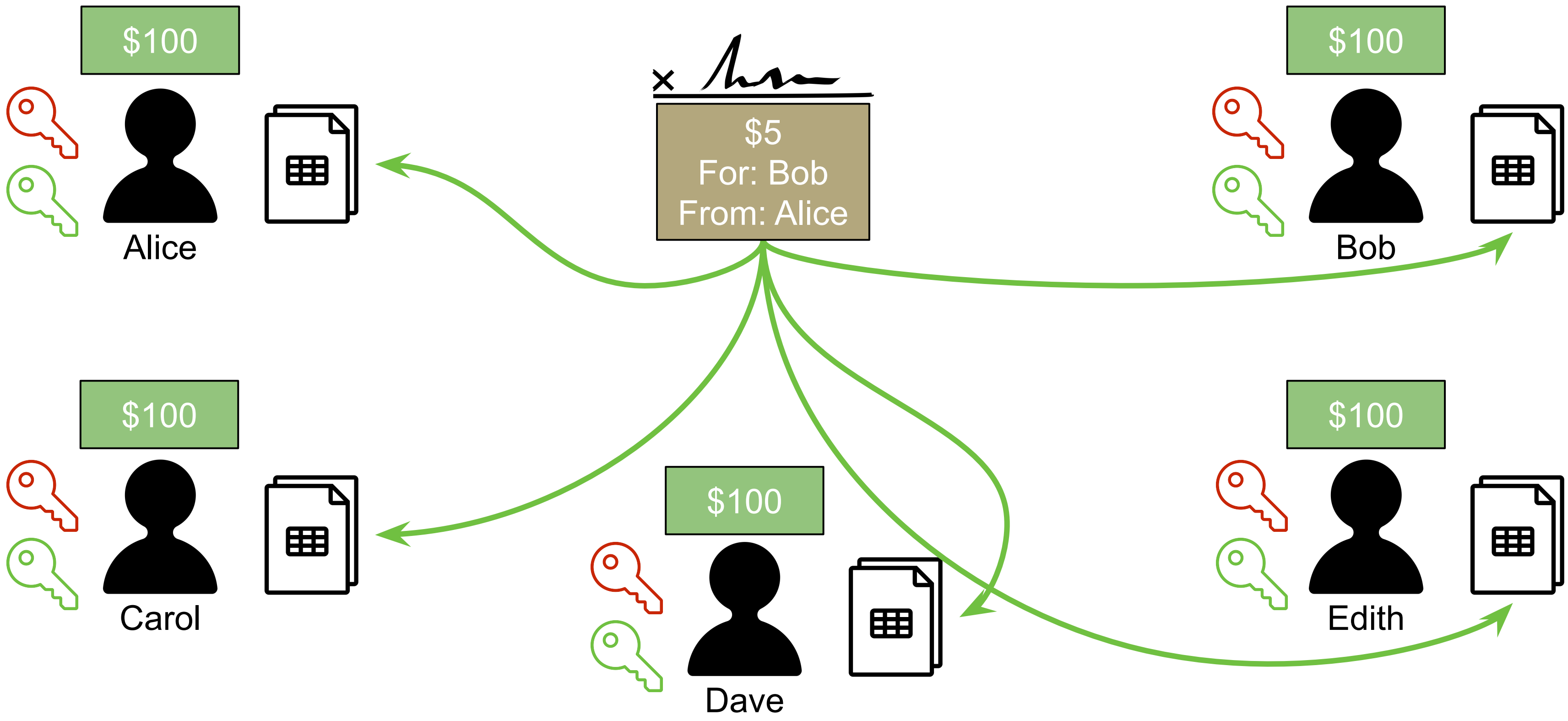
Blockchains



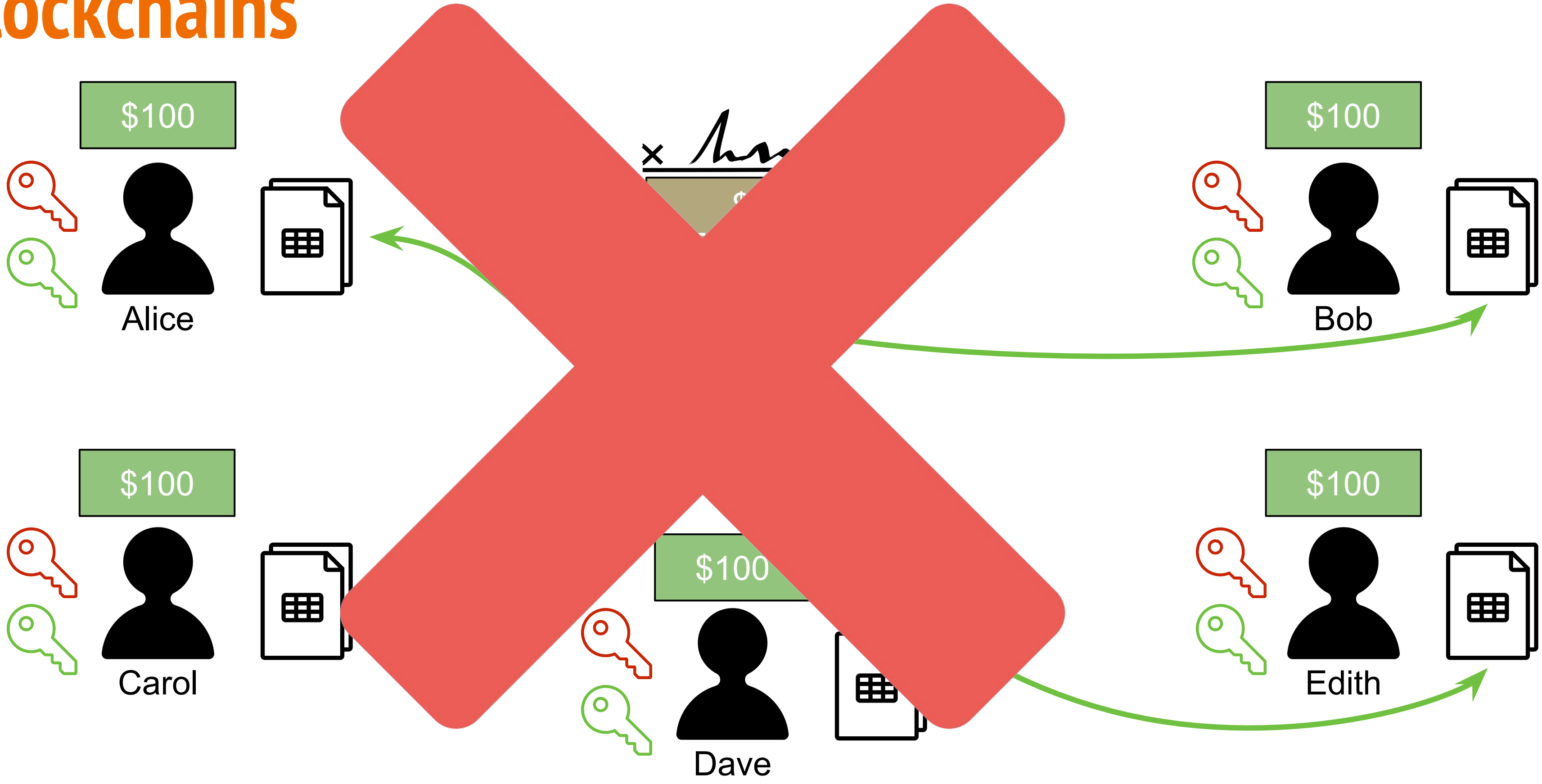
Blockchains



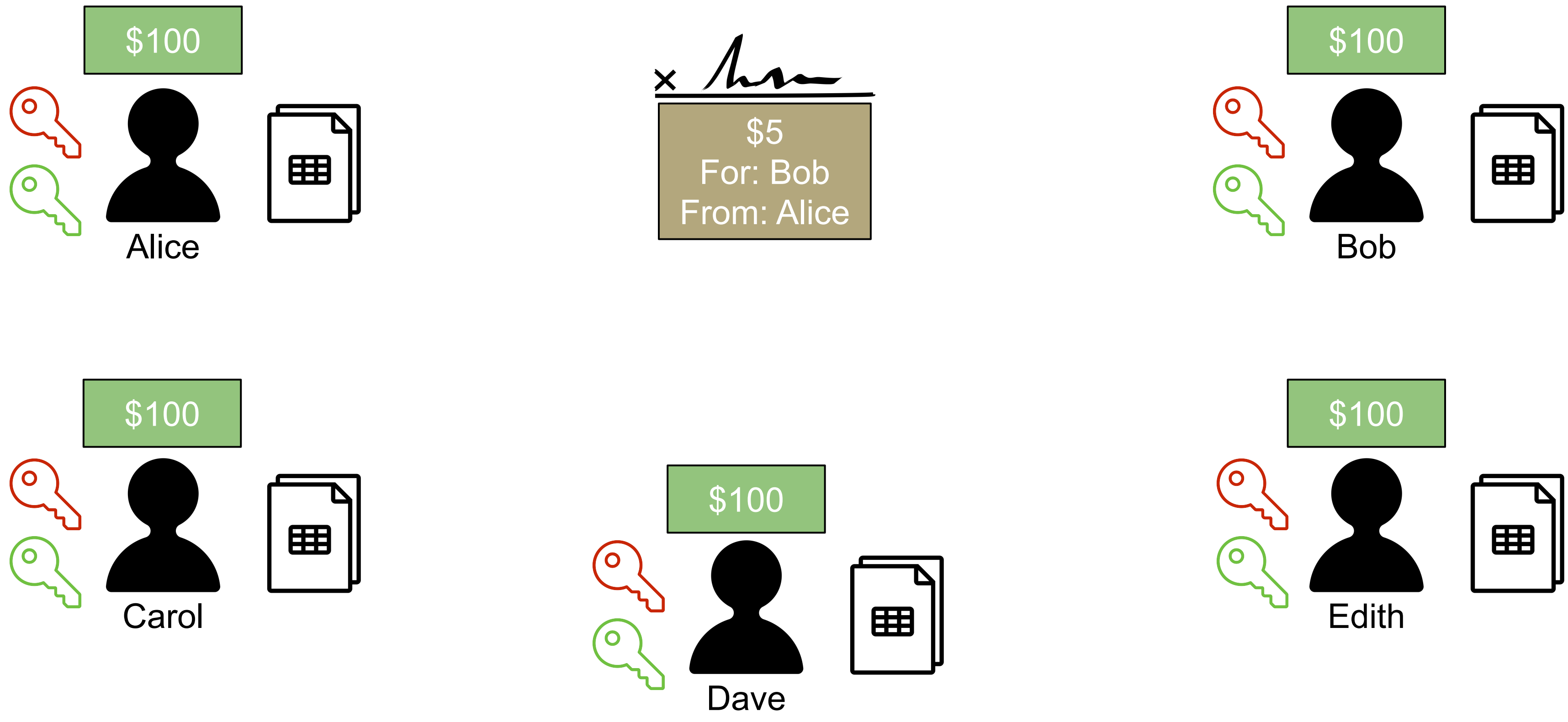
Blockchains: everyone updates on their own asap!



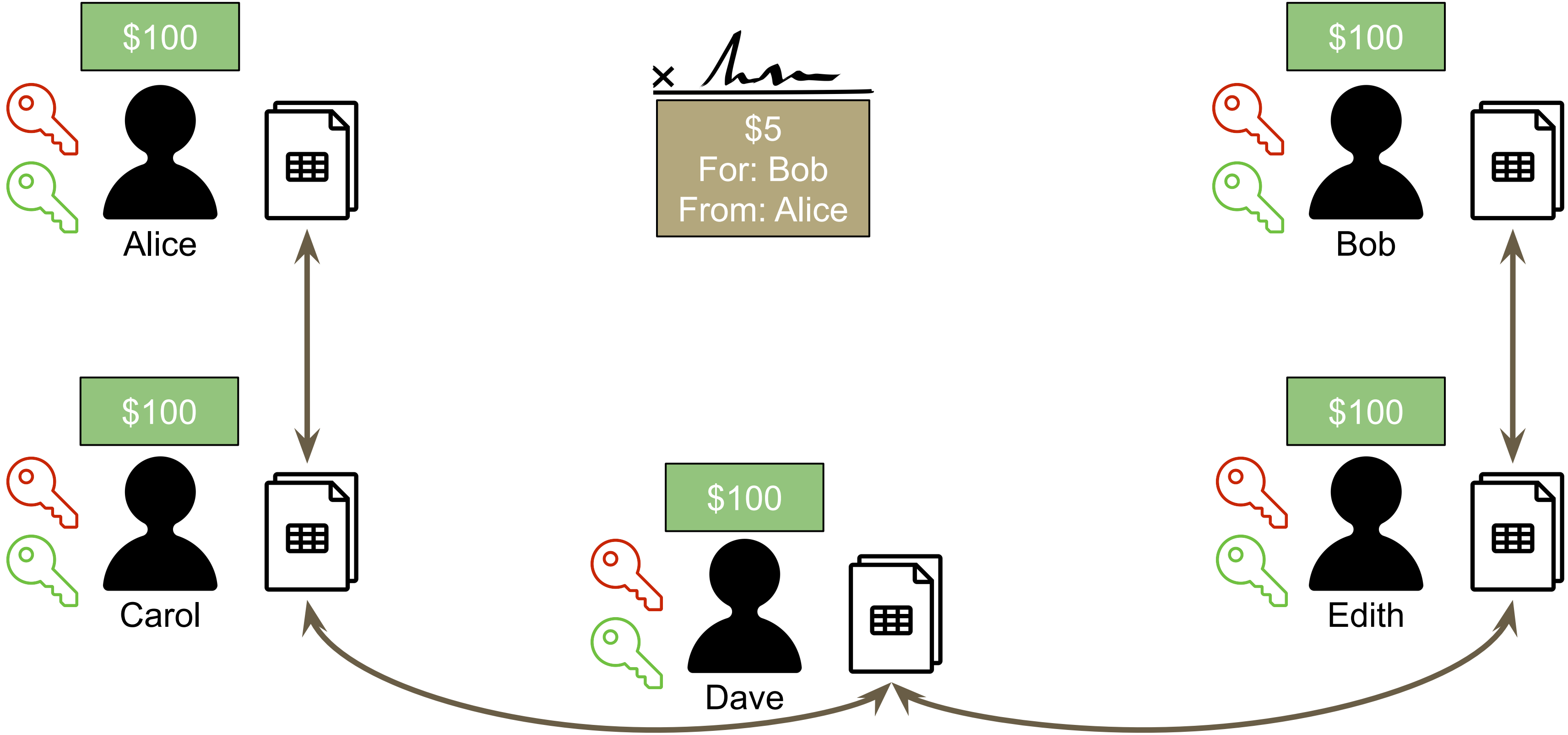
Blockchains



Blockchains: stay in sync with code and NO trust



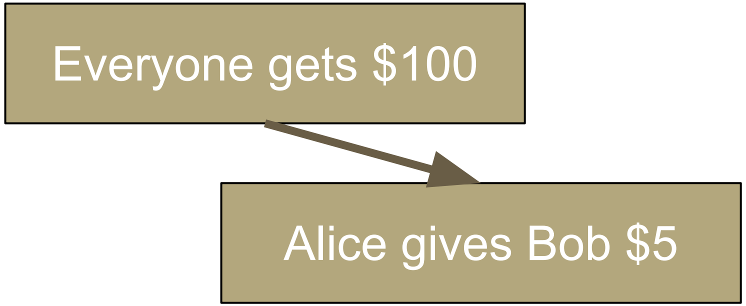
Blockchains: store in blocks chained together



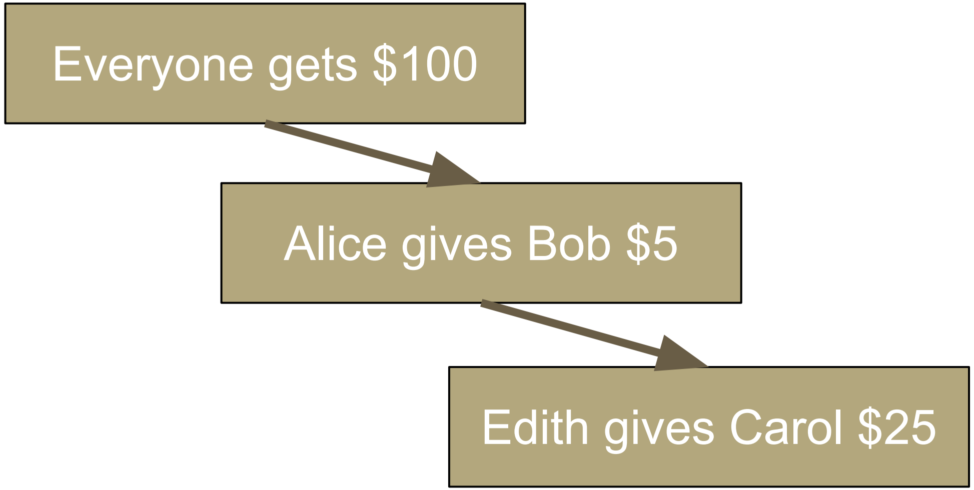
Blockchain: a cryptographically-verifiable Tx chain

Everyone gets $100

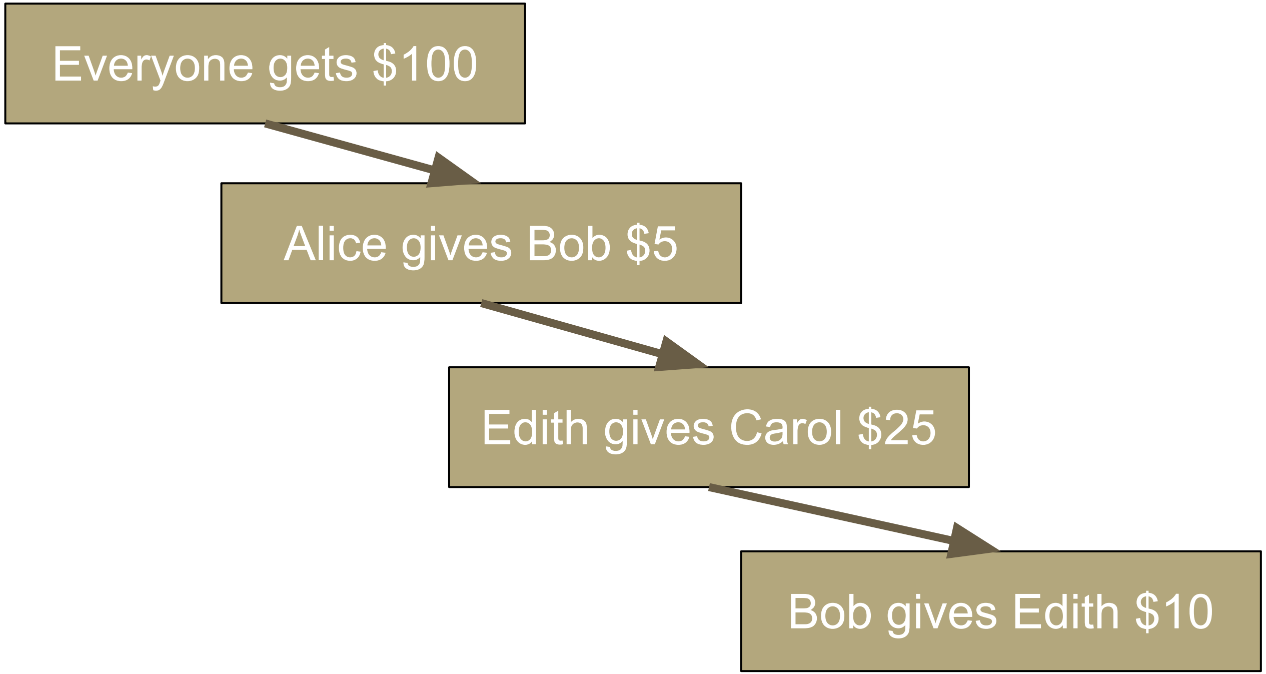
Blockchain: a cryptographically-verifiable Tx chain



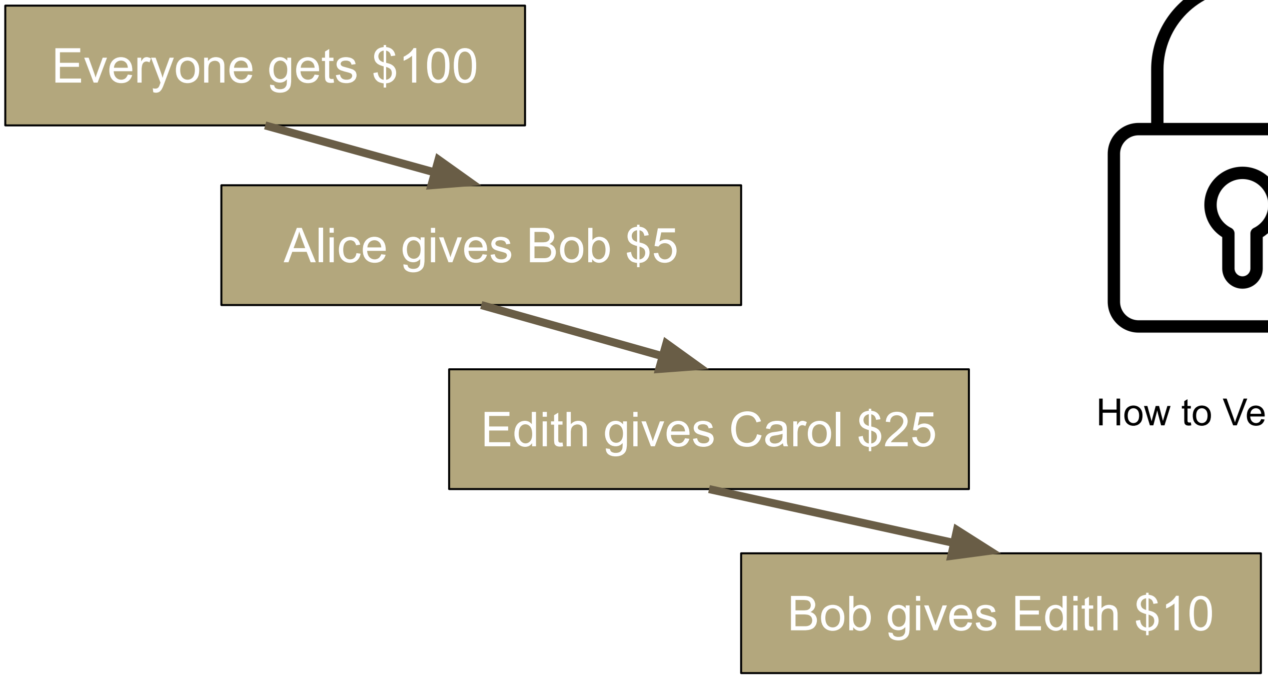
Blockchain: a cryptographically-verifiable Tx chain



Blockchain: a cryptographically-verifiable Tx chain



Blockchain: a cryptographically-verifiable Tx chain



How to Verify? How to Encrypt?

Once again, we need some simple math
(don't we love math by now?!)

Remember functions?

$$f(x) = x^2 + 8$$

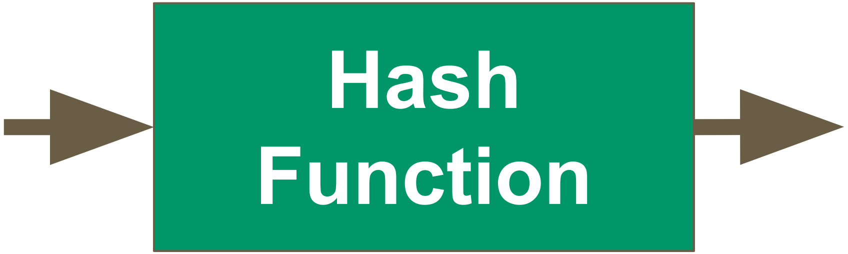
Functions in Math

- Simply put, a function is a (mathematical) operation ...
- ... one input equals to one output
- f(x) where x is the input value
- Example:
  - our function is “Doubling” →
  - $f(x) = 2x$ →
  - Take an input, then double it (or multiply by 2)
  - For x=**4 (i.e. input is 4)**, then the **output is 8**
- But then a funny thing happens ...

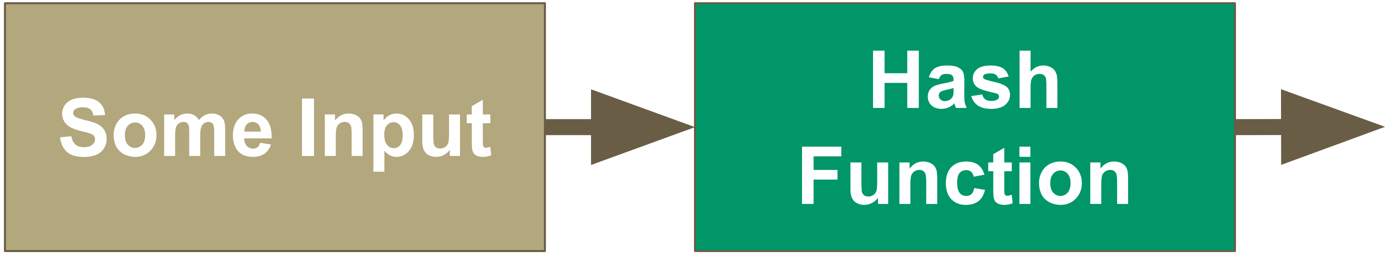
Functions in Math

- But then a funny thing happens ...
- ... our function is still “Doubling” →
- So what if I give you the output only? Can you figure out the input?
- OF COURSE ... we’ll just reverse the function
- Example:
    - our function is “Doubling” →
    - f(x) = 2x →
    - If the output is **44**, then the input is ...
    - **22** ;-)
- Most functions in math are Two-way Functions (reversible)
- But then ...

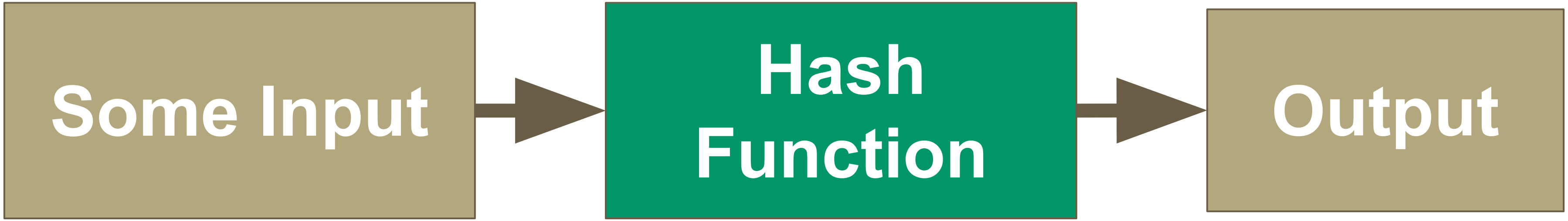
Hashing (One Way Functions)



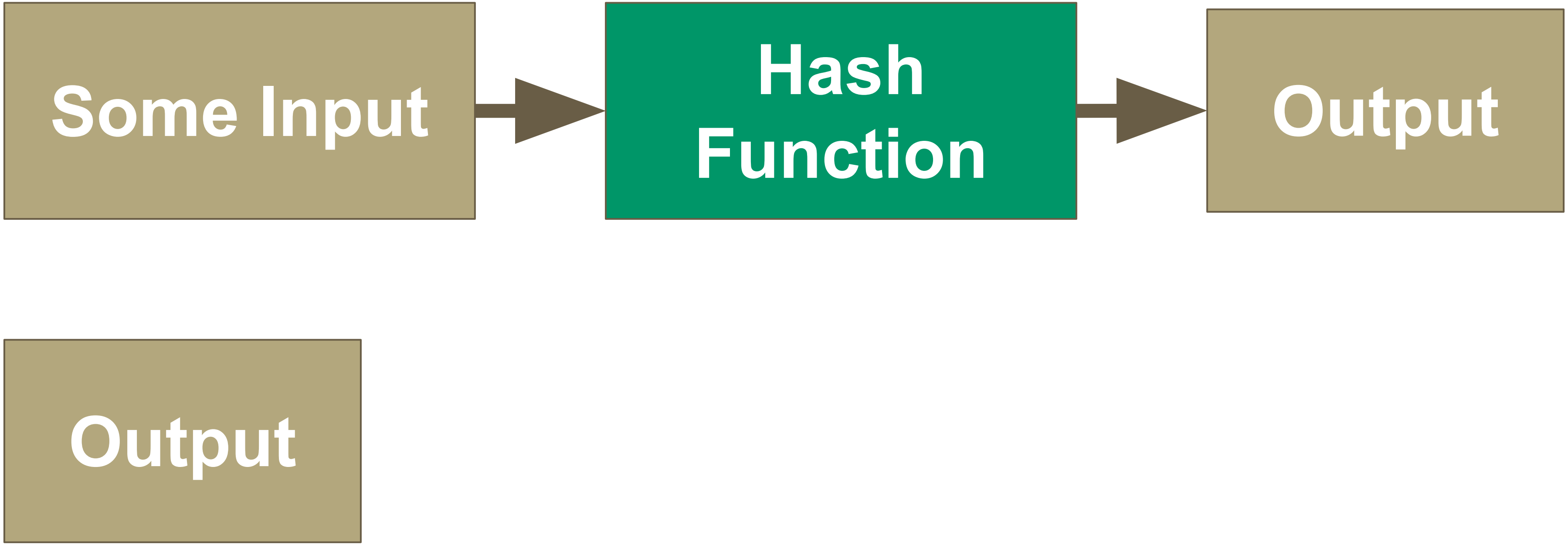
Hashing (One Way Functions)



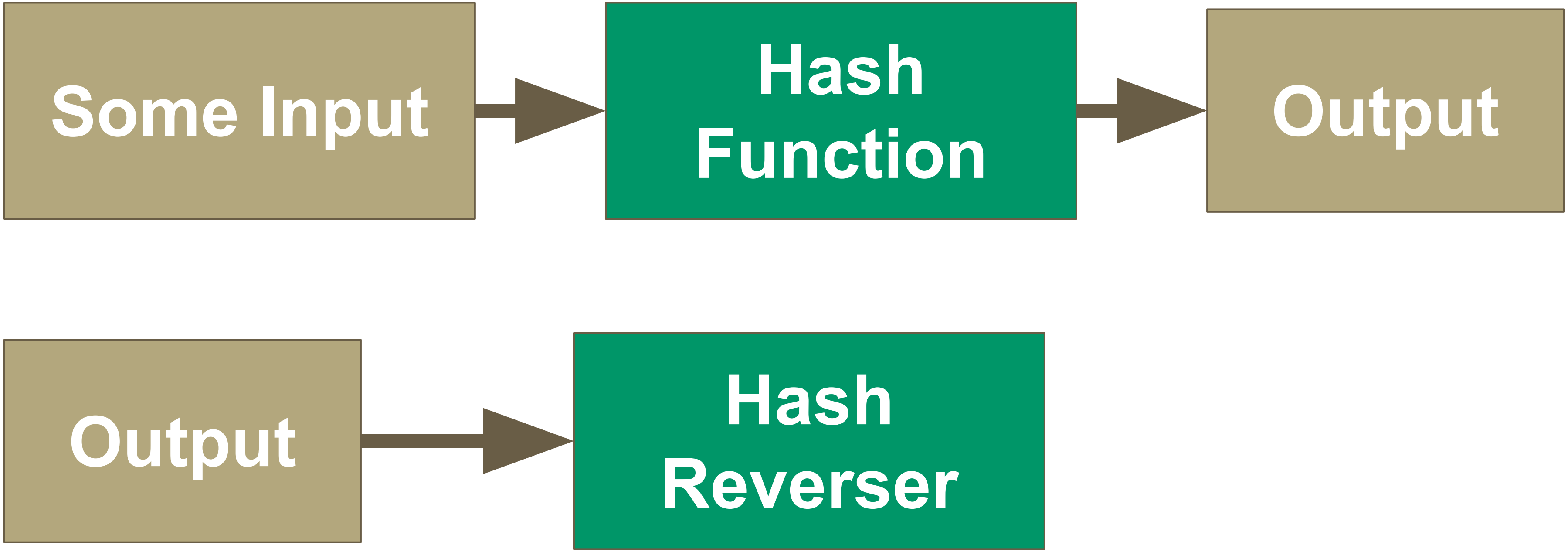
Hashing (One Way Functions)



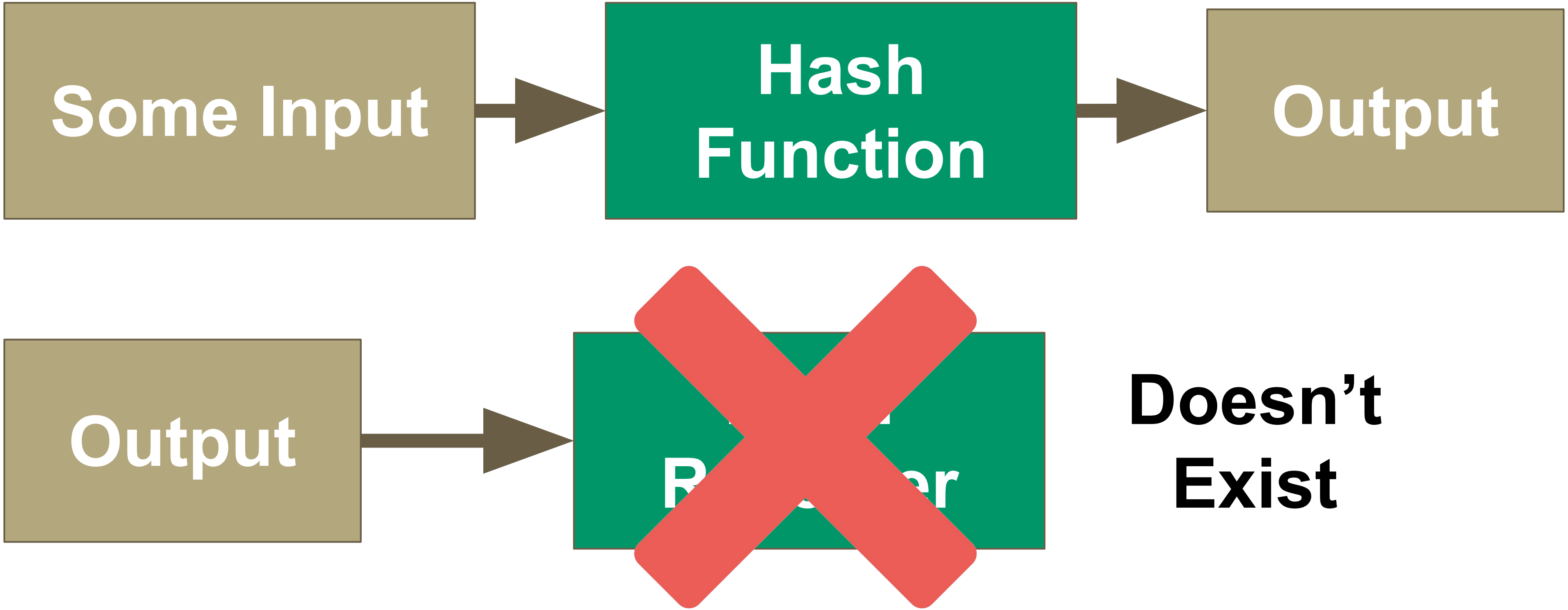
Hashing (One Way Functions)



Hashing (One Way Functions)



Hashing (One Way Functions)



A great example of a One-way Function …

Another great example of a One-way Function

…

Real-World One-Way Function (Hashing Function)

Real-World One-Way Function (Hashing Function)

Cartage New England Inc
26 Allen Ln Ipswich 01938....................978 356-9960
Cartagema Lydia
18 Jewett Ros 02131.................... 617 323-7639
Cartagena Avith
9 Bancroft Rox 02119....................617 442-9780
B Hyd 02136.................................. 617 361-5253
Jessica 50 Decatur Cha 02129..........617 241-0152
Lucilla 174 Harvard Cam 02139......... 617 491-5621
M 95 Rowe Ros 02131........................617 323-9713
Melvin 501 Green Cam 02139............617 576-1061
Carte Nicholas
18 Appleton Boston 02116..............617 695-6996
Cartegena O 4 Milford Bos 02118.........617 338-8219
Carten Thos J Sr & Claire
1 Paradise Rd Mil 02186................ 617 698-6163
Thomas & Kathleen
50 Thompson Ln Mil 02186..............617 696-6919
Carter A Ros 02131............................ 617 327-2257
A Roxbury.. 617 442-5230
A 31 Bethune Wy Roxbury 02119......... 617 442-1219
A 260 Putnam Av Cambridge 02139...... 617 492-4174
A M 255 Maschsts Av Bos 02115......... 617 266-7153

Carter F 24 Hillock Ros 02131..............617 327-1105
Faye & Ricky
357 Columbus Av Bos 02116............617 437-7331
Francis S 134 Temple W Rox 02132.. 617 323-6781
Franklin & Anne
221 Mt Auburn Cam 02138..............617 354-0798
Fred 42 Haverford Jam 02130............617 524-3078
Fred 96 Hinckley Rd Mil 02186...........617 698-1343
G & R 8 Verdun Dor 02124................617 436-8906
G T 27 Franklin Av Som 02145............617 623-7121
Gayle 25 Frontenac Dor 02124........... 617 825-0322
Geo S 115 Moss Hill Rd Jam 02130.....617 522-3215
George 125 Nashua Bos 02114..........617 367-9548
Carter Halliday Associate
107 S Street Bos 02111........................617 456-1689
Carter Harry F
26 Runng Brk Rd W Rox 02132......... 617 325-5465
Carter Hide Co Inc
146 Summer Bos 02110........................617 542-7987
Carter Hilary 61 Harvey Cam 02140.....617 876-2750
Horace
241 Walnut Av Roxbury 02119.........617 442-5307
Howard Jr 26 Notre Dme Rox 02119.617 445-5552

Real-World One-Way Function (Hashing Function)

SuperPages.com 195

Cartage New England Inc	26 Allen Ln Ipswich 01938	978 356-9960
Cartagema Lydia	18 Jewett Ros 02131	617 323-7639
Cartagena Avith	9 Bancroft Rox 02119	617 442-9780
B	Hyd 02136	617 361-5253
Jessica	50 Decatur Cha 02129	617 241-0152
Lucilla	174 Harvard Cam 02139	617 491-5621
M	95 Rowe Ros 02131	617 323-9713
Melvin	501 Green Cam 02139	617 576-1061
Carte Nicholas	18 Appleton Boston 02116	617 695-6996
Cartegena O	4 Milford Bos 02118	617 338-8219
Carten Thos J Sr & Claire	1 Paradise Rd Mil 02186	617 698-6163
Thomas & Kathleen	50 Thompson Ln Mil 02186	617 696-6919
Carter A	Ros 02131	617 327-2257
A	Roxbury	617 442-5230
A	31 Bethune Wy Roxbury 02119	617 442-1219
A	260 Putnam Av Cambridge 02139	617 492-4174
A M	255 Maschsts Av Bos 02115	617 266-7153

Carter F	24 Hillock Ros 02131	617 327-1105
Faye & Ricky	357 Columbus Av Bos 02116	617 437-7331
Francis S	134 Temple W Rox 02132	617 323-6781
Franklin & Anne	221 Mt Auburn Cam 02138	617 354-0798
Fred	42 Haverford Jam 02130	617 524-3078
Fred	96 Hinckley Rd Mil 02186	617 698-1343
G & R	8 Verdun Dor 02124	617 436-8906
G T	27 Franklin Av Som 02145	617 623-7121
Gayle	25 Frontenac Dor 02124	617 825-0322
Geo S	115 Moss Hill Rd Jam 02130	617 522-3215
George	125 Nashua Bos 02114	617 367-9548
Carter Halliday Associate	107 S Street Bos 02111	617 456-1689
Carter Harry F	26 Runng Brk Rd W Rox 02132	617 325-5465
Carter Hide Co Inc	146 Summer Bos 02110	617 542-7987
Carter Hilary	61 Harvey Cam 02140	617 876-2750
Horace	241 Walnut Av Roxbury 02119	617 442-5307
Howard Jr	26 Notre Dme Rox 02119	617 445-5552

Our Function is =
for a given input, find the output

Our Function is =

for a given input (name) $\longrightarrow$

find the output (corresponding phone number)

A Real-World Hashing Function

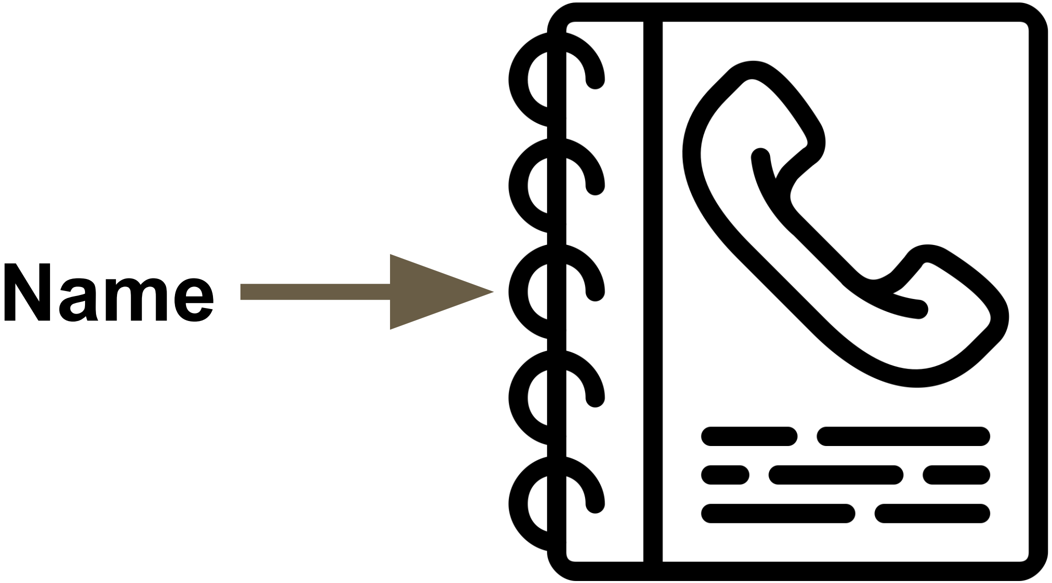
SuperPages.com 195

Cartage New England Inc
26 Allen Ln Ipswich 01938....................978 356-9960
Cartagema Lydia
18 Jewett Ros 02131.................... 617 323-7639
Cartagena Avith
9 Bancroft Rox 02119....................617 442-9780
B Hyd 02136.................................. 617 361-5253
Jessica 50 Decatur Cha 02129..........617 241-0152
Lucilla 174 Harvard Cam 02139......... 617 491-5621
M 95 Rowe Ros 02131.......................617 323-9713
Melvin 501 Green Cam 02139...........617 576-1061
Carte Nicholas
18 Appleton Boston 02116..............617 695-6996
Cartegena O 4 Milford Bos 02118........617 338-8219
Carten Thos J Sr & Claire
1 Paradise Rd Mil 02186................ 617 698-6163
Thomas & Kathleen
50 Thompson Ln Mil 02186.............617 696-6919
Carter A Ros 02131.......................... 617 327-2257
A Roxbury.. 617 442-5230
A 31 Bethune Wy Roxbury 02119......... 617 442-1219
A 260 Putnam Av Cambridge 02139...... 617 492-4174
A M 255 Maschsts Av Bos 02115......... 617 266-7153

Carter F 24 Hillock Ros 02131..............617 327-1105
Faye & Ricky
357 Columbus Av Bos 02116...........617 437-7331
Francis S 134 Temple W Rox 02132.. 617 323-6781
Franklin & Anne
221 Mt Auburn Cam 02138..............617 354-0798
Fred 42 Haverford Jam 02130...........617 524-3078
Fred 96 Hinckley Rd Mil 02186...........617 698-1343
G & R 8 Verdun Dor 02124.................617 436-8906
G T 27 Franklin Av Som 02145............617 623-7121
Gayle 25 Frontenac Dor 02124........... 617 825-0322
Geo S 115 Moss Hill Rd Jam 02130.....617 522-3215
George 125 Nashua Bos 02114.........617 367-9548
Carter Halliday Associate
107 S Street Bos 02111.......................617 456-1689
Carter Harry F
26 Runng Brk Rd W Rox 02132......... 617 325-5465
Carter Hide Co Inc
146 Summer Bos 02110.......................617 542-7987
Carter Hilary 61 Harvey Cam 02140..... 617 876-2750
Horace
241 Walnut Av Roxbury 02119..........617 442-5307
Howard Jr 26 Notre Dme Rox 02119. 617 445-5552

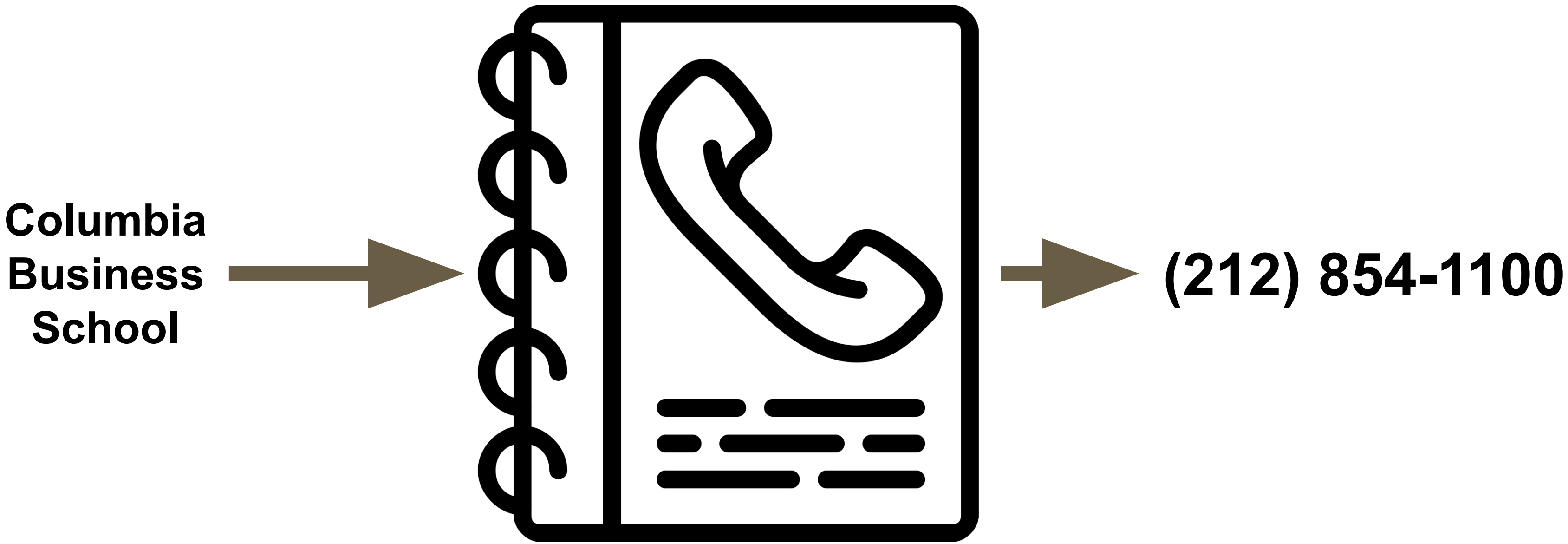
A Real-World Hashing Function

A Real-World Hashing Function



A Real-World Hashing Function

A Real-World Hashing Function

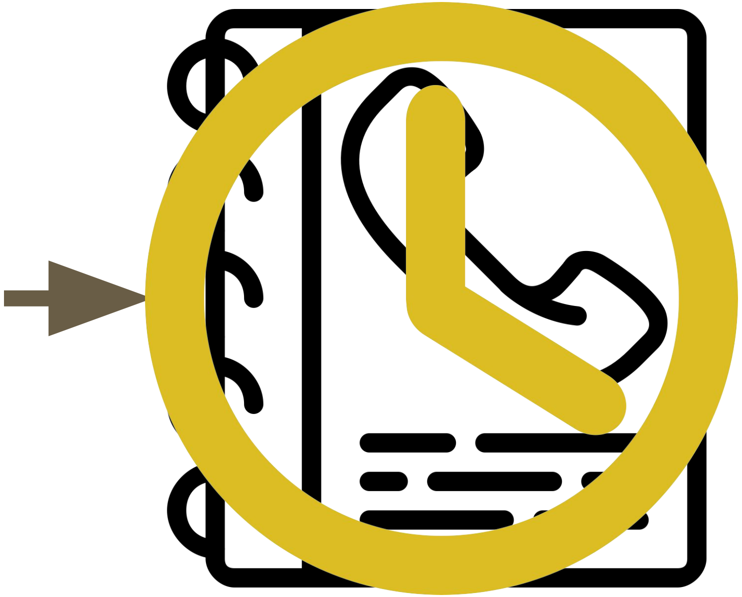


A Real-World Hashing Function

(212) 854-5553

A Real-World Hashing Function

(212) 854-5553




SPACEY
CROWE
PEARCE
BASINGER
DeVITO
Everything Is Suspect...
Everyone Is For Sale...
And Nothing Is What It Seems.
L.A. Confidential


Ginger, you grab a reverse directory
and shag a name and address for me?


Crestview-2239.

So that you know what one-way functions are, let's continue to learn more about hashing and hashing tables ...

Imagine we have a database of over 50 million phone numbers of our customers in the United States. My database does not allow sorting, so how do I find the name of a business associated with a phone number in our database?

Business Name	Phone Number
Stone Rock Capital LLC	212-854-3487
Simple Basic Partners LLP	213-718-1696
Blue Pebble Capital LLC	212-376-3900
Navy Rock Ventures LLC	323-839-1748
Sky Limit Venture Partners LLP	650-337-6291

Business Name	Phone Number
Stone Rock Capital LLC	212-854-3487
Simple Basic Partners LLP	213-718-1696
Blue Pebble Capital LLC	212-376-3900
Navy Rock Ventures LLC	323-839-1748
Sky Limit Venture Partners LLP	650-337-6291

Let’s develop a method (Protocol or Algorithm) to simplify these phone numbers and be able to create sub-categories for storing in our database …

212-854-3487 (take a number from our directory)

21 28 54 34 87 (separate into two-digit numbers)

2+1 2+8 5+4 3+4 8+7 (add up the digits of each tw-digit pair until you get a single-digit number)

3 10 9 7 15
3 1+0 9 7 1+5
3 1 9 7 6 (Done! Then combine to form a 5-digit category number for storing)
31976

Business Name	Phone Number	Category
Stone Rock Capital LLC	212-854-3487	31976
Simple Basic Partners LLP	213-718-1696	
Blue Pebble Capital LLC	212-376-3900	
Navy Rock Ventures LLC	323-839-1748	
Sky Limit Venture Partners LLP	650-337-6291	

213-718-1696 (take a number from our directory)

21 37 18 16 96 (separate into two-digit numbers)

2+1 3+7 1+8 1+6 9+6 (add up the digits of each tw-digit pair until you get a single-digit number)

3 10 9 7 15
3 1+0 9 7 1+5
3 1 9 7 6 (Done! Then combine to form a 5-digit category number for storing)
31976

Business Name	Phone Number	Category
Stone Rock Capital LLC	212-854-3487	31976
Simple Basic Partners LLP	213-718-1696	31976 (is this a problem?!)
Blue Pebble Capital LLC	212-376-3900	
Navy Rock Ventures LLC	323-839-1748	
Sky Limit Venture Partners LLP	650-337-6291	

78742938817753999196055303459477291037892373684068

78 74 29 38 81 77 53 99 91 96 05 53 03 45 94 77 29 10 37 89 23 73 68 40 68

7+8 7+4 2+9 3+8 ...

15 13 11 11 ...

1+5 1+3 1+1 1+1 ...

6 4 2 2 ...

Let's continue to learn more about hashing and hashing tables …

Name	ID Codes
Dara	330i
Cara	X7
Bea	X3
Alice	M4
Ella	128i

Name ▼	ID Codes
Alice	M4
Bea	X3
Cara	X7
Dara	330i
Ella	128i

Name	**ID Codes** ▼
Ella	128i
Dara	330i
Alice	M4
Bea	X3
Cara	X7

Again, imagine no sorting is allowed ...
or the table has tens of thousands of rows and hundred of columns (big data)

Name	Codes
Stadtverordnetenversammlung	2840
KraftfahrzeugHaftpflichtversicherung	9508
Siebentausendzweihundertvierundfünfzig	7254
Rechtsschutzversicherungsgesellschaften	3126
Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz	5434
Donaudampfschifffahrtselektrizitätenhauptbetriebswerkbauunterbeamtengesellschaft	8923

Item Number	Tariff Code
78742938817753999196055303459477291037892373684068	z9m0
76539710192327255231902237652982747470592661143566	0h23
88984727710651739231245830019043173775547558023984	3f26
77603278128172851537873810966507560948211829756526	787y
46527684654614009996682441601858375203324083908888	8nc6

Item Number	Tariff Code
78742938817753999196055303459477291037892373684068	z9m0
76539710192327255231902237652982747470592661143566	0h23
88984727710651739231245830019043173775547558023984	3f26
77603278128172851537873810966507560948211829756526	787y
46527684654614009996682441601858375203324083908888	8nc6

78742938817753999196055303459477291037892373684068

78 74 29 38 81 77 53 99 91 96 05 53 03 45 94 77 29 10 37 89 23 73 68 40 68

7+8 7+4 2+9 3+8 ...

15 13 11 11 ...

1+5 1+3 1+1 1+1 ...

6 4 2 2 ...

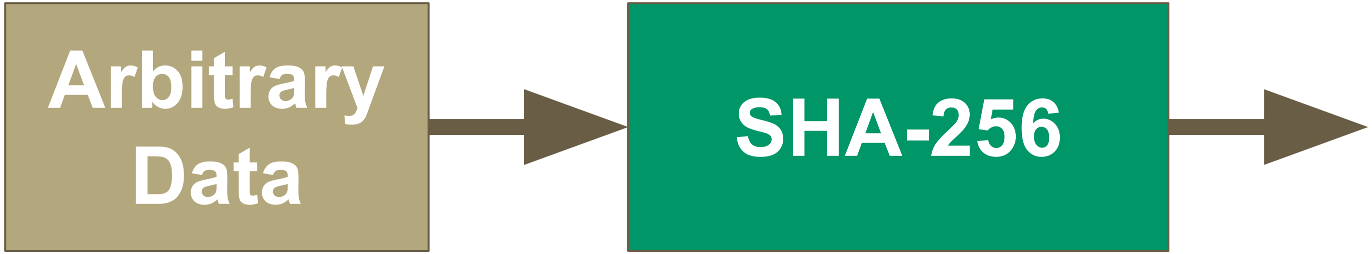
Hash	Item Number	Tariff Code
1458	78742938817753999196055303459477291037892373684068	z9m0
5624	76539710192327255231902237652982747470592661143566	0h23
4548	88984727710651739231245830019043173775547558023984	3f26
4465	77603278128172851537873810966507560948211829756526	787y
2677	46527684654614009996682441601858375203324083908888	8nc6

We need to REALLY minimize the chance of two items having the same hash …

SHA to the rescue!

Bitcoin's Hashing Function

Bitcoin's Hashing Function



Bitcoin's Hashing Function

Bitcoin's Hashing Function

SHA-256

SHA-256 hash: a number with the range:

$$0 \rightarrow 2^{256}$$

SHA-256

2^256 equals to:

115792089237316195423570985008687907853269984665640564039457584007913129639936

SHA-256

SHA-256 hash: a number with the range:

$$0 \rightarrow 2^{256}$$

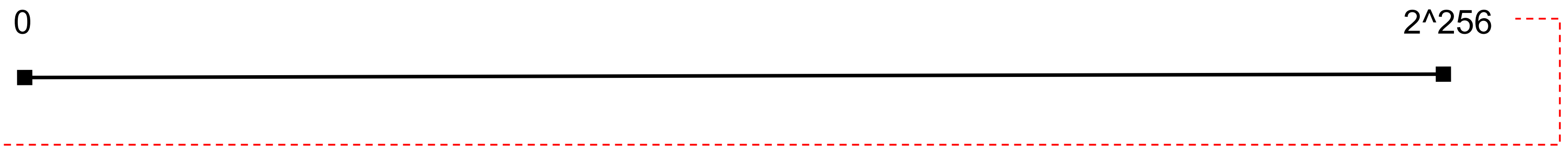
2^256 = 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936

SHA-256: Using an unimaginably large number!

Note that 2^{256} is approximately **10^{77}**

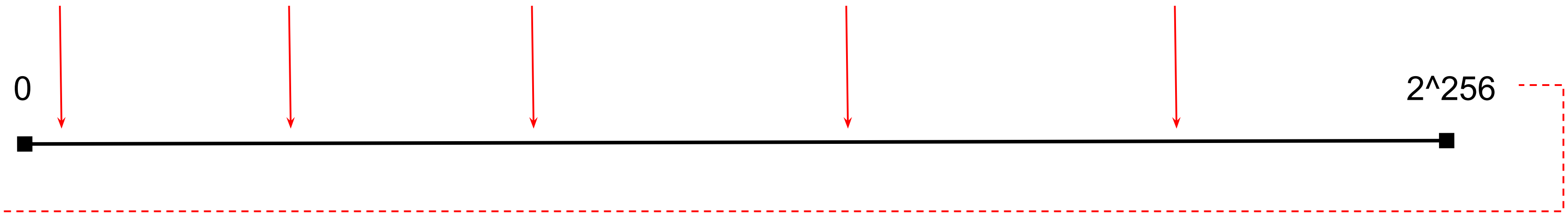
The sum of all the atoms in the universe are estimated to be **10^{80}** (or between 10^{78} and 10^{82})

SHA-256 Hash: a continuous number line



2^256 = 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936

SHA-256 Hash: a continuous number line



2^256 = 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936

SHA-256: points on the long line

Each point would be consisting of many digits:

0
1
2
3
4
8
25
9387
23430174432
57098500868790785
731619542357098500868790785326998466564 0
48532699846659078598957488137489713847985466645240492
115792089237316195423570985008687907853269984665640564039457584007913129639

Numerical Encoding

	Example	Digits Used
Binary Number	11011000	01
Decimal Number	2128541100	0123456789
Hexadecimal Number	7edef5ac	0123456789abcdef

Odometer (mileage count)

Odometer (mileage count)

Odometer (mileage count)

Numerical Encoding

	Example	Digits Used
Binary Number	11011000	01
Decimal Number	2128541100	0123456789
Hexadecimal Number	7edef5ac	0123456789abcdef

SHA-256 Hash

0 2^256

Decimal	**HexaDecimal**
0	0000000000000000000000000000000000000000000000000000000000000000

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001

SHA-256 Hash

0 — 2^{256}

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000a

SHA-256 Hash

0 ——————————————————————— 2^{256}

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000 **a**
11	000000000000000000000000000000000000000000000000000000000000000 **b**
15	000000000000000000000000000000000000000000000000000000000000000 **f**

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000a
11	000000000000000000000000000000000000000000000000000000000000000b
15	000000000000000000000000000000000000000000000000000000000000000f
16	00000000000000000000000000000000000000000000000000000000000000 **10**

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000a
11	000000000000000000000000000000000000000000000000000000000000000b
15	000000000000000000000000000000000000000000000000000000000000000f
16	0000000000000000000000000000000000000000000000000000000000000010
17	0000000000000000000000000000000000000000000000000000000000000011

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000a
11	000000000000000000000000000000000000000000000000000000000000000b
15	000000000000000000000000000000000000000000000000000000000000000f
16	0000000000000000000000000000000000000000000000000000000000000010
17	0000000000000000000000000000000000000000000000000000000000000011
1,000	00000000000000000000000000000000000000000000000000000000000003e8
1,000,000	00000000000000000000000000000000000000000000000000000000000 f4240

Numerical Encoding

	Example	Digits Used
Binary Number	11011000	01
Decimal Number	2128541100	0123456789
Hexadecimal Number	7edef5ac	0123456789abcdef


FROM THE DIRECTOR OF GLADIATOR AND PROMETHEUS
HELP IS ONLY
140 MILLION
MILES AWAY
MATT DAMON
THE MARTIAN
IN CINEMAS SEPTEMBER 30
IN 3D
#TheMartian

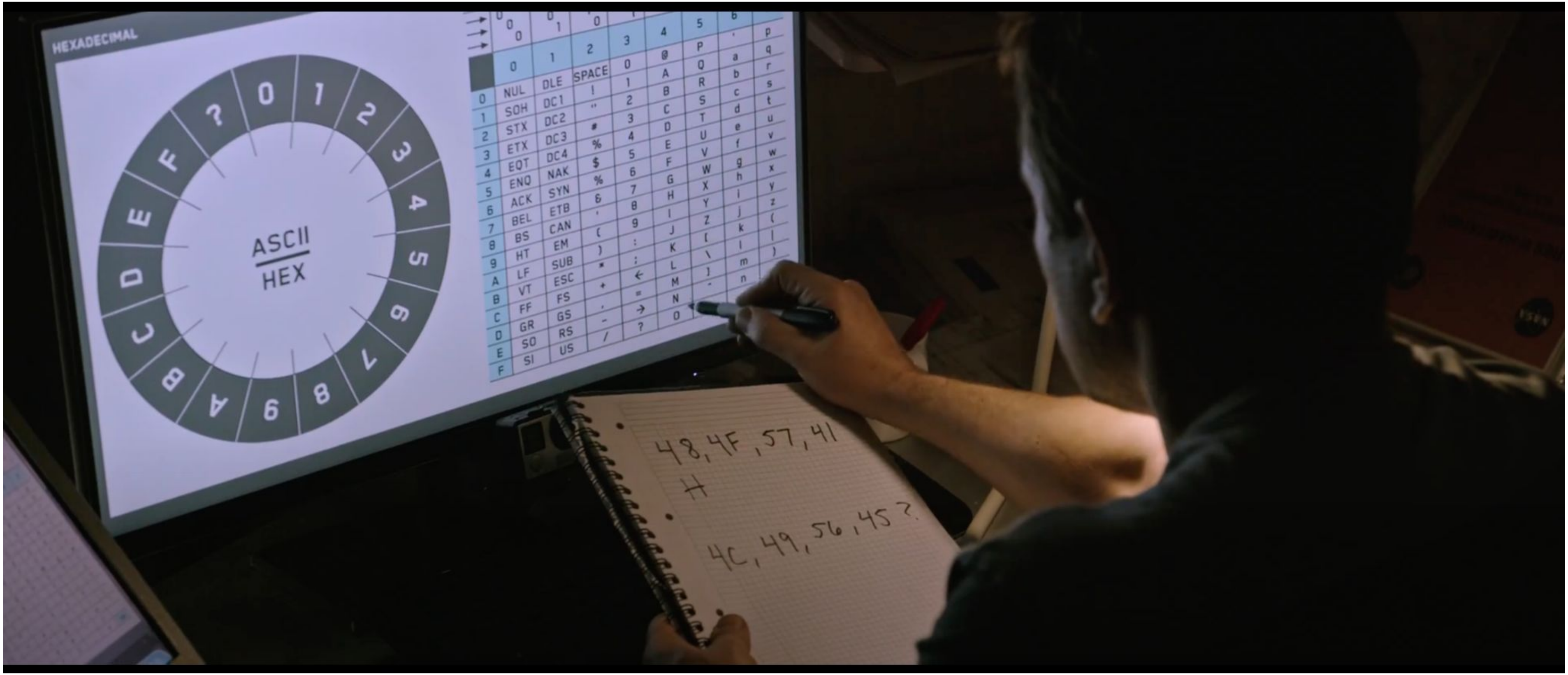

4K ULTRA HD
ASCII
HEX

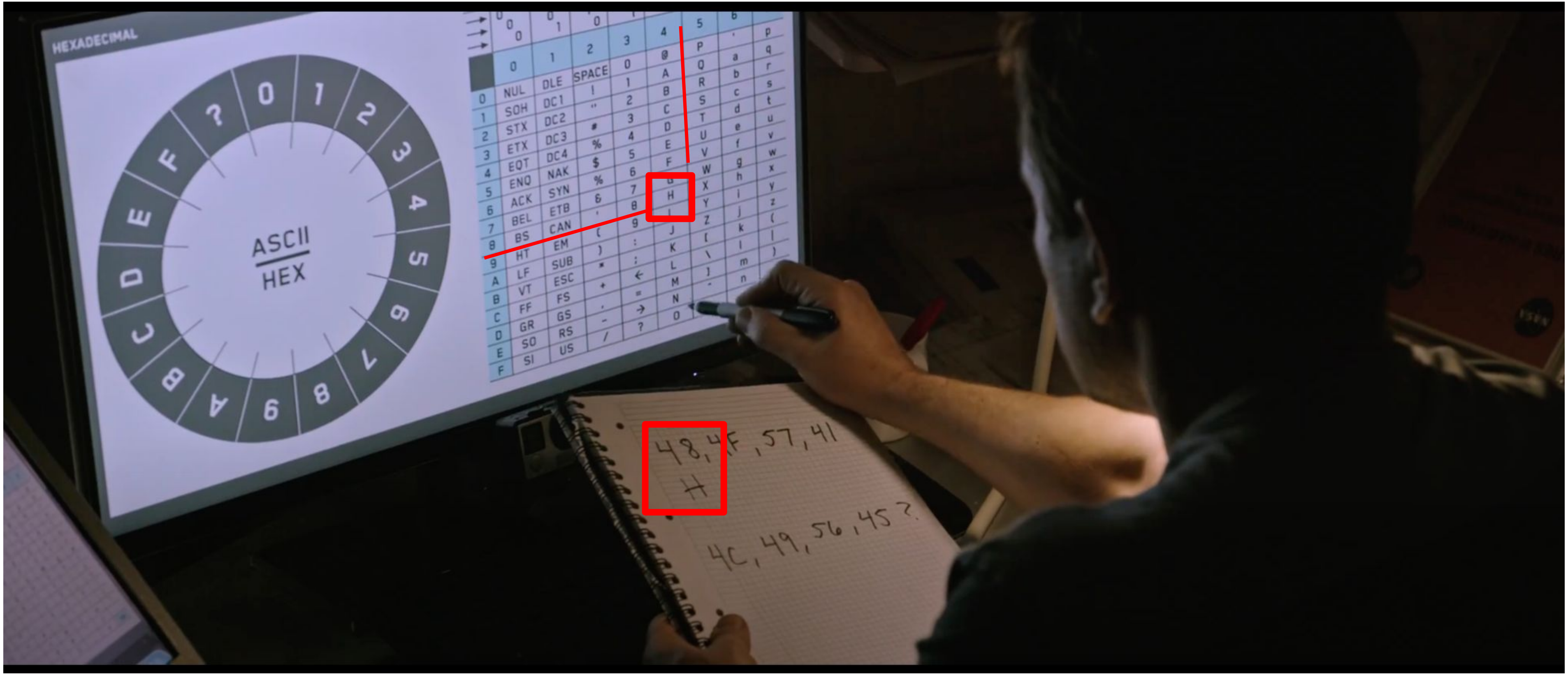
Decimal - Binary - Octal - Hex – ASCII
Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Decimal - Binary - Octal - Hex – ASCII
Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	“	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	‘	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL


HEXADECIMAL
ASCII
HEX
0 1 2 3 4 5 6 7 8 9 A B C D E F ?
48, 4F, 57, 41
H
4C, 49, 56, 45 ?


HEXADECIMAL
ASCII
HEX
48, 4F, 57, 41
H
4C, 49, 56, 45 ?

A few thoughts on “Collisions”…

Remember these?

Brute force the unlocking of this briefcase …

Brute force the unlocking of this briefcase ...
... it will take you 3 seconds per each try.

How long will it take to "hack" the briefcase open without knowing the secret lock code?

Combination Lock (3 rotary dials)

1	4	8
Digits	**Digits**	**Digits**
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

$10 \times 10 \times 10 = 1{,}000 = 10^3$

Combination Lock (3 rotary dials)

$10 \times 10 \times 10 = 1{,}000 = 10^3$ total possible combinations

With 3 seconds per each combination, we will need:
3×10^3 second (or 3,000 seconds)

There are 60 seconds in each minutes, so:

$(3 \times 10^3) \div 60 = 50$ minutes max to open each lock
OR
$3{,}000 \div 60 = 50$ minutes max to open each lock

Combination Lock (3 rotary dials)

10 x 10 x 10 = 1,000 = 10^3 total possible combinations

3 seconds per each combination
1 min = 60 seconds

So 60 ÷ 3 = 20 combinations per minute
1,000 ÷ 20 = 50 minutes to open each lock

Combination Lock (3 rotary dials)

1	4	8
Digits	**Digits**	**Digits**
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

$$10 \times 10 \times 10 = 1{,}000 = 10^3$$

3	d	e	7	6	b	e	1	8	c	...	5
Digits	**Digits**	**Digits**	**Digits**	**Digits**	**Digits**	**Digits**	**Digits**	**Digits**	**Digits**		**Digits**
0	0	0	0	0	0	0	0	0	0		0
1	1	1	1	1	1	1	1	1	1		1
2	2	2	2	2	2	2	2	2	2		2
3	3	3	3	3	3	3	3	3	3		3
4	4	4	4	4	4	4	4	4	4		4
5	5	5	5	5	5	5	5	5	5		5
6	6	6	6	6	6	6	6	6	6		6
7	7	7	7	7	7	7	7	7	7		7
8	8	8	8	8	8	8	8	8	8		8
9	9	9	9	9	9	9	9	9	9		9
a	a	a	a	a	a	a	a	a	a		a
b	b	b	b	b	b	b	b	b	b		b
c	c	c	c	c	c	c	c	c	c		c
d	d	d	d	d	d	d	d	d	d		d
e	e	e	e	e	e	e	e	e	e		e
f	f	f	f	f	f	f	f	f	f		f
16	16	16	16	16	16	16	16	16	16	...	16

SHA-256 Hash: Why 64 characters?

Text	**SHA-256 Hash (HexaDecimal)**
hello	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
	16x16x16x16x16x16x16x16x16x... [16 multiplied by itself 64 times]

SHA-256 Hash: Why 64 characters?

0 ——————————————————— 2^256

Text	SHA-256 Hash (HexaDecimal)
hello	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

16x16x16x16x16x16x16x16x16x... [16 multiplied by itself 64 times]

$$16^{64} = (2^4)^{64} = 2^{4x64} = 2^{256}$$

SHA-256

SHA-256 hash: a number with the range:

$$0 \rightarrow 2^{256}$$

2^256 = 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936

SHA-256: Using an unimaginably large number!

Note that 2^{256} is approximately **10^{77}**

The sum of all the atoms in the universe are estimated to be **10^{80}** (or between 10^{78} and 10^{82})

Let’s brute-force a SHA-256 collision by using state-of-the-art machine:

Some of the Fastest Machines

Bitmain
Antminer S21 Hyd (335Th)

Description

Model **Antminer S21 Hyd (335Th)** from **Bitmain** mining **SHA-256 algorithm** with a maximum hashrate of **335Th/s** for a power consumption of **5360W.**

Bitmain
Antminer S19 XP Hyd (255Th)

Description

Model **Antminer S19 XP Hyd (255Th)** from **Bitmain** mining **SHA-256 algorithm** with a maximum hashrate of **255Th/s** for a power consumption of **5304W.**

Some of the Fastest Machines

Bitmain
Antminer S21 Hyd (335Th)

Description

Model **Antminer S21 Hyd (335Th)** from **Bitmain** mining **SHA-256 algorithm** with a maximum hashrate of **335Th/s** for a power consumption of **5360W.**

Bitmain
Antminer S19 XP Hyd (255Th)

Description

Model **Antminer S19 XP Hyd (255Th)** from **Bitmain** mining **SHA-256 algorithm** with a maximum hashrate of **255Th/s** for a power consumption of **5304W.**

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

$(10^{77}) \div (255 \times 10^{12}) = 3.92 \times 10^{62}$ seconds

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

$(10^{77}) \div (255 \times 10^{12}) = 3.92 \times 10^{62}$ seconds

How many years will that be?

Well, there are appx (365 days x 24 hrs x 60 mins x 60 secs) seconds per year, so there are appx 31,536,000 seconds per year, OR 3.15×10^{7} secs/year

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

$(10^{77}) \div (255 \times 10^{12}) = 3.92 \times 10^{62}$ seconds

How many years will that be?
With 31,536,000 seconds per year, OR 3.15×10^7 secs/year →

$(3.92 \times 10^{62}$ seconds$) \div (3.15 \times 10^7$ secs/year$) = 1.24 \times 10^{55}$ years

Let's get 1 billion (10^9) of these machines, so:
$(1.24 \times 10^{55}$ years$) \div 10^9 =$ **1.24×10^{46} years**

Let's find a SHA-256 collision

So, how long will it take with <u>one billion</u> machines at 255 Th/s to run through all numbers between 0 and 10^{77}?

$(1.24 \times 10^{55}$ years$) \div 10^9 =$ 1.24×10^{46} years

Age of the Universe: 13.8×10^9 years
Age of Earth: $4.5 \quad \times 10^9$ years

We would need these many universe lifetimes to make it:
$(1.24 \times 10^{46}$ yrs$) \div (13.8 \times 10^9$ yrs$) = 8.99 \times 10^{35}$
898,550,724,637,681,159,420,289,855,072,463,768

899,000,000,000,000,000,000,000,000,000,000,000 😱🤯

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

$(10^{77}) \div (255 \times 10^{12}) = 3.92 \times 10^{62}$ seconds

$(3.92 \times 10^{62}$ seconds$) \div (3.15 \times 10^{7}$ secs/year$) = 1.24 \times 10^{55}$ years

Let's get 1 billion (10^9) of these machines, so:
$(1.24 \times 10^{55}$ years$) \div 10^9 =$ **1.24×10^{46} years**

Let's find a SHA-256 collision

1 Terahash = 1 trillion hashes per second = 10^{12} h/s
SHA-256 is appx. 10^{77} total possible numbers (i.e. hashes)

So, **how long will it take with one machine at 255 Th/s to run through all numbers between 0 and 10^{77}**?

$(10^{77}) \div (255 \times 10^{12}) = 3.92 \times 10^{62}$ seconds

$(3.92 \times 10^{62}$ seconds$) \div (3.15 \times 10^{7}$ secs/year$) = 1.24 \times 10^{55}$ years

Let's get 1 billion (10^9) of these machines, so:
$(1.24 \times 10^{55}$ years$) \div 10^9 =$ **1.24×10^{46} years**

At $3,000 a machine, we'd need $3,000,000,000,000 just to buy them
(3 trillion dollars … annual GDP of France!)

SHA-256 Hash

Decimal	HexaDecimal
0	0000000000000000000000000000000000000000000000000000000000000000
1	0000000000000000000000000000000000000000000000000000000000000001
2	0000000000000000000000000000000000000000000000000000000000000002
9	0000000000000000000000000000000000000000000000000000000000000009
10	000000000000000000000000000000000000000000000000000000000000000a
11	000000000000000000000000000000000000000000000000000000000000000b
15	000000000000000000000000000000000000000000000000000000000000000f
16	0000000000000000000000000000000000000000000000000000000000000010
17	0000000000000000000000000000000000000000000000000000000000000011
1000	00000000000000000000000000000000000000000000000000000000000003e8
(2^256)-1	
2^256	

SHA-256 Hash

Decimal	HexaDecimal	
0	0000000000000000000000000000000000000000000000000000000000000000	
1	0000000000000000000000000000000000000000000000000000000000000001	
2	0000000000000000000000000000000000000000000000000000000000000002	
9	0000000000000000000000000000000000000000000000000000000000000009	
10	000000000000000000000000000000000000000000000000000000000000000a	
11	000000000000000000000000000000000000000000000000000000000000000b	
15	000000000000000000000000000000000000000000000000000000000000000f	
16	0000000000000000000000000000000000000000000000000000000000000010	
17	0000000000000000000000000000000000000000000000000000000000000011	
1000	00000000000000000000000000000000000000000000000000000000000003e8	
(2^256)-1	ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff	
2^256	1000000000000000000000000000000000000000000000000000000000000000	**0**

SHA-256 Hex Encoding

Instead of a long hash consisting of many digits:

```
0
1
2
3
4
8
25
9387
23430174432
57098500868790785
73161954235709850086879078532699846656405640
48532699846659078598957488137489713847985466452404920
115792089237316195423570985008687907853269984665640564039457584007913129639
```

SHA-256 Hex Encoding

Instead of a long hash consisting of many digits:

```
0
1
25
23430174432
57098500868790785
7316195423570985008687907853269984665640
48532699846659078598957488137489713847985466645240492
115792089237316195423570985008687907853269984665640564039457584007913129639
```

SHA-256 Hex Encoding

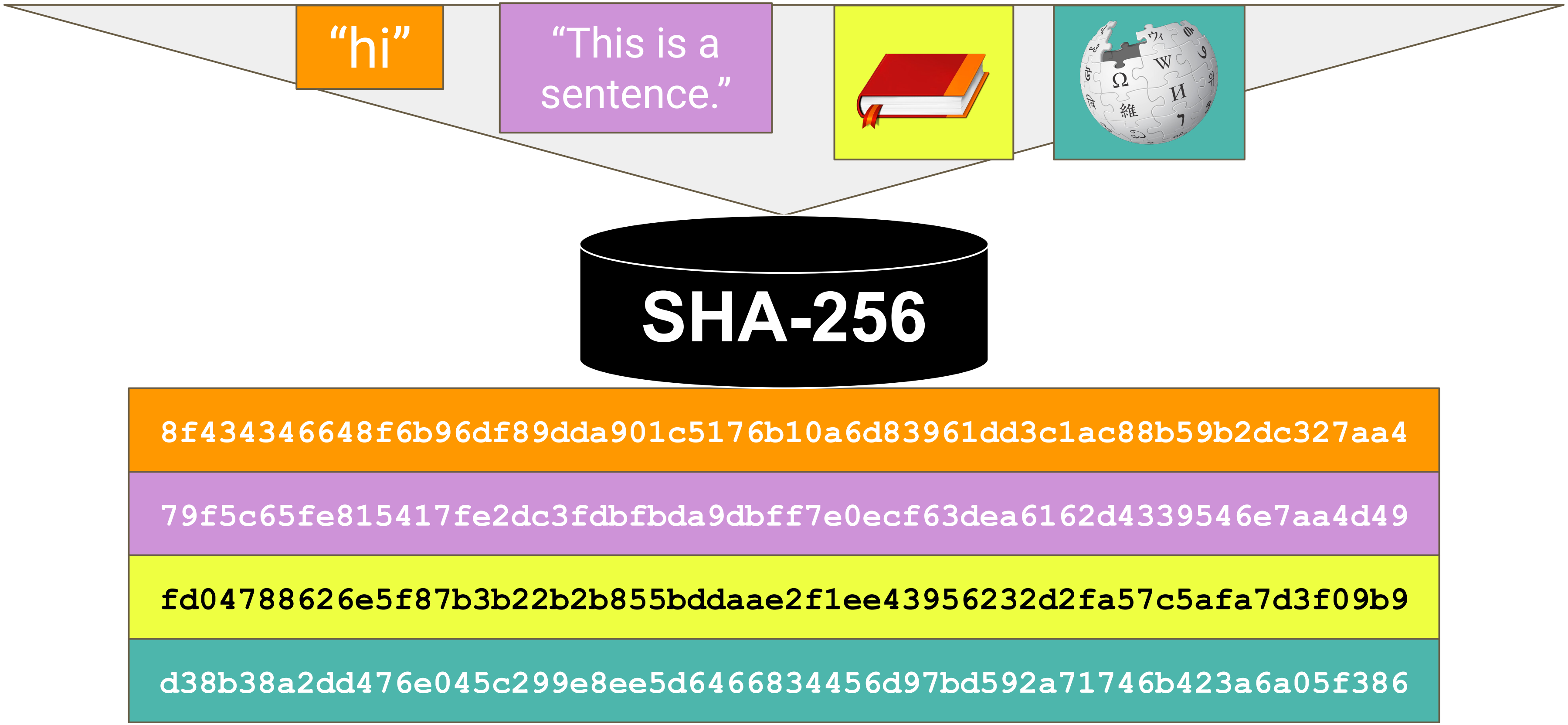
Instead of a long hash consisting of many digits:

```
0
1
25
23430174432
57098500868790785
731619542357098500868790785326998466564O
```

We have (a <u>fixed</u> string of 64 characters … always):

```
fd04788626e5f87b3b22b2b855bddaae2f1ee43956232d2fa57c5afa7d3f09b9
4faa640f3077ded9d2b7fc6f429050defc5d26e08e5b241edadd39a49e56af51
933e1c934309c9d942921fcebcd8fc398553f2c39ccb162cb53bd998149b042b
```

SHA-256



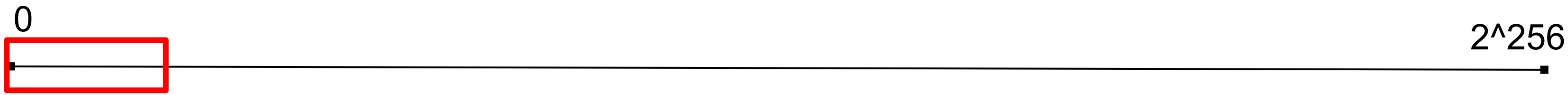
“hi”
“This is a sentence.”
SHA-256
8f434346648f6b96df89dda901c5176b10a6d83961dd3c1ac88b59b2dc327aa4
79f5c65fe815417fe2dc3fdbfbda9dbff7e0ecf63dea6162d4339546e7aa4d49
fd04788626e5f87b3b22b2b855bddaae2f1ee43956232d2fa57c5afa7d3f09b9
d38b38a2dd476e045c299e8ee5d6466834456d97bd592a71746b423a6a05f386

DEMO: Hash (SHA-256)

SHA-256 Hash: Remember why 64 characters?

Text	SHA-256 Hash (HexaDecimal)
hello	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

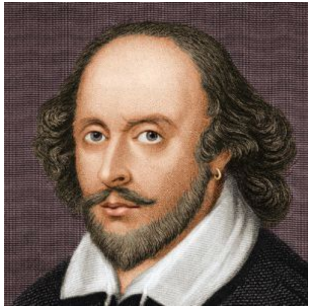
SHA-256 Hash: setting thresholds

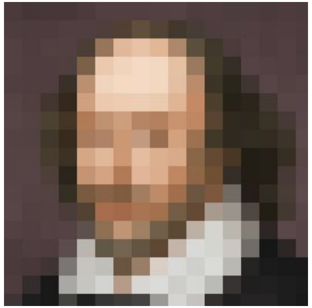


$$\left(\frac{1}{16}\right) \times 2^{256}$$

Numbers with one <u>leading</u> zero

SHA-kespeare



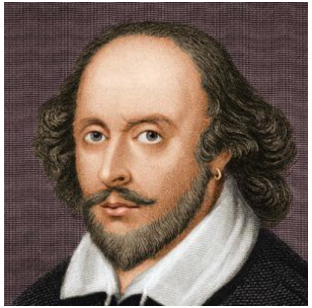


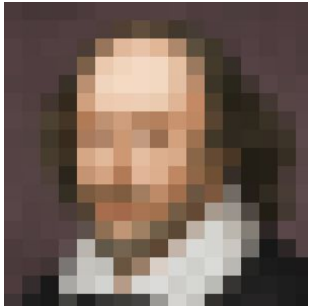
Approx 110,968 lines of Shakespeare written (all works)

Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros			
3 Leading Zeros			
2 Leading Zeros			
1 Leading Zero			
No Leading Zeros			

Calculate the expected number of lines from no leading zero to 4 leading zeros. You have 5 minutes.

SHA-kespeare

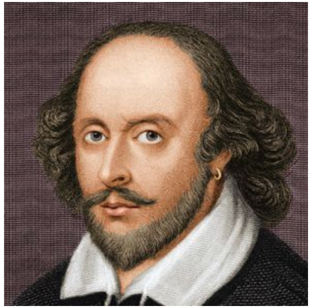


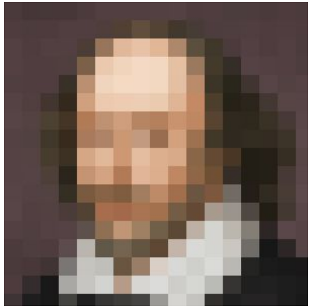


Approx 110,968 lines of Shakespeare written (all works)

Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros			
3 Leading Zeros			
2 Leading Zeros			
1 Leading Zero			
No Leading Zeros			

SHA-kespeare

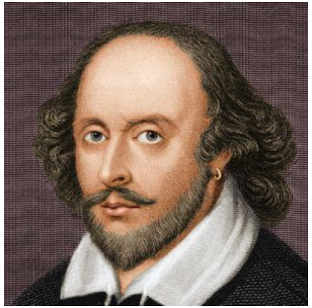


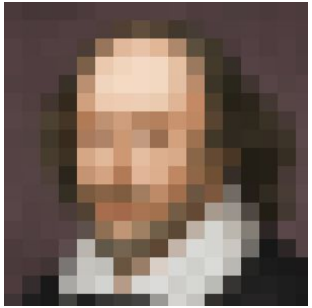


Approx 110,968 lines of Shakespeare written (all works)

Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros			
3 Leading Zeros			
2 Leading Zeros			
1 Leading Zero		~6,935	
No Leading Zeros		~104,033	

SHA-kespeare

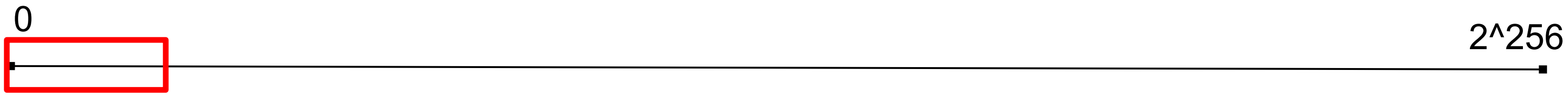




Approx 110,968 lines of Shakespeare written (all works)

Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros			
3 Leading Zeros			
2 Leading Zeros			
1 Leading Zero		~~~6,935~~	
No Leading Zeros		~104,033	

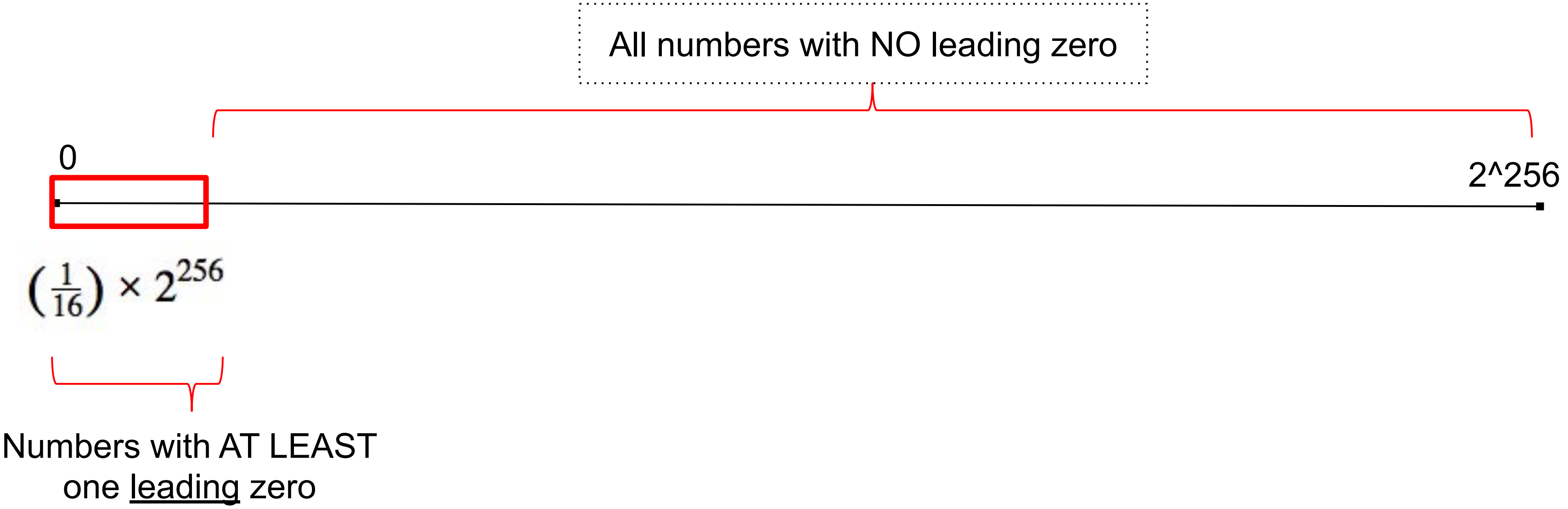
SHA-256 Hash: setting thresholds



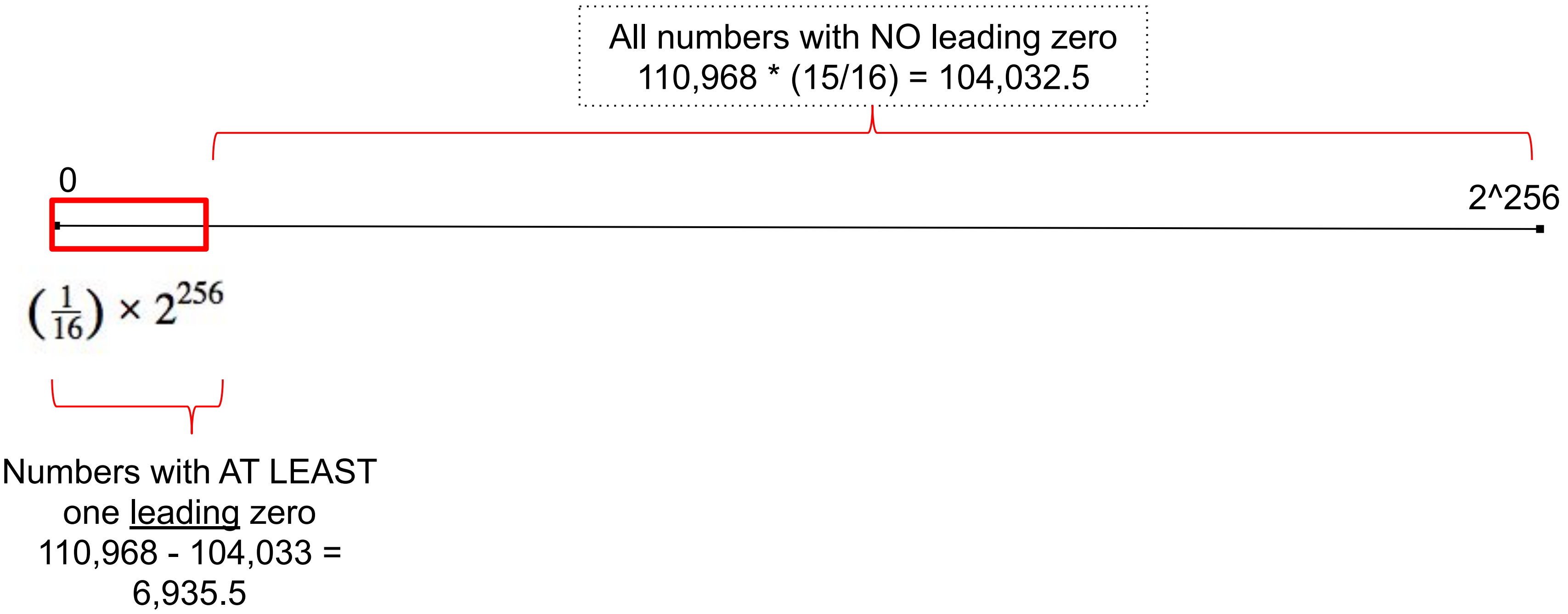
$$\left(\frac{1}{16}\right) \times 2^{256}$$

Numbers with **AT LEAST** one leading zero

SHA-256 Hash: setting thresholds



SHA-256 Hash: setting thresholds

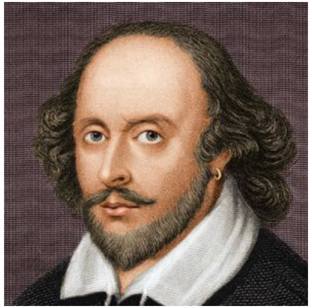


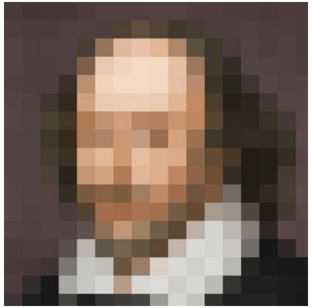
SHA-256 Hash: setting thresholds

	Value	Value(15/16)	Value-Value(15/16)
	A	B	C
1	110,968	**104,033**	6,936
2	6,936	**6,502**	433
3	433	**406**	27
4	27	**25**	**2**

110,968	**=A1*15/16**	=A1-B1
=C1	**=A2*15/16**	=A2-B2
=C2	**=A3*15/16**	=A3-B3
=C3	**=A4*15/16**	=A4-B4

SHA-kespeare

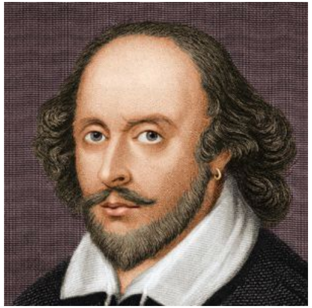


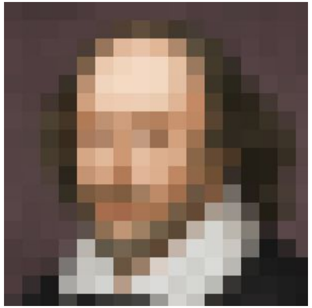


Approx 110,968 lines of Shakespeare written (all works)

Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros		~2	0.0018%
3 Leading Zeros		~25	0.0160%
2 Leading Zeros		~406	0.3740%
1 Leading Zero		~6,502	6.0044%
No Leading Zeros		~104,033	93.603%

SHA-kespeare

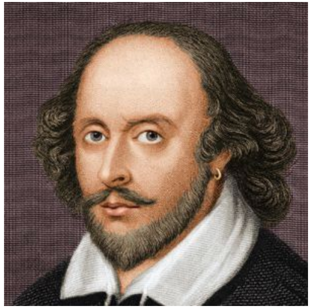




Approx 110,968 lines of Shakespeare written (all works)

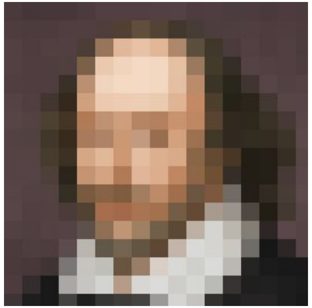
Hash Criteria	Lines of Shakespeare	Expected Number of Lines	Actual % of lines (out of ~110,968)
4 Leading Zeros	2	~2	0.0018%
3 Leading Zeros	18	~25	0.0160%
2 Leading Zeros	415	~406	0.3740%
1 Leading Zero	6,663	~6,502	6.0044%
No Leading Zeros	103,870	~104,033	93.603%

SHA-kespeare



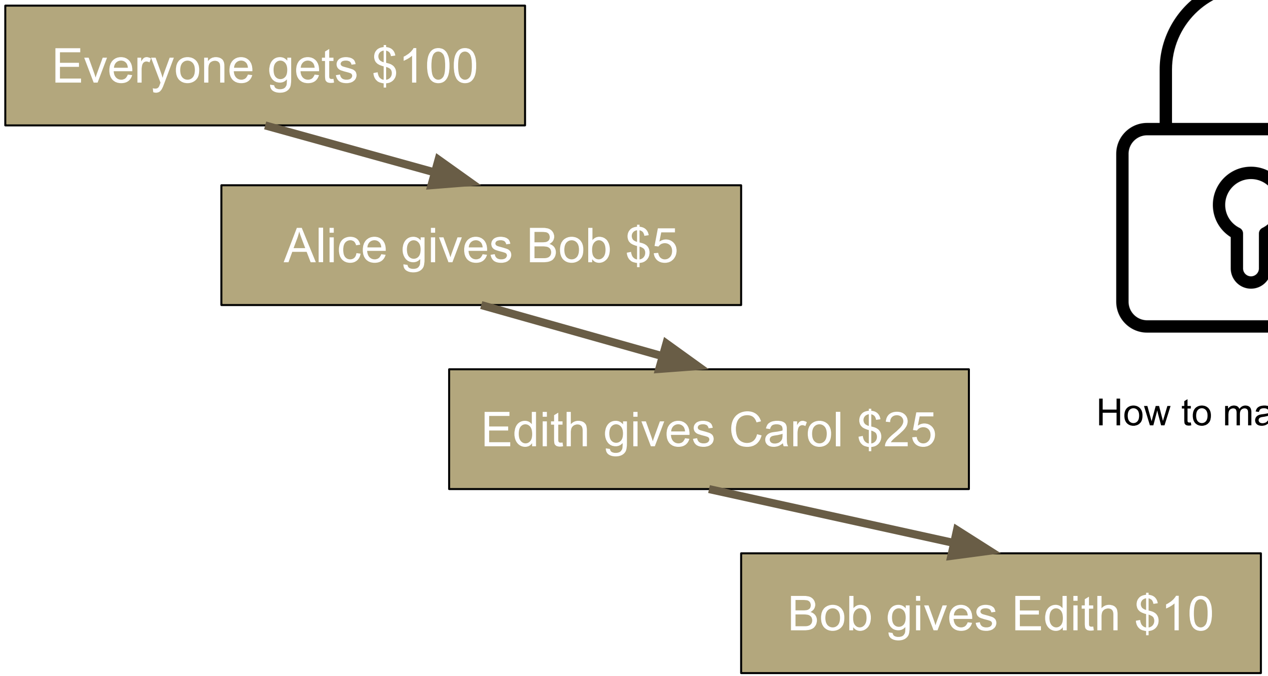
Hamlet, Act I, Scene 2:

King. Have you your father's leave?
What says Polonius?



000055779d9bda7accb203c8256e6106e
2d44d68025b83624af59e31c3527275

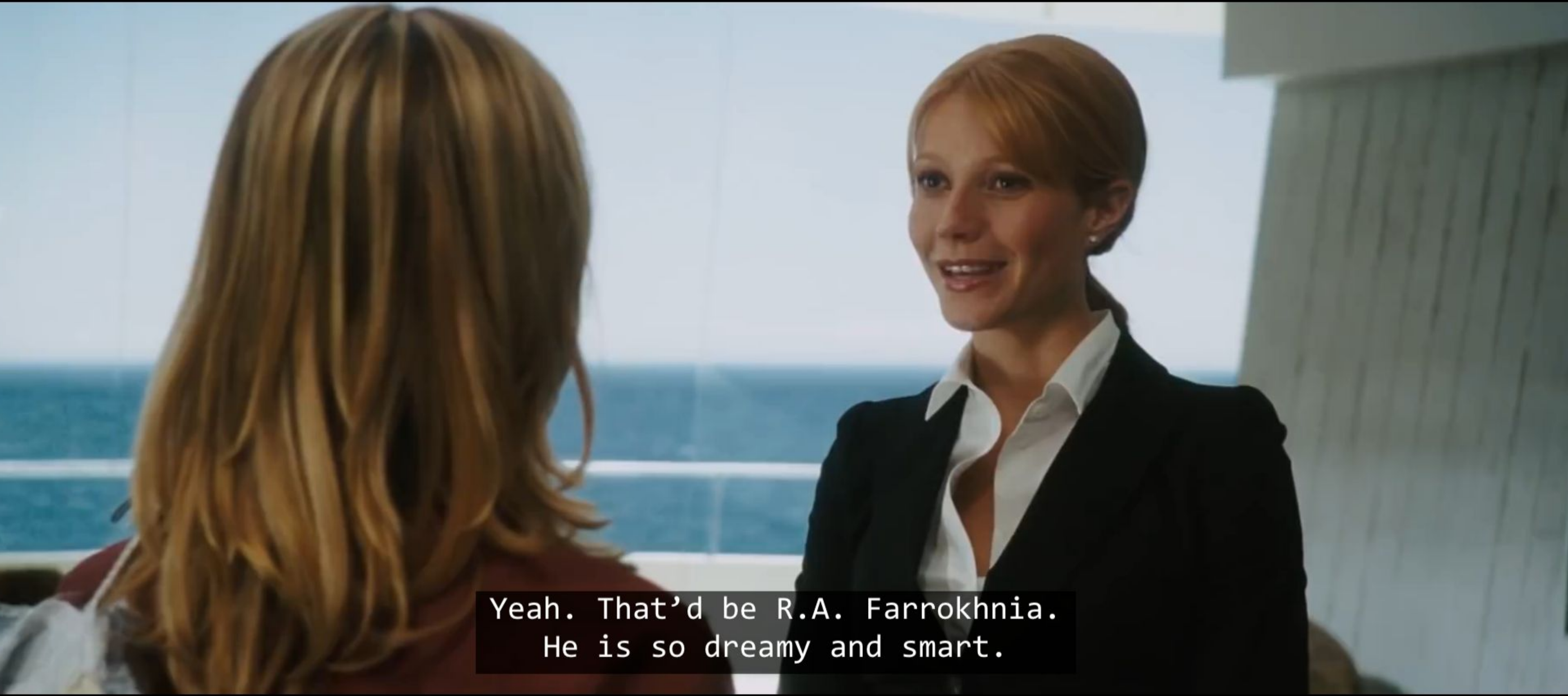
Blockchain: a cryptographically-verifiable Tx chain



How to make the chain secured?

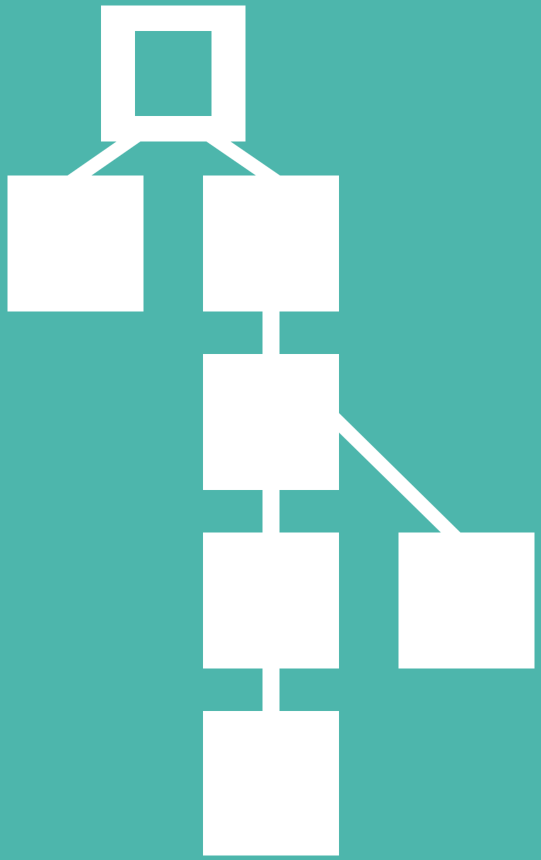

ROBERT
DOWNEY JR.
TERRENCE
HOWARD
JEFF
BRIDGES
GWYNETH
AND PALTROW
STARK INDUSTRIES
IRON MAN
COMING SOON


No. But I hear Tony met his match in this Columbia
guy that’s apparently super smart. Ring a bell?


Yeah. That’d be R.A. Farrokhnia.
He is so dreamy and smart.

IV. Building the Blockchain

Using all we've learned to build an immutable chain of "digital assets" (and more)



Exercise: let's do a (theoretical) deal!

Parties involved (client wants to use their own legal & accounting)

1.
2.
3.
4.

Exercise: let’s do a deal!

Parties involved (client wants to use their own legal & accounting)

1. **Bridget Fonda (BF); Commercial Bank Corp (CBC); IB**

Exercise: let’s do a deal!

Parties involved (client wants to use their own legal & accounting)

1. Bridget Fonda (BF); Commercial Bank Corp (CBC); IB
2. Robert Farrokhnia (RF); Columbia University (COL); Advisor

Exercise: let’s do a deal!

Parties involved (client wants to use their own legal & accounting)

1. Bridget Fonda (BF); Commercial Bank Corp (CBC); IB
2. Robert Farrokhnia (RF); Columbia University (COL); Advisor
3. Jeff Dewey (JD); Dewey, Cheatem & Howe (DCH); Law

Exercise: let's do a deal!

Parties involved (client wants to use their own legal & accounting)

1. Bridget Fonda (BF); Commercial Bank Corp (CBC); IB
2. Robert Farrokhnia (RF); Columbia University (COL); Advisor
3. Jeff Dewey (JD); Dewey, Cheatem & Howe (DCH); Law
4. Alex Runne (AR); Steel, Runne & Hyde (SRH): Accounting

We will have lots of documents going back and forth.

Exercise: let’s do a deal!

Our document naming convention, or <u>protocol</u>:

[type of doc]_[company name]_[author’s initials]_[author’s employer]_[date: mm/dd/yy]_[version number: v#]

[type of doc]_[company name]_[author's initials]_[author's employer]_[date: mm/dd/yy]_[version number: v#]

PPM_Newco_RF_COL_041523_v1

[type of doc]_[company name]_[author's initials]_[author's employer]_[date: mm/dd/yy]_[version number: v#]

PPM_Newco_RF_COL_041523_v1
PPM_Newco_BF_CBC_041623_v2

Exercise: naming protocol sorted by "Name"

Name	Date Modified	Size	Kind
PPM_Newco_AR_SRH_072817_v9	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_BF_CBC_072717_v8	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_RF_COL_072717_v5	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_AR_SRH_072717_v4	Today, 5:44 PM	23 KB	Microsoft Word document
PPM_Newco_JD_DCH_072617_v3	Today, 4:19 PM	22 KB	Microsoft Word document
PPM_Newco_BF_CBC_072617_v2	Today, 4:19 PM	22 KB	Microsoft Word document
PPM_Newco_RF_COL_072517_v1	Today, 3:49 PM	22 KB	Microsoft Word document

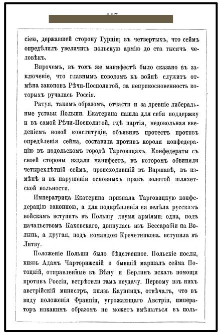
Exercise: naming protocol sorted by “Date Modified”

Name	Date Modified	Size	Kind
PPM_Newco_AR_SRH_072717_v4	Today, 5:44 PM	23 KB	Microsoft Word document
PPM_Newco_AR_SRH_072817_v9	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_BF_CBC_072717_v8	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_RF_COL_072717_v5	Today, 4:20 PM	23 KB	Microsoft Word document
PPM_Newco_JD_DCH_072617_v3	Today, 4:19 PM	22 KB	Microsoft Word document
PPM_Newco_BF_CBC_072617_v2	Today, 4:19 PM	22 KB	Microsoft Word document
PPM_Newco_RF_COL_072517_v1	Today, 3:49 PM	22 KB	Microsoft Word document

What can do wrong? How to fix the system?

Let's build a blockchain, connecting and linking verified digital files in an immutable way with a shared ledger to keep track of it all that every party can see.

PPM v1



[redacted]

сіею, державшей сторону Турціи; въ четвертыхъ, что сеймъ опредѣлилъ увеличить польскую армію до ста тысячъ человѣкъ.

Впрочемъ, въ томъ же манифестѣ было сказано въ заключеніе, что главнымъ поводомъ къ войнѣ служитъ отмѣна законовъ Рѣчи-Посполитой, за неприкосновенность которыхъ ручалась Россія.

Ратуя, такимъ образомъ, отчасти и за древніе либеральные уставы Польши. Екатерина нашла для себя поддержку и въ самой Рѣчи-Посполитой, гдѣ партія, недовольная введеніемъ новой конституціи, объявивъ протестъ противъ опредѣленія сейма, составила противъ короля конфедерацію въ подольскомъ городѣ Тарговицахъ. Конфедераты съ своей стороны издали манифестъ, въ которомъ обвиняли четырехлѣтній сеймъ, происходившій въ Варшавѣ, въ измѣнѣ и въ нарушеніи основныхъ правъ золотой шляхетской вольности.

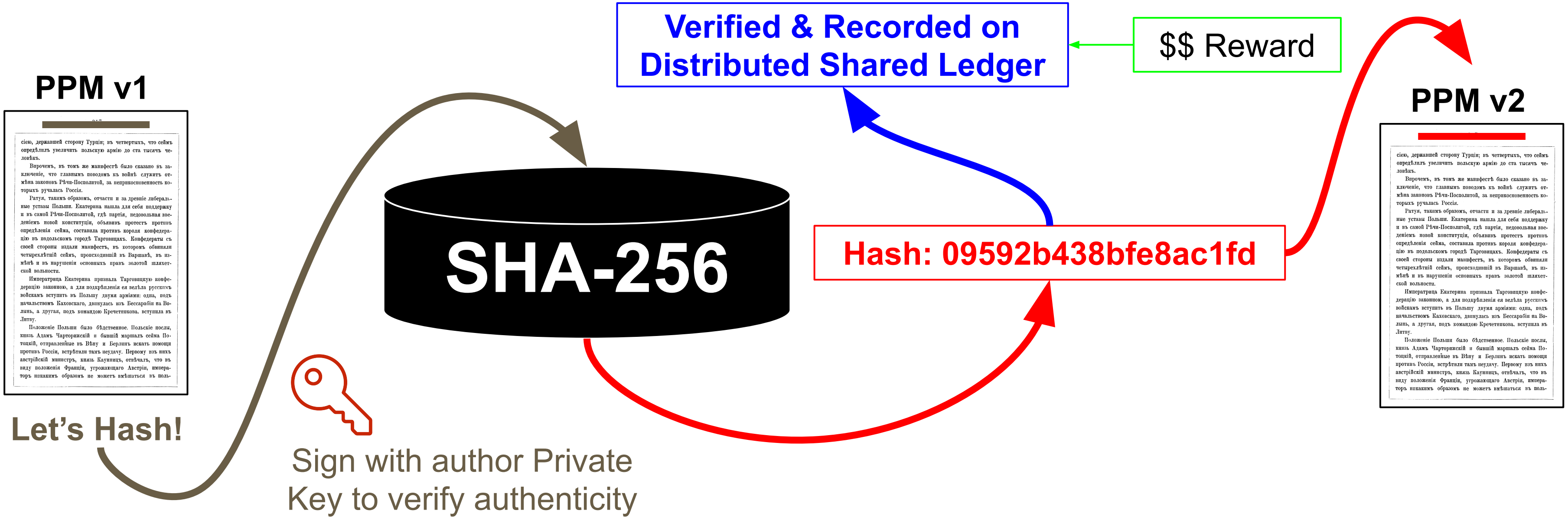
Императрица Екатерина признала Тарговицкую конфедерацію законною, а для подкрѣпленія ея велѣла русскимъ войскамъ вступить въ Польшу двумя арміями: одна, подъ начальствомъ Каховскаго, двинулась изъ Бессарабіи на Волынь, а другая, подъ командою Кречетникова, вступила въ Литву.

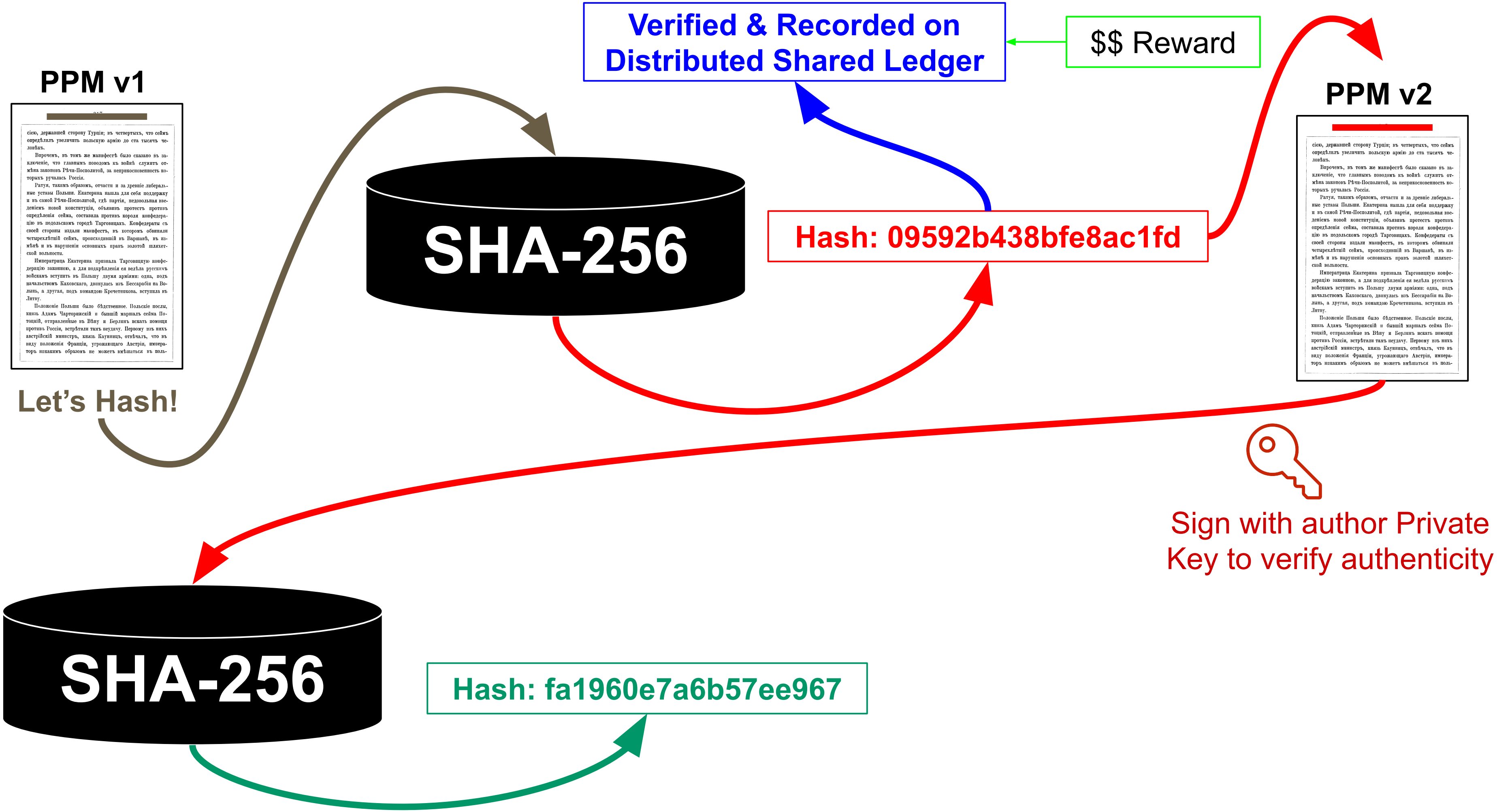
Положеніе Польши было бѣдственное. Польскіе послы, князь Адамъ Чарторижскій и бывшій маршалъ сейма Потоцкій, отправленные въ Вѣну и Берлинъ искать помощи противъ Россіи, встрѣтили тамъ неудачу. Первому изъ нихъ австрійскій министръ, князь Кауницъ, отвѣчалъ, что въ виду положенія Франціи, угрожающаго Австріи, императоръ никакимъ образомъ не можетъ вмѣшаться въ поль-

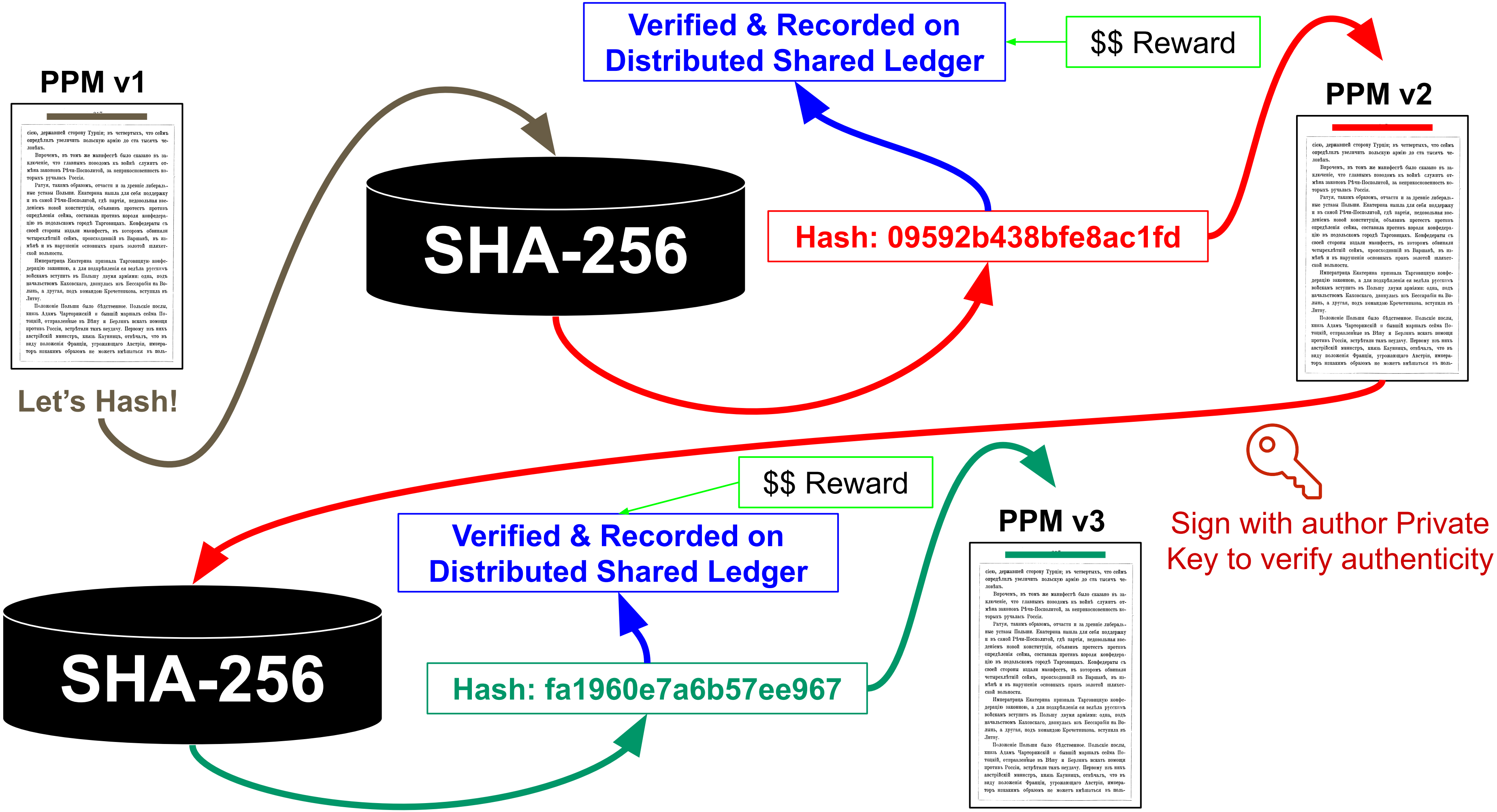
Let's Hash!

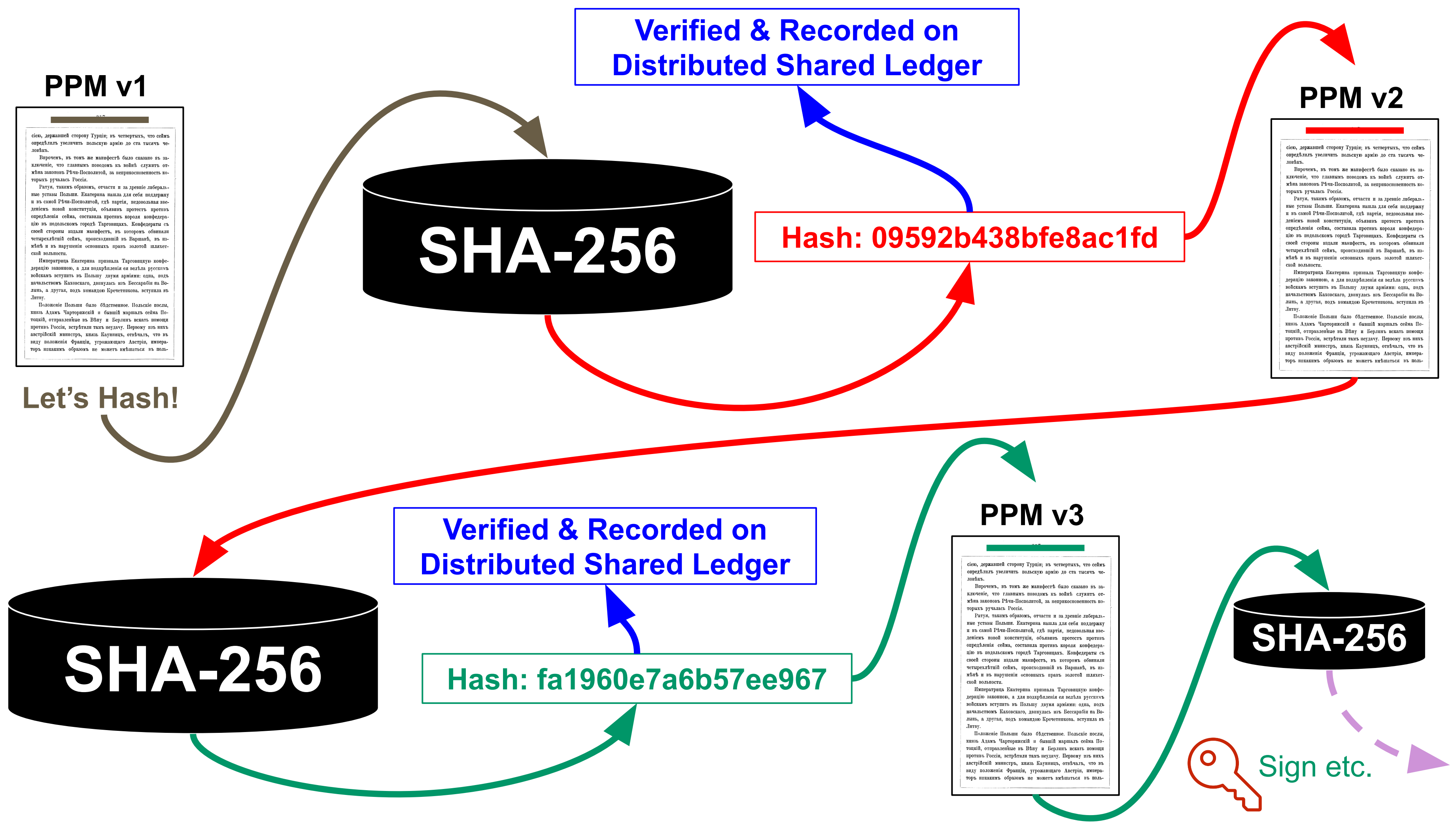
Sign with author Private
Key to verify authenticity


PPM v1
Let's Hash!
SHA-256
Hash: 09592b438bfe8ac1fd
Sign with author Private
Key to verify authenticity

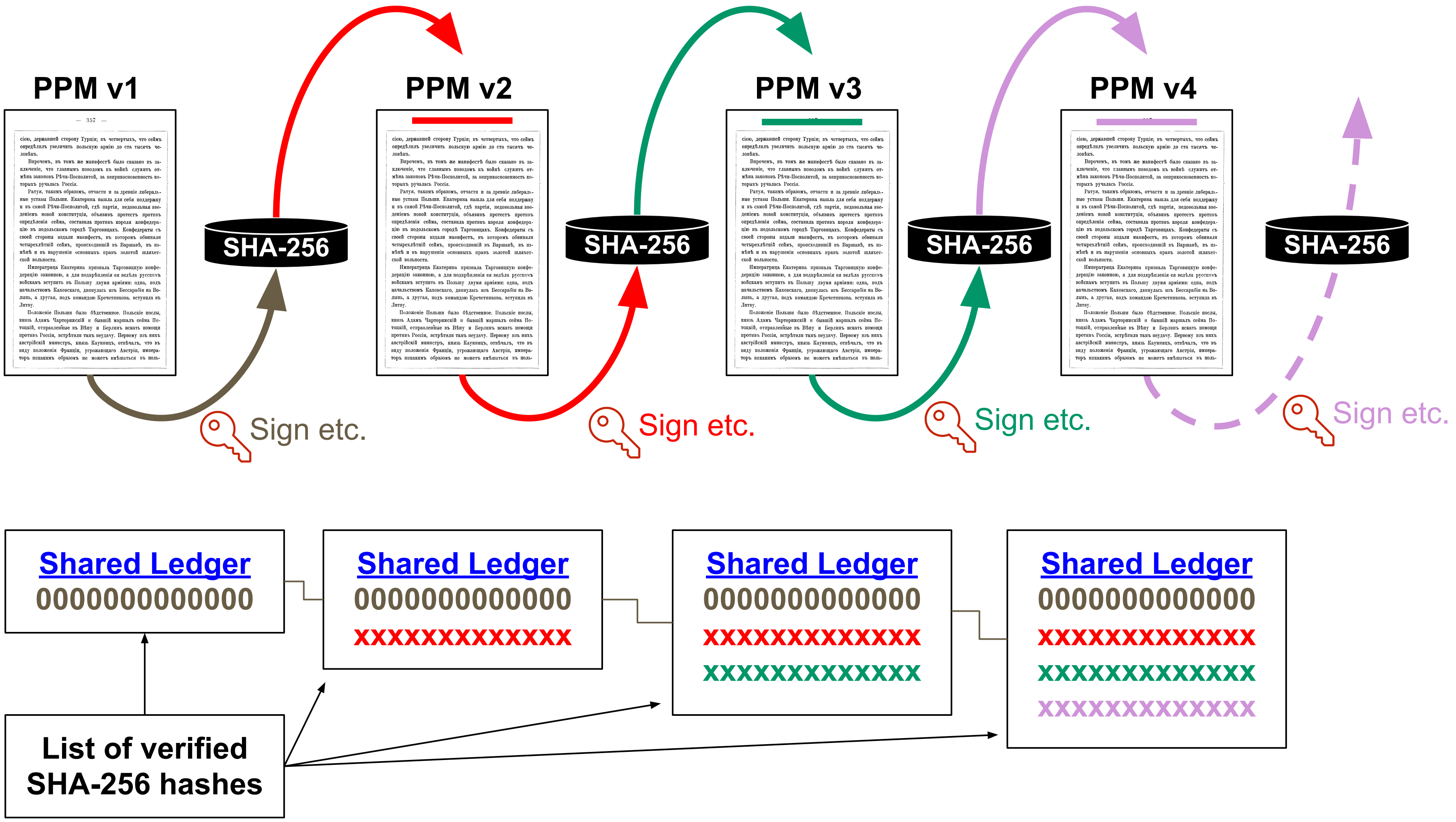

PPM v1
Let's Hash!
SHA-256
Sign with author Private Key to verify authenticity
Hash: 09592b438bfe8ac1fd
Verified & Recorded on Distributed Shared Ledger
$$ Reward
PPM v2


PPM v1
Let's Hash!
SHA-256
Verified & Recorded on
Distributed Shared Ledger
$$ Reward
PPM v2
Hash: 09592b438bfe8ac1fd
Sign with author Private
Key to verify authenticity
SHA-256
Hash: fa1960e7a6b57ee967


PPM v1
Let's Hash!
SHA-256
Verified & Recorded on Distributed Shared Ledger
$$ Reward
Hash: 09592b438bfe8ac1fd
PPM v2
SHA-256
$$ Reward
Verified & Recorded on Distributed Shared Ledger
Hash: fa1960e7a6b57ee967
PPM v3
Sign with author Private Key to verify authenticity



PPM v1
Let's Hash!
SHA-256
Verified & Recorded on Distributed Shared Ledger
Hash: 09592b438bfe8ac1fd
PPM v2
SHA-256
Verified & Recorded on Distributed Shared Ledger
Hash: fa1960e7a6b57ee967
PPM v3
SHA-256
Sign etc.


PPM v1
PPM v2
PPM v3
PPM v4
SHA-256
SHA-256
SHA-256
SHA-256
Sign etc.
Sign etc.
Sign etc.
Sign etc.
Shared Ledger
0000000000000
Shared Ledger
0000000000000
xxxxxxxxxxxxx
Shared Ledger
0000000000000
xxxxxxxxxxxxx
xxxxxxxxxxxxx
Shared Ledger
0000000000000
xxxxxxxxxxxxx
xxxxxxxxxxxxx
xxxxxxxxxxxxx
List of verified SHA-256 hashes

One of the earliest papers on "Blockchain"

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South Street
Morristown, N.J. 07960-1910

Abstract

The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

*Appeared, with minor editorial changes, in *Journal of Cryptology*, Vol. 3, No. 2, pp. 99–111, 1991.

One of the earliest papers on "Blockchain"

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South Street
Morristown, N.J. 07960-1910

Abstract

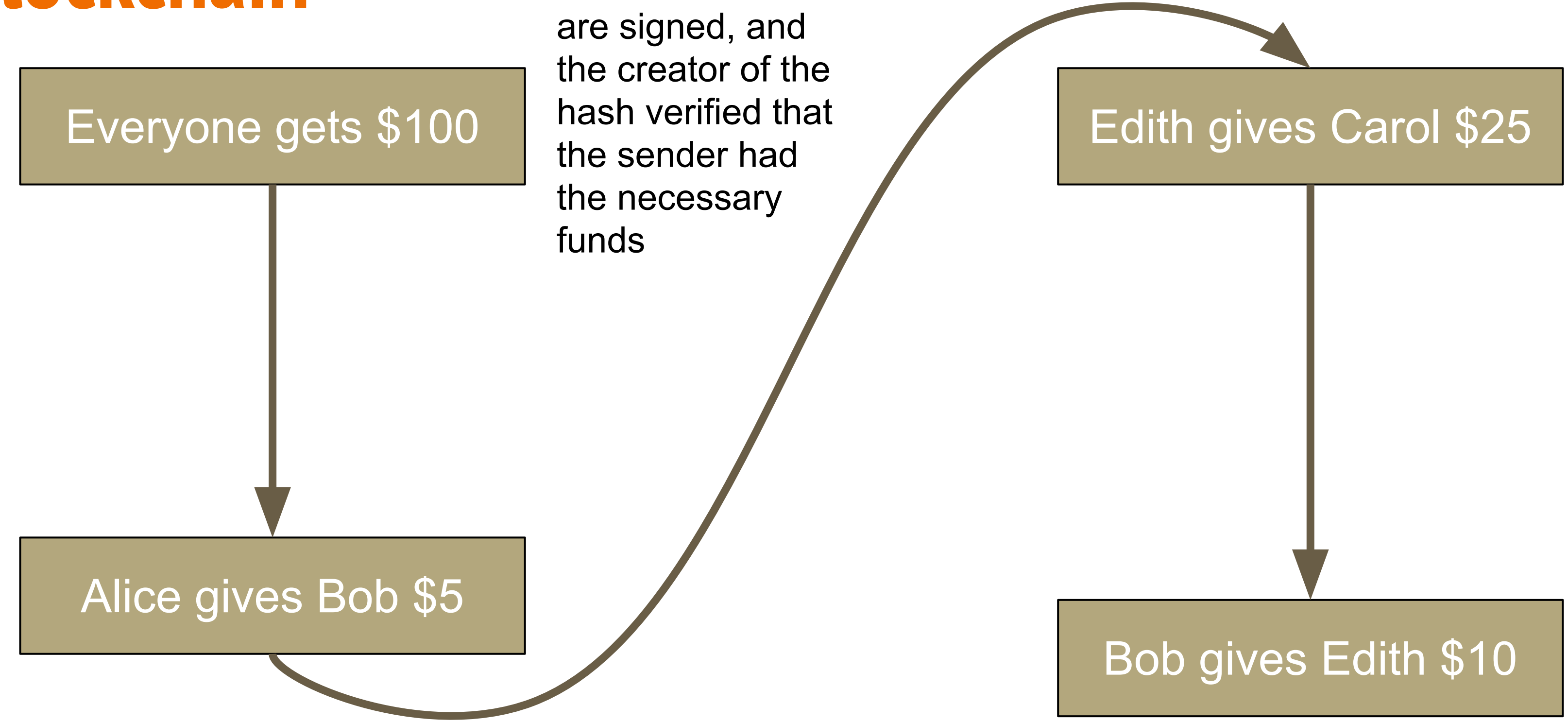
The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

*Appeared, with minor editorial changes, in *Journal of Cryptology*, Vol. 3, No. 2, pp. 99–111, 1991.

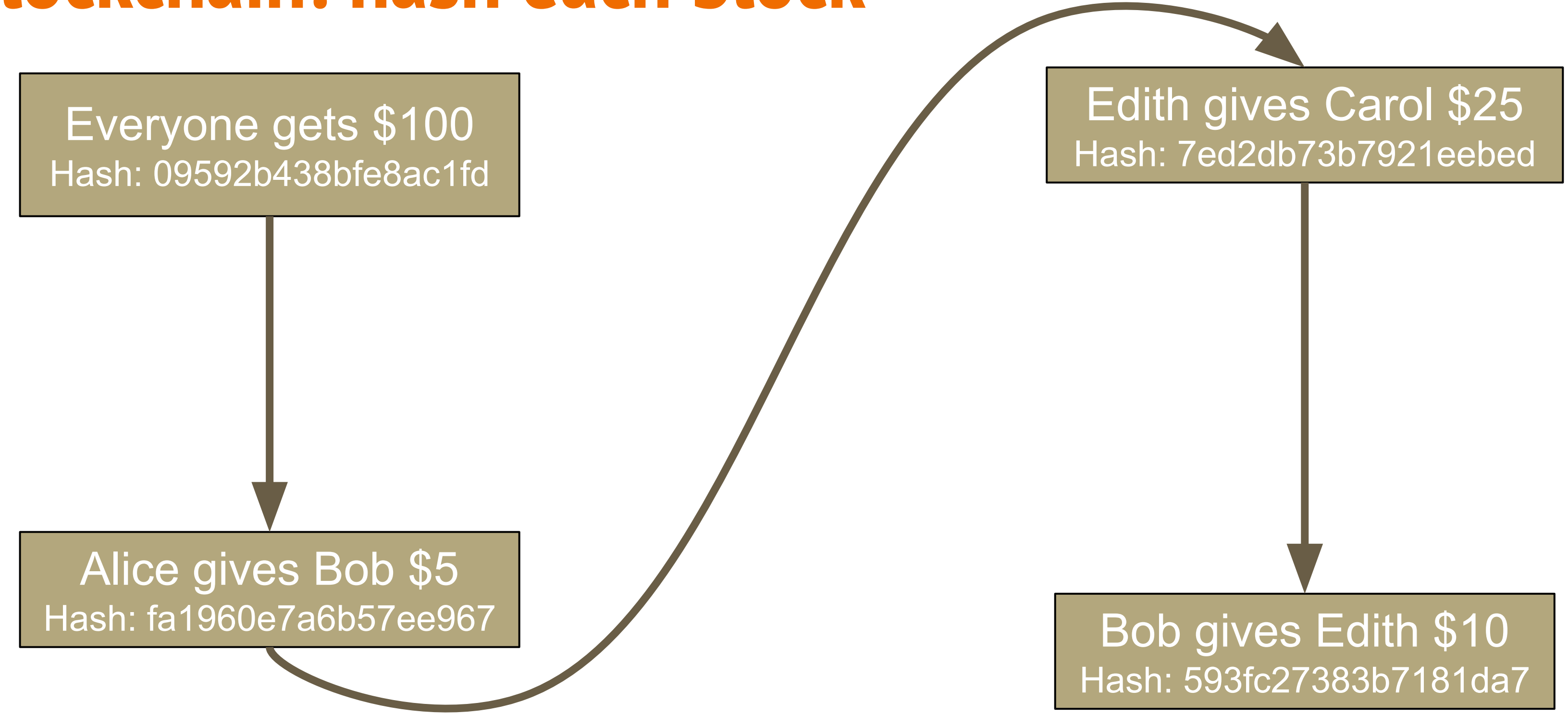
Let’s power on …

Blockchain

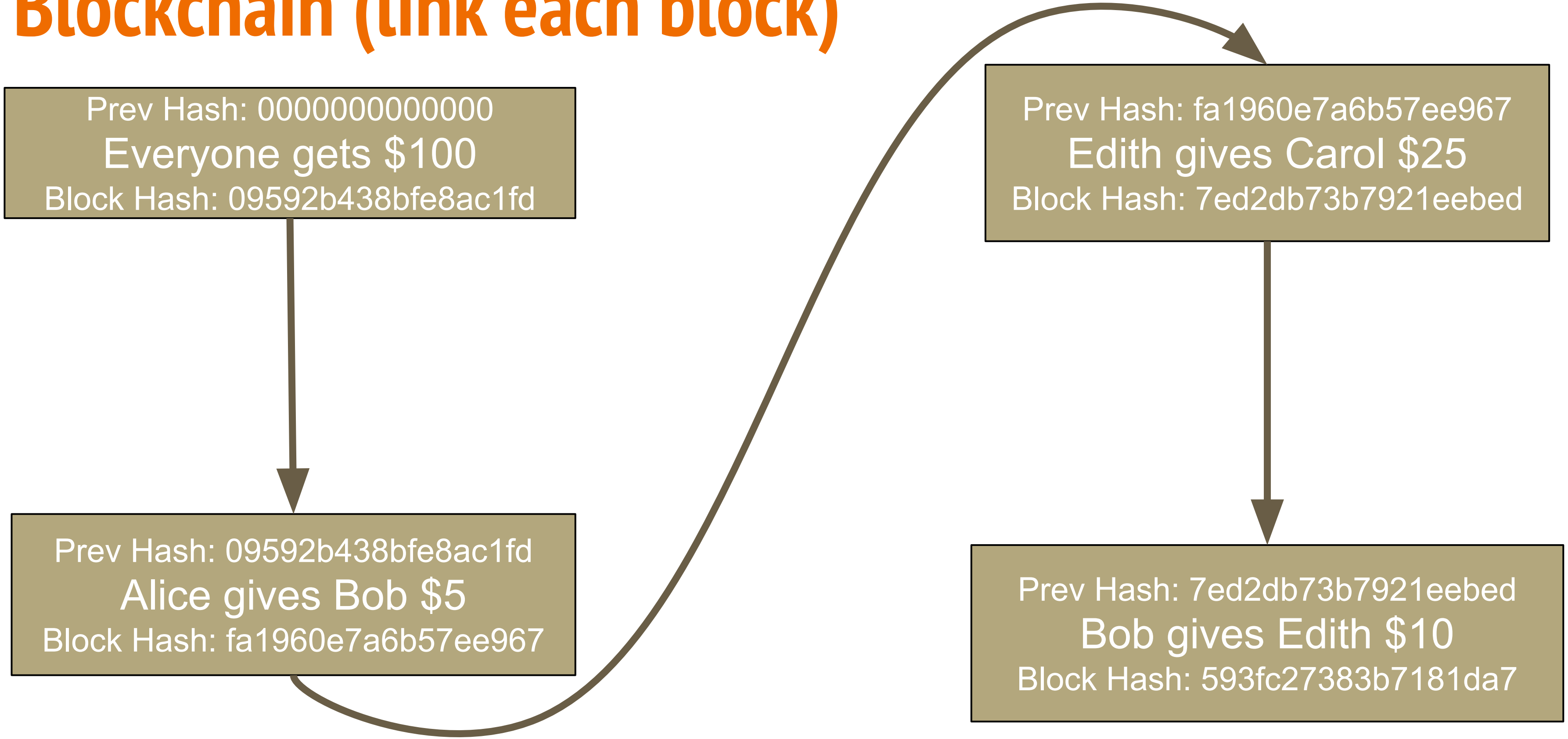
Assume all transactions here are signed, and the creator of the hash verified that the sender had the necessary funds



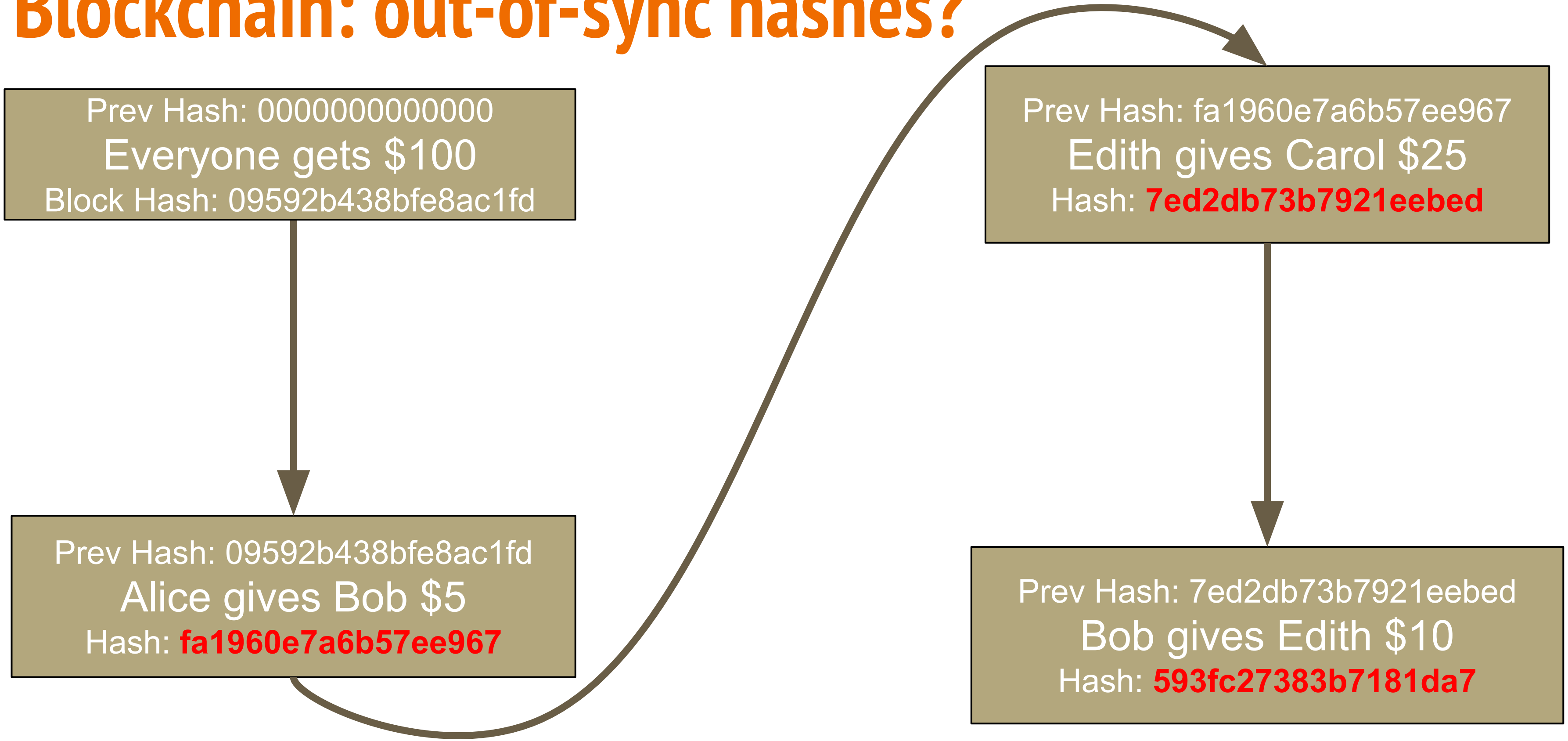
Blockchain: hash each block



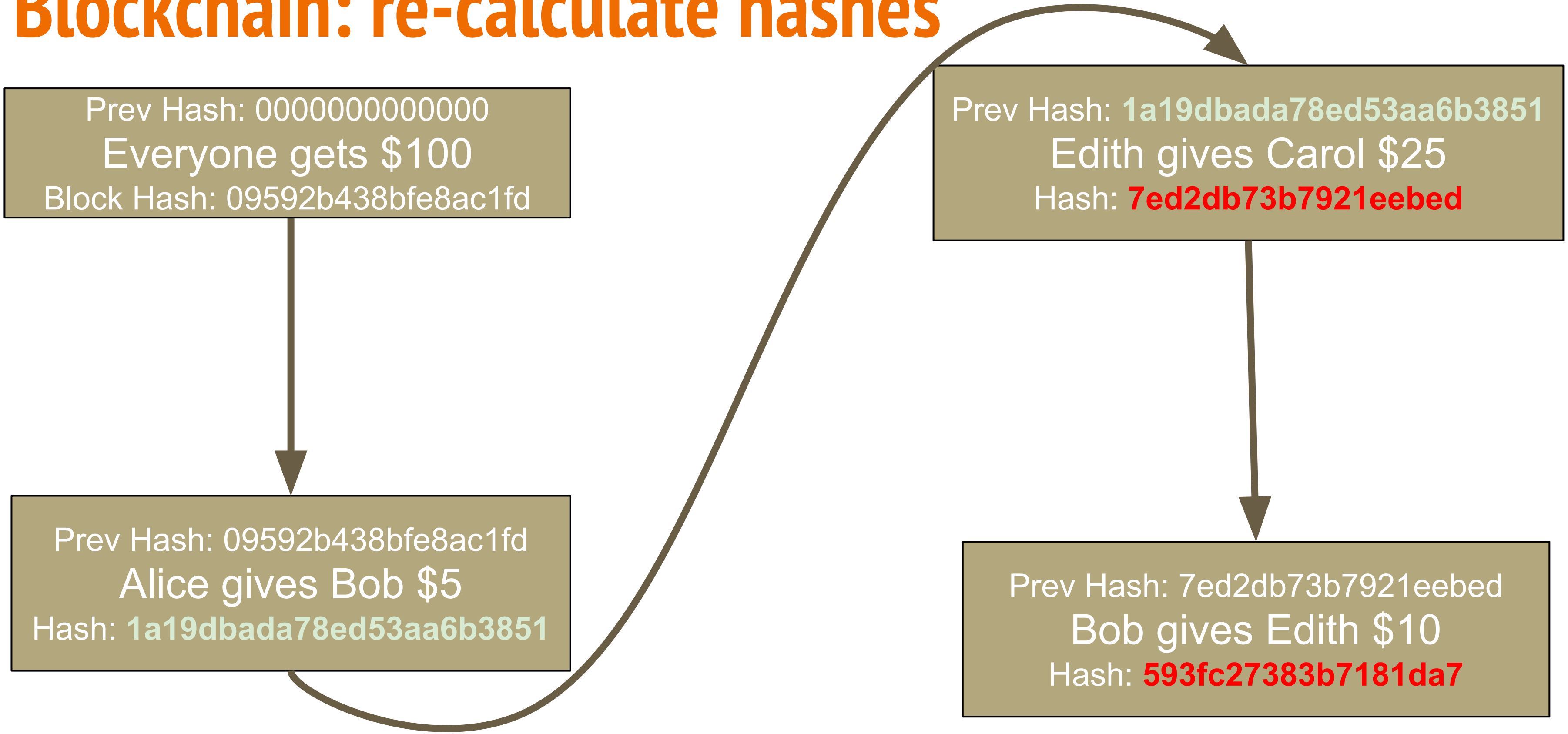
Blockchain (link each block)



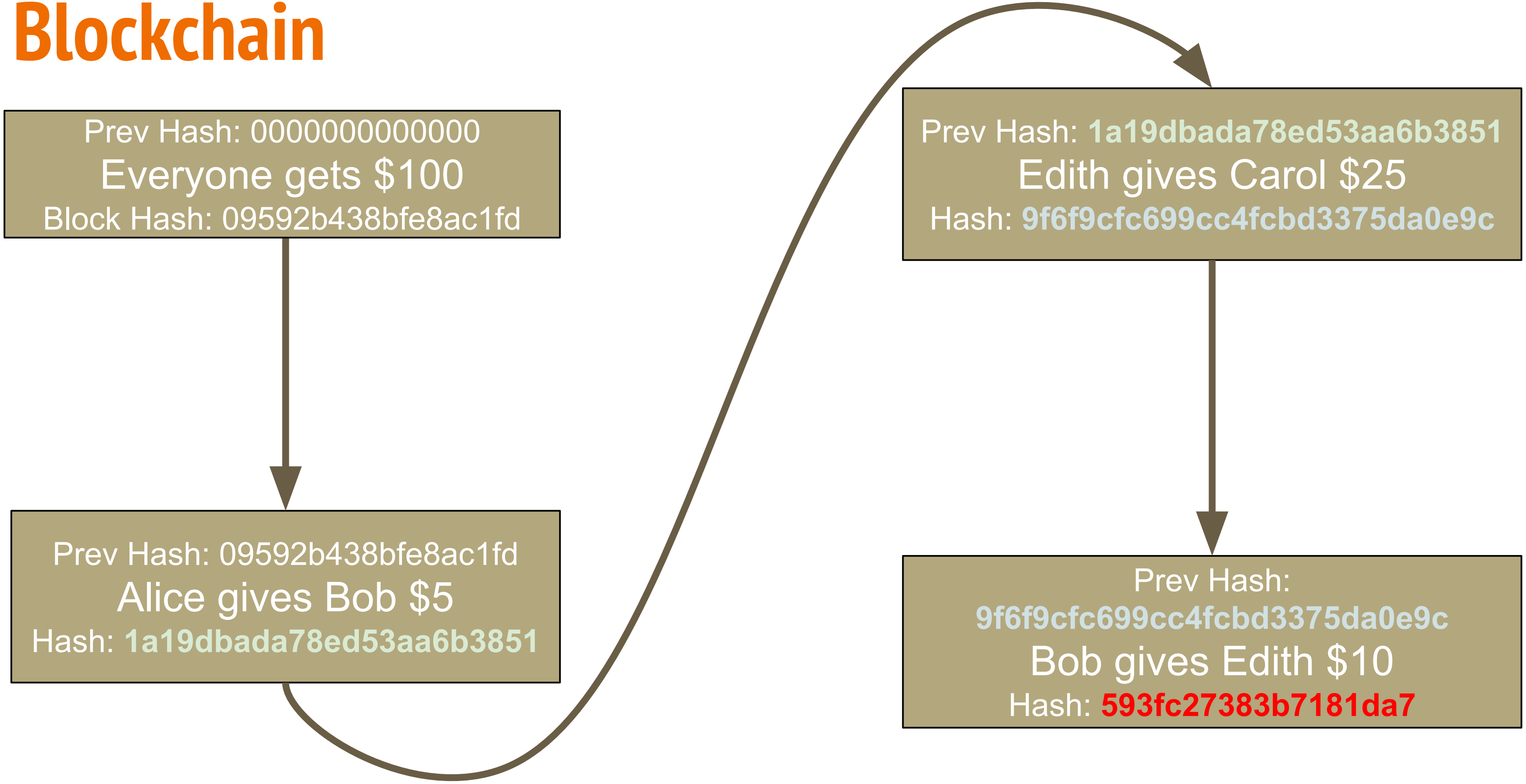
Blockchain: out-of-sync hashes?



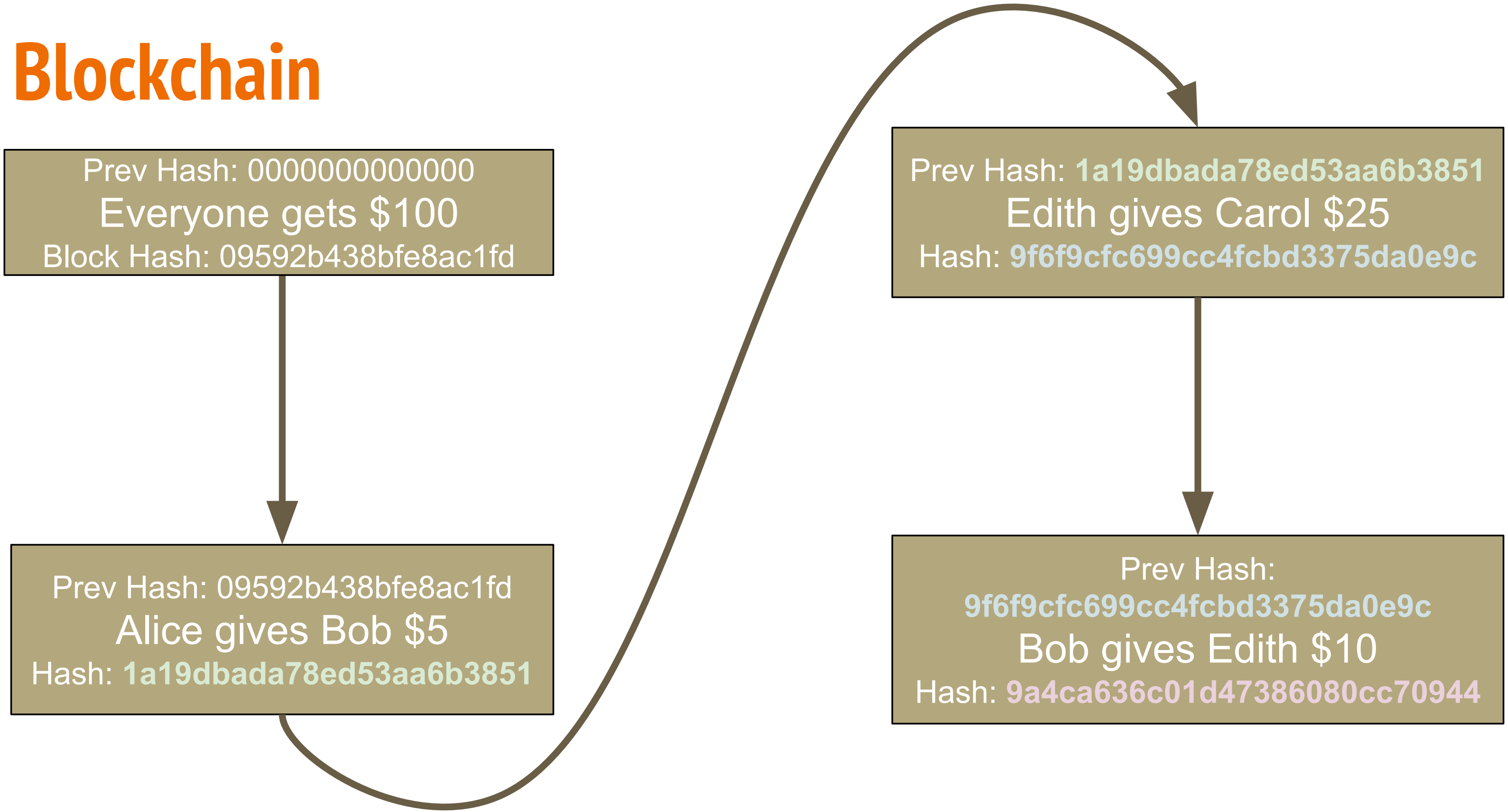
Blockchain: re-calculate hashes



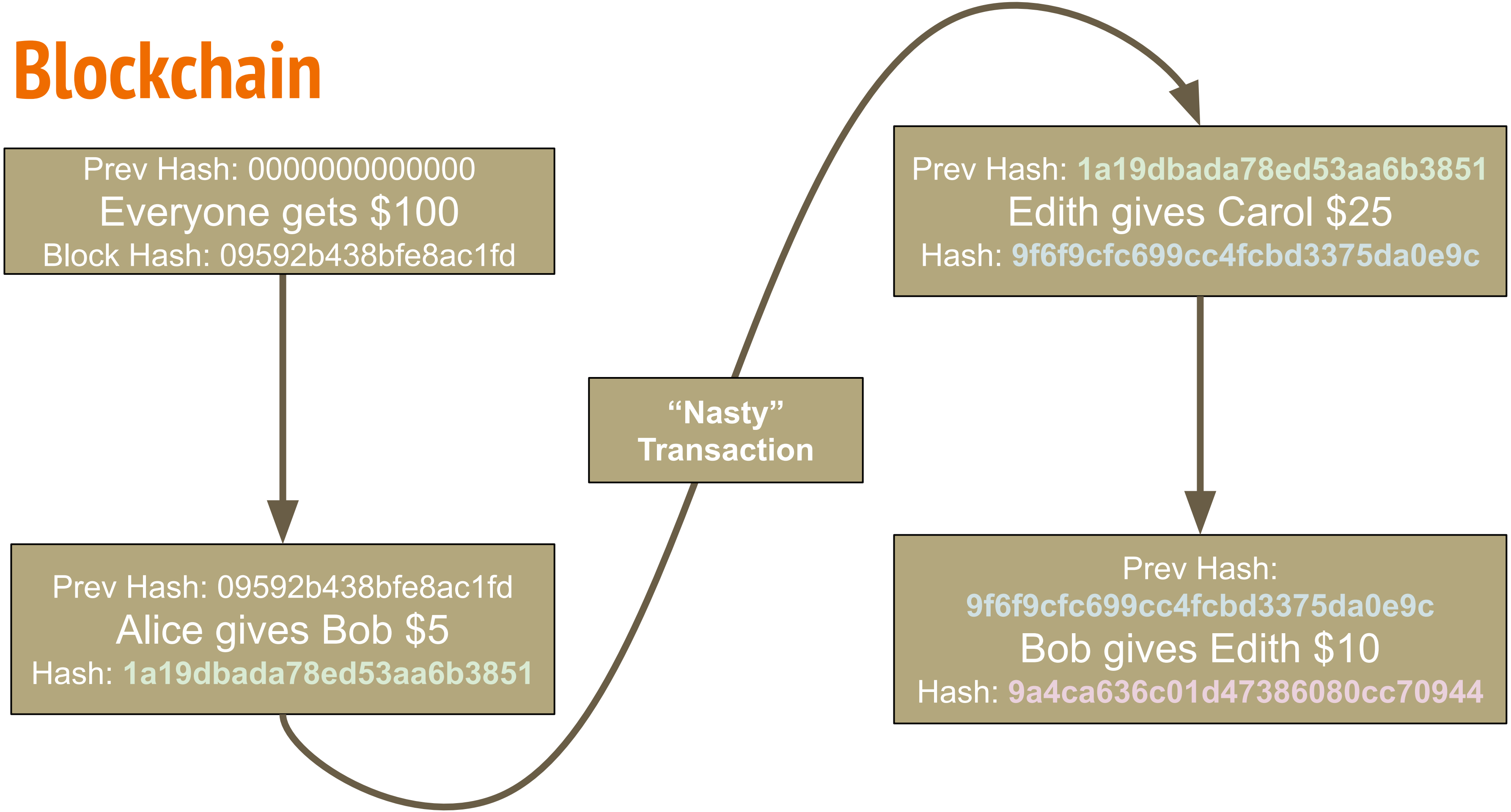
Blockchain


Prev Hash: 0000000000000
Everyone gets $100
Block Hash: 09592b438bfe8ac1fd
Prev Hash: 09592b438bfe8ac1fd
Alice gives Bob $5
Hash: 1a19dbada78ed53aa6b3851
Prev Hash: 1a19dbada78ed53aa6b3851
Edith gives Carol $25
Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Prev Hash:
9f6f9cfc699cc4fcbd3375da0e9c
Bob gives Edith $10
Hash: 593fc27383b7181da7

Blockchain

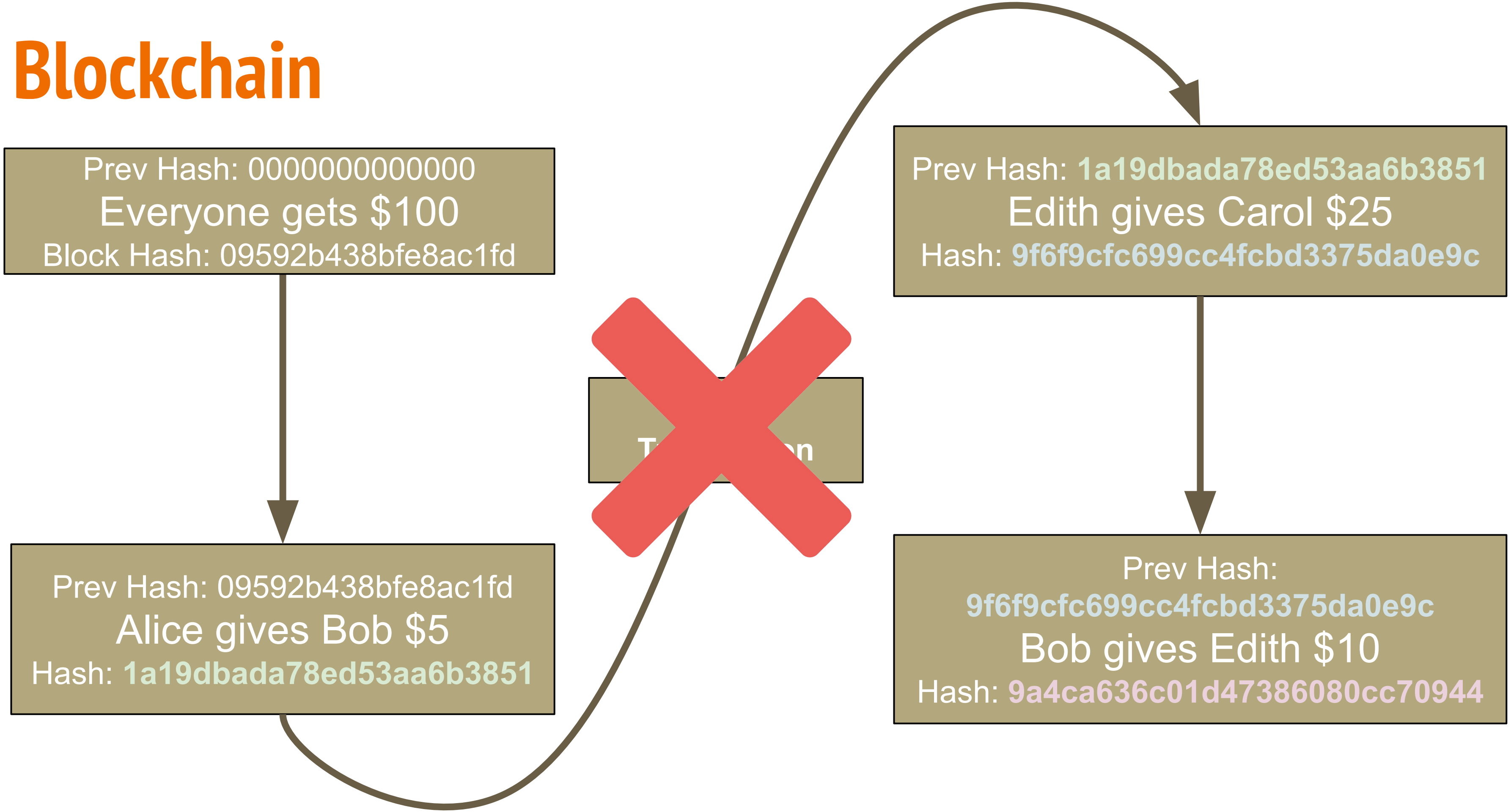

Prev Hash: 0000000000000
Everyone gets $100
Block Hash: 09592b438bfe8ac1fd
Prev Hash: 09592b438bfe8ac1fd
Alice gives Bob $5
Hash: 1a19dbada78ed53aa6b3851
Prev Hash: 1a19dbada78ed53aa6b3851
Edith gives Carol $25
Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Prev Hash:
9f6f9cfc699cc4fcbd3375da0e9c
Bob gives Edith $10
Hash: 9a4ca636c01d47386080cc70944

Blockchain

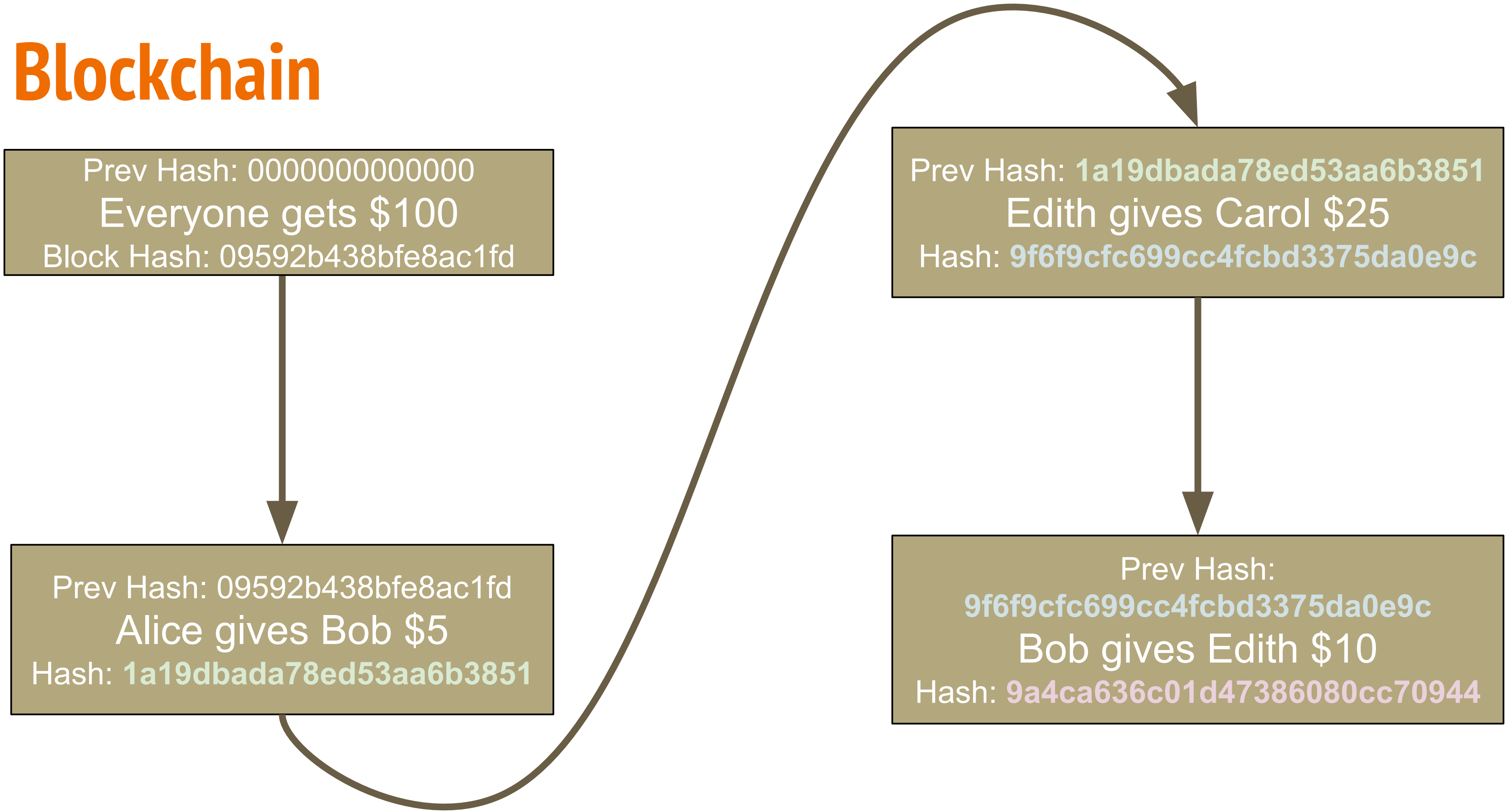


Prev Hash: 0000000000000
Everyone gets $100
Block Hash: 09592b438bfe8ac1fd
Prev Hash: 09592b438bfe8ac1fd
Alice gives Bob $5
Hash: 1a19dbada78ed53aa6b3851
“Nasty” Transaction
Prev Hash: 1a19dbada78ed53aa6b3851
Edith gives Carol $25
Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Prev Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Bob gives Edith $10
Hash: 9a4ca636c01d47386080cc70944

Blockchain

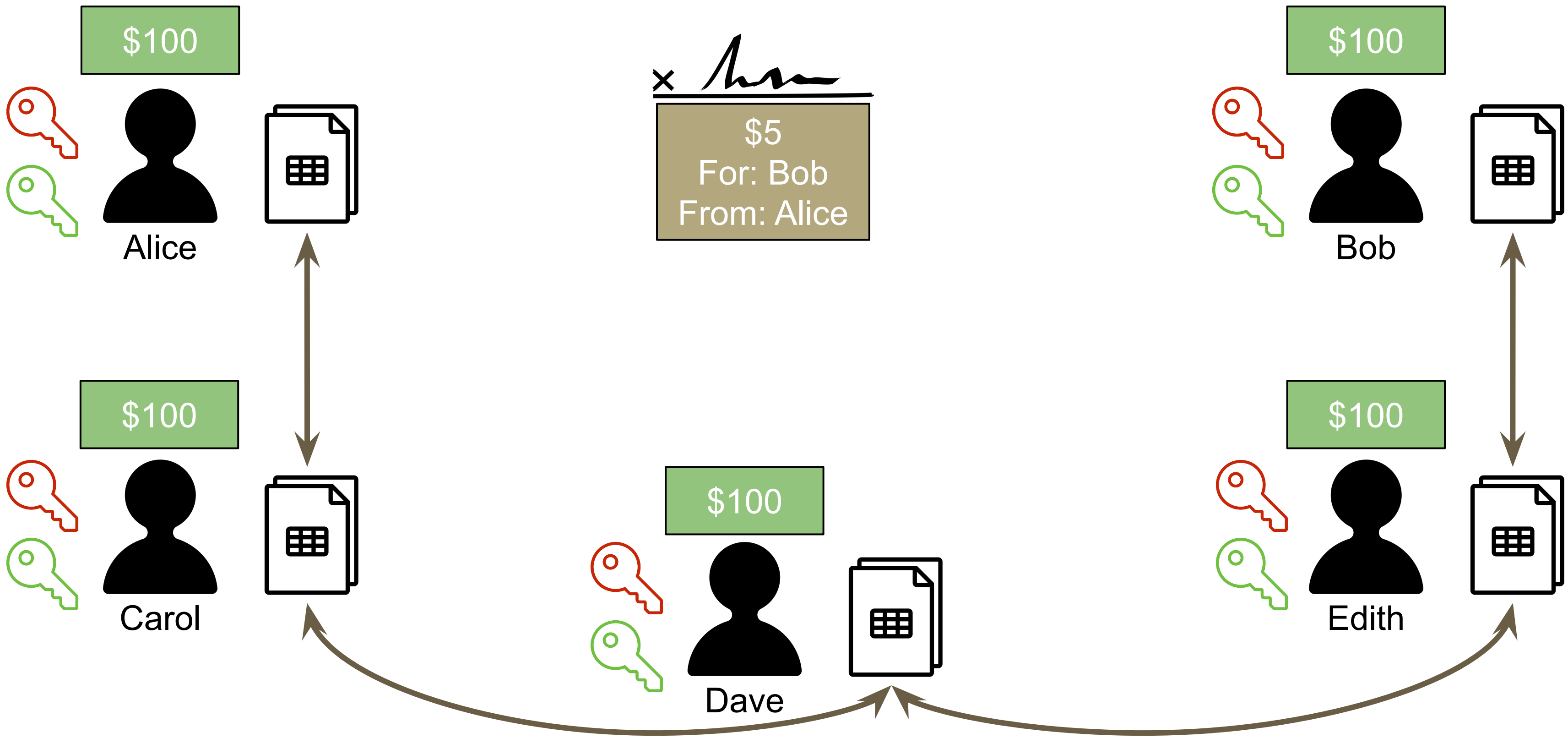

Prev Hash: 0000000000000
Everyone gets $100
Block Hash: 09592b438bfe8ac1fd
Prev Hash: 09592b438bfe8ac1fd
Alice gives Bob $5
Hash: 1a19dbada78ed53aa6b3851
Prev Hash: 1a19dbada78ed53aa6b3851
Edith gives Carol $25
Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Prev Hash:
9f6f9cfc699cc4fcbd3375da0e9c
Bob gives Edith $10
Hash: 9a4ca636c01d47386080cc70944

Blockchain

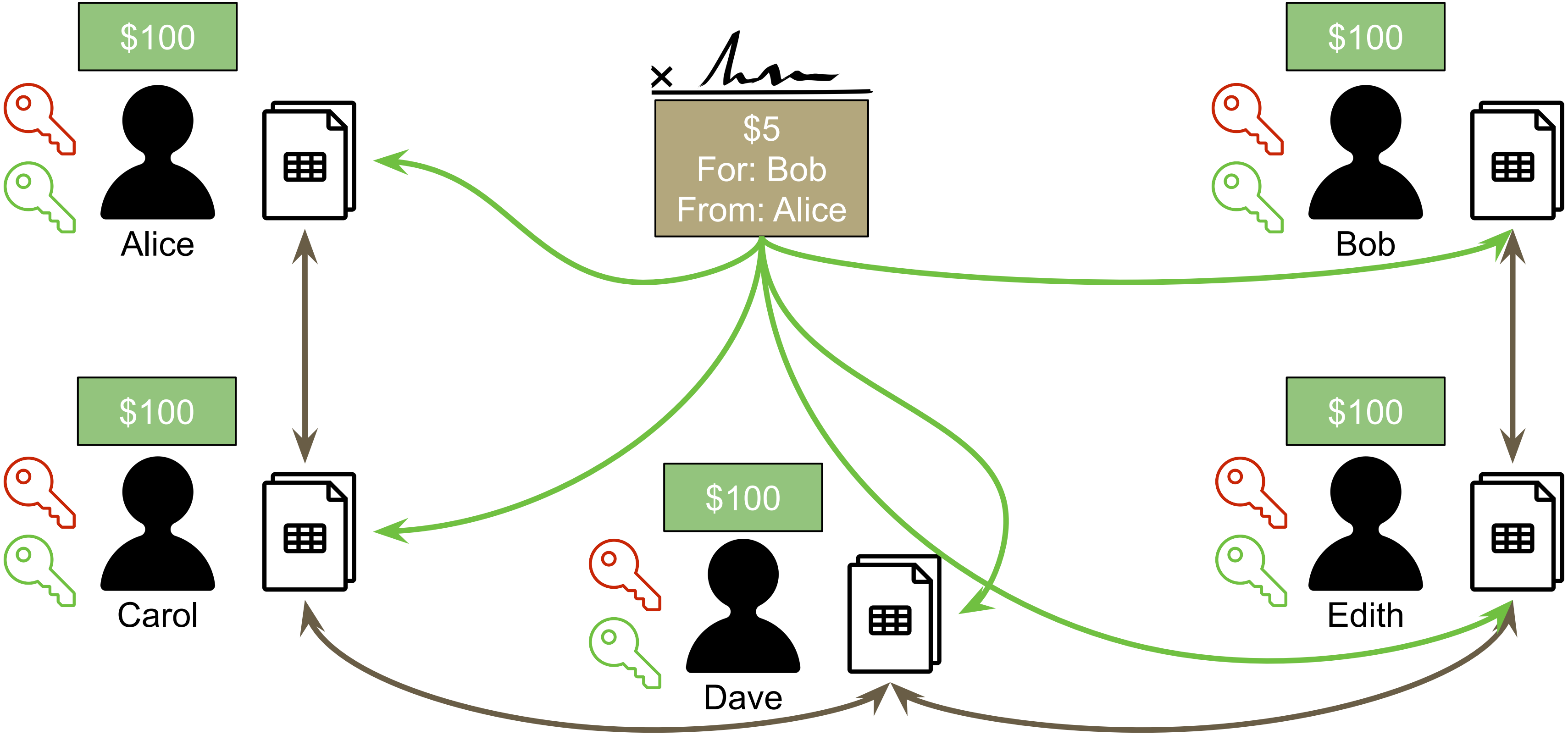

Prev Hash: 0000000000000
Everyone gets $100
Block Hash: 09592b438bfe8ac1fd
Prev Hash: 09592b438bfe8ac1fd
Alice gives Bob $5
Hash: 1a19dbada78ed53aa6b3851
Prev Hash: 1a19dbada78ed53aa6b3851
Edith gives Carol $25
Hash: 9f6f9cfc699cc4fcbd3375da0e9c
Prev Hash:
9f6f9cfc699cc4fcbd3375da0e9c
Bob gives Edith $10
Hash: 9a4ca636c01d47386080cc70944

DEMO: Blockchain

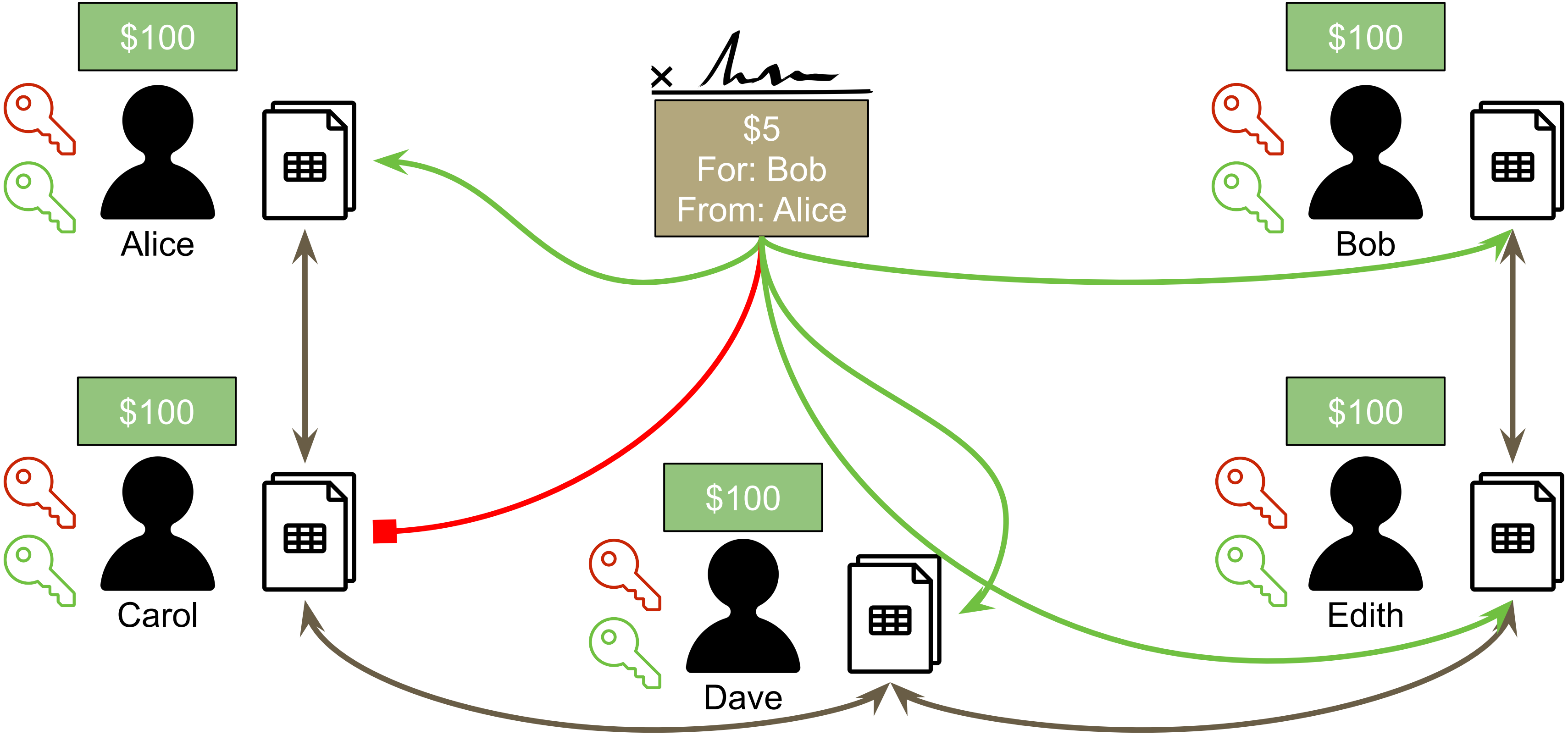
Blockchains



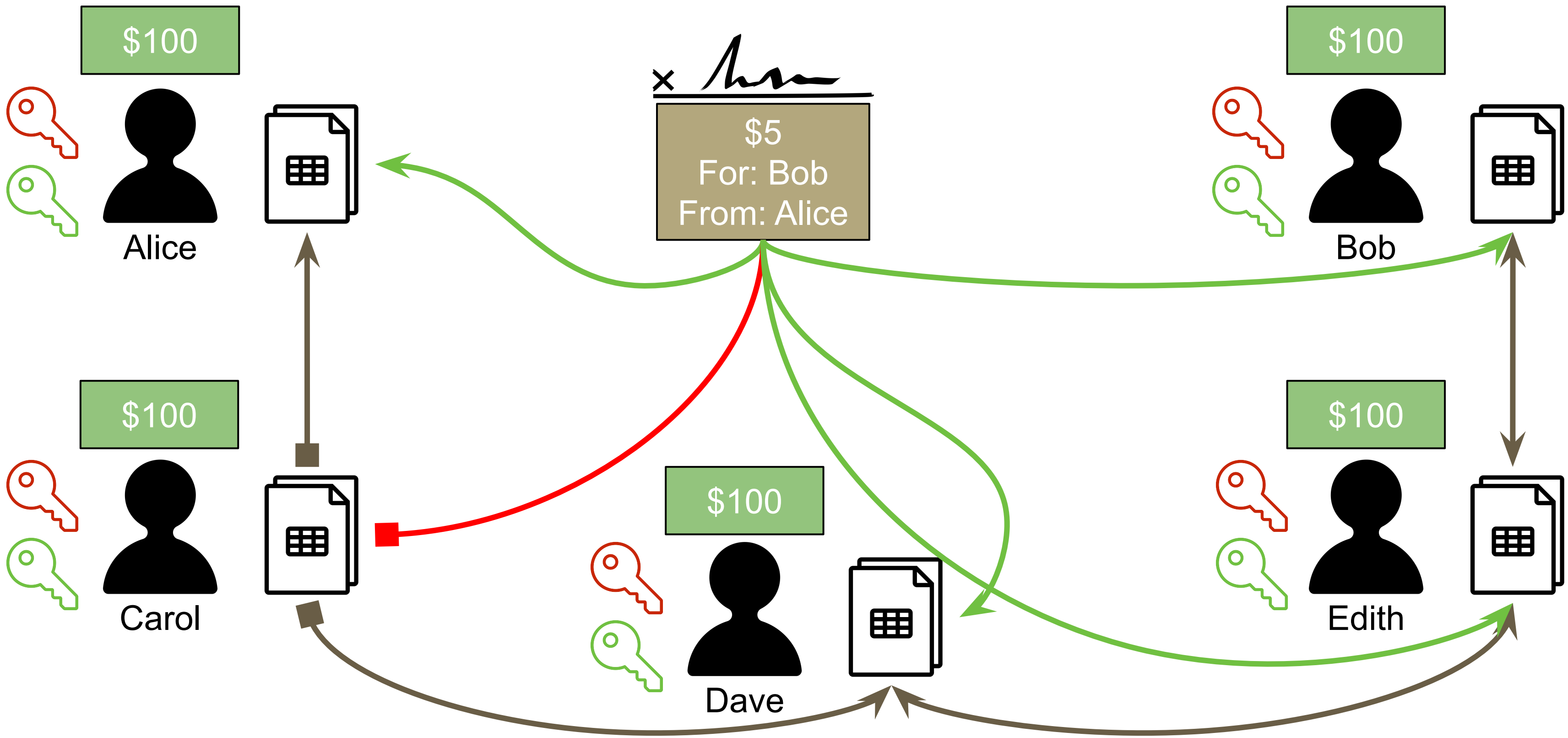
Blockchains



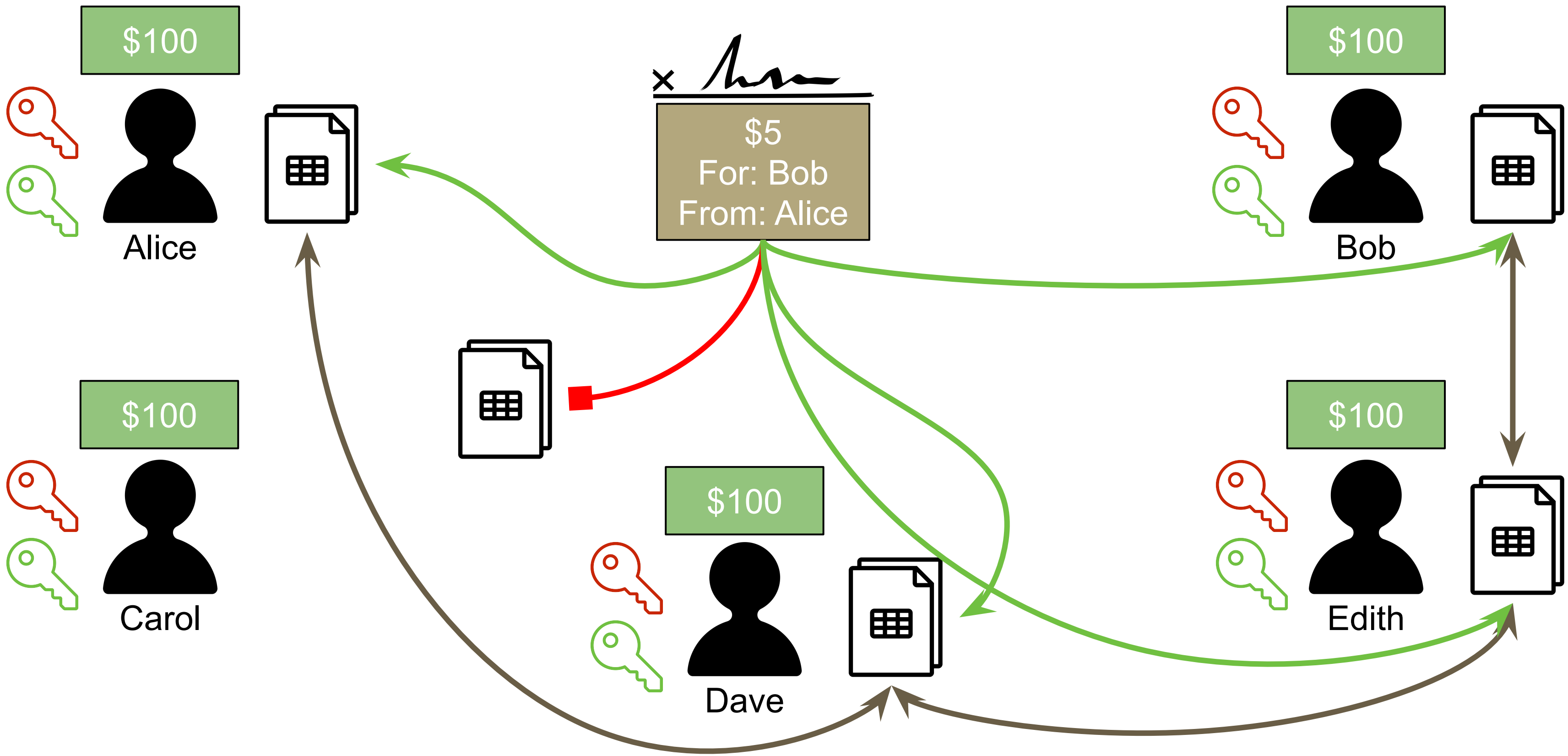
Blockchains



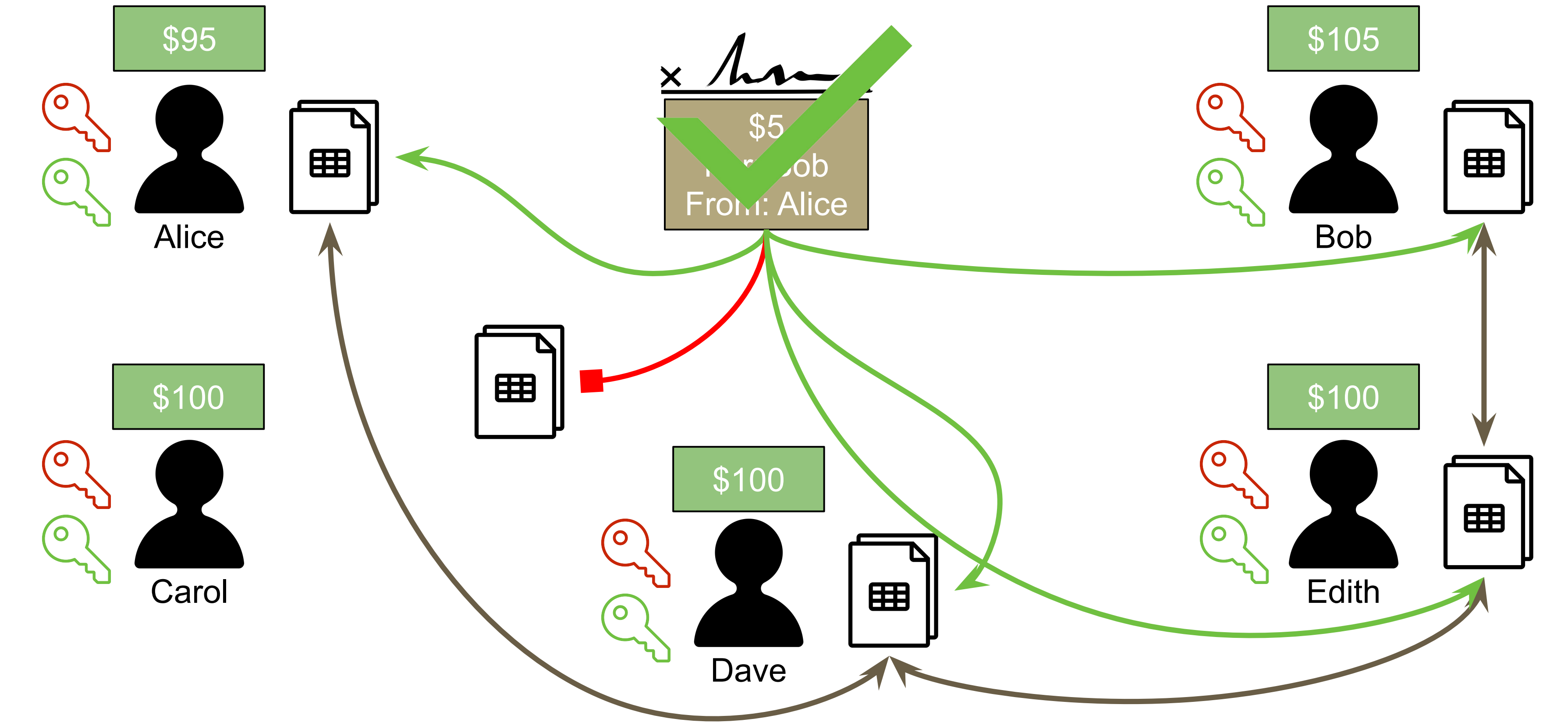
Blockchains



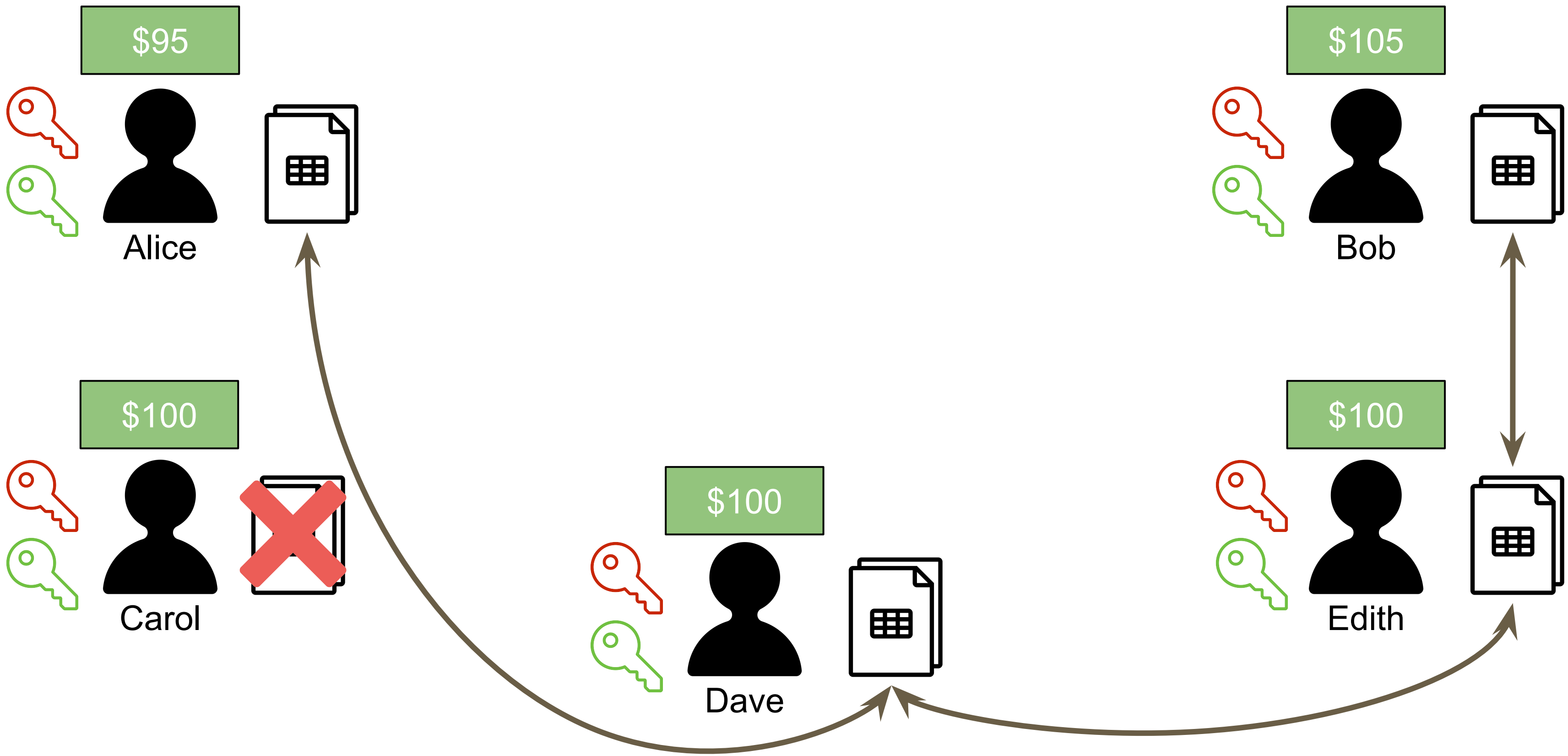
Blockchains



Blockchains



Blockchains



Blockchain Recap

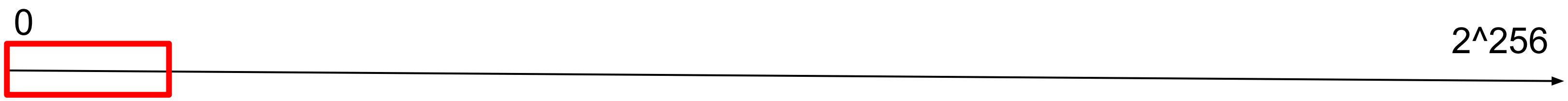
1. The transaction is broadcasted to the world.
2. Each node that receives the broadcast verifies via the signature and their copy of the ledger that the sending party has the funds to send that amount of money, and that the transaction actually came from the sending party.
3. Each updates their ledger in a cryptographically consistent and verifiable way, forever cementing the transaction as part of the chain.
4. Once the majority of nodes have updated their ledger with the valid transaction, the recipient of the money effectively “has” the new money because they now, according to the ledger shared by the majority, have the funds they need to send a new, valid transaction with the funds they received.

DEMO: Distributed

V. Bitcoin

Leveraging the blockchain to create a decentralized digital crypto-currency.

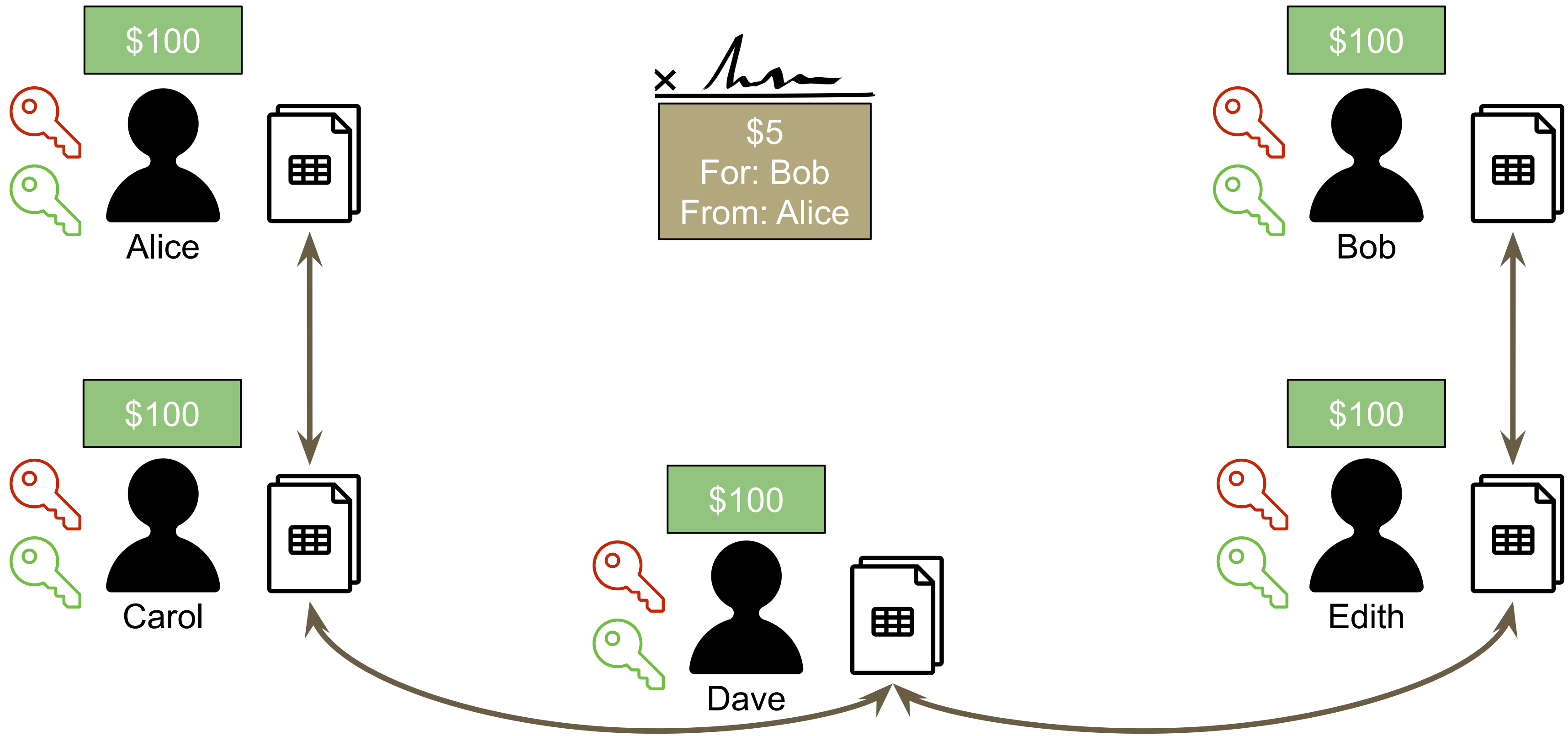
SHA-256 Hash



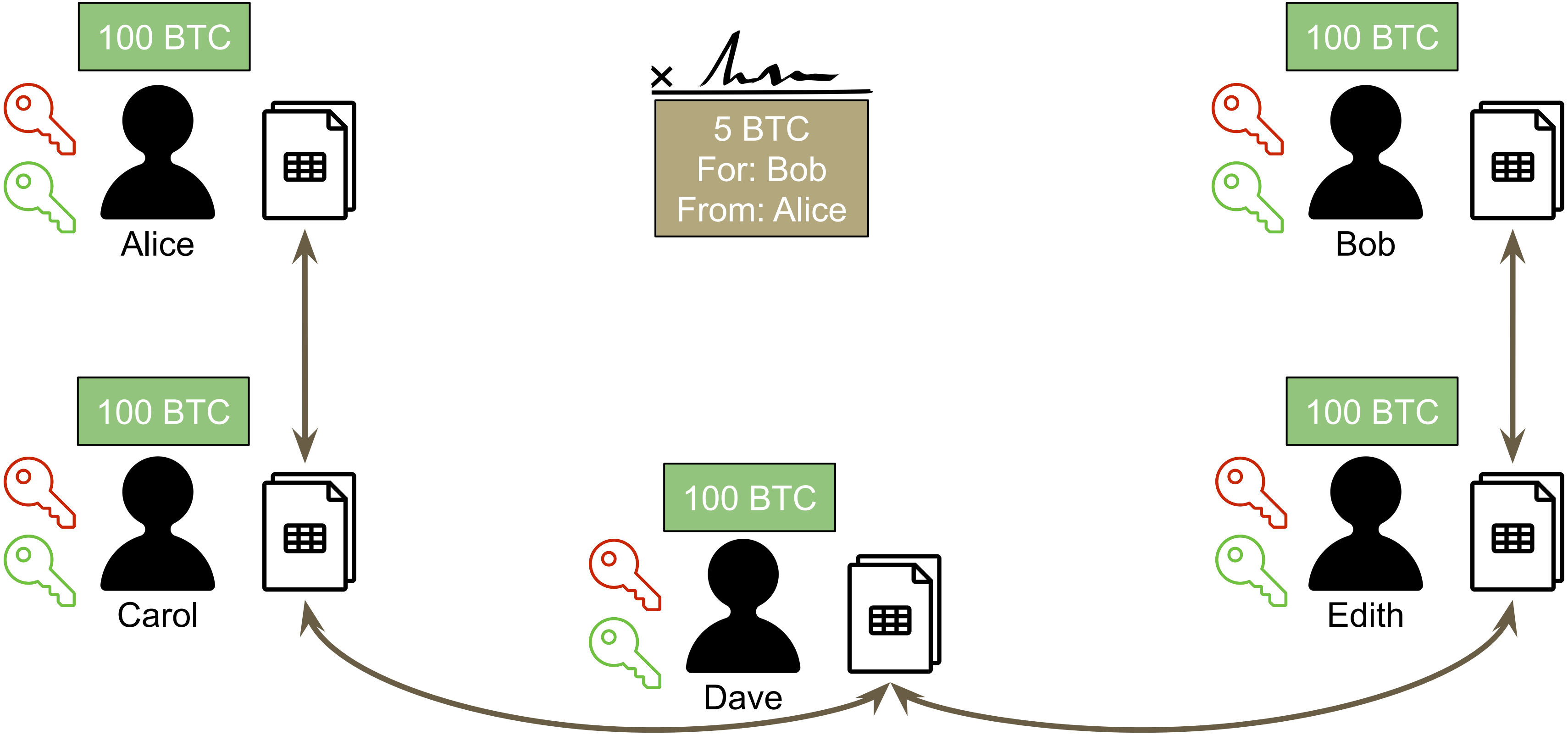
$$\left(\frac{1}{16}\right) \times 2^{256}$$

```
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000001
0000000000000000000000000000000000000000000000000000000000000002
0000000000000000000000000000000000000000000000000000000000000009
000000000000000000000000000000000000000000000000000000000000000a
000000000000000000000000000000000000000000000000000000000000000b
000000000000000000000000000000000000000000000000000000000000000f
0000000000000000000000000000000000000000000000000000000000000010
03b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
1000000000000000000000000000000000000000000000000000000000000000
```

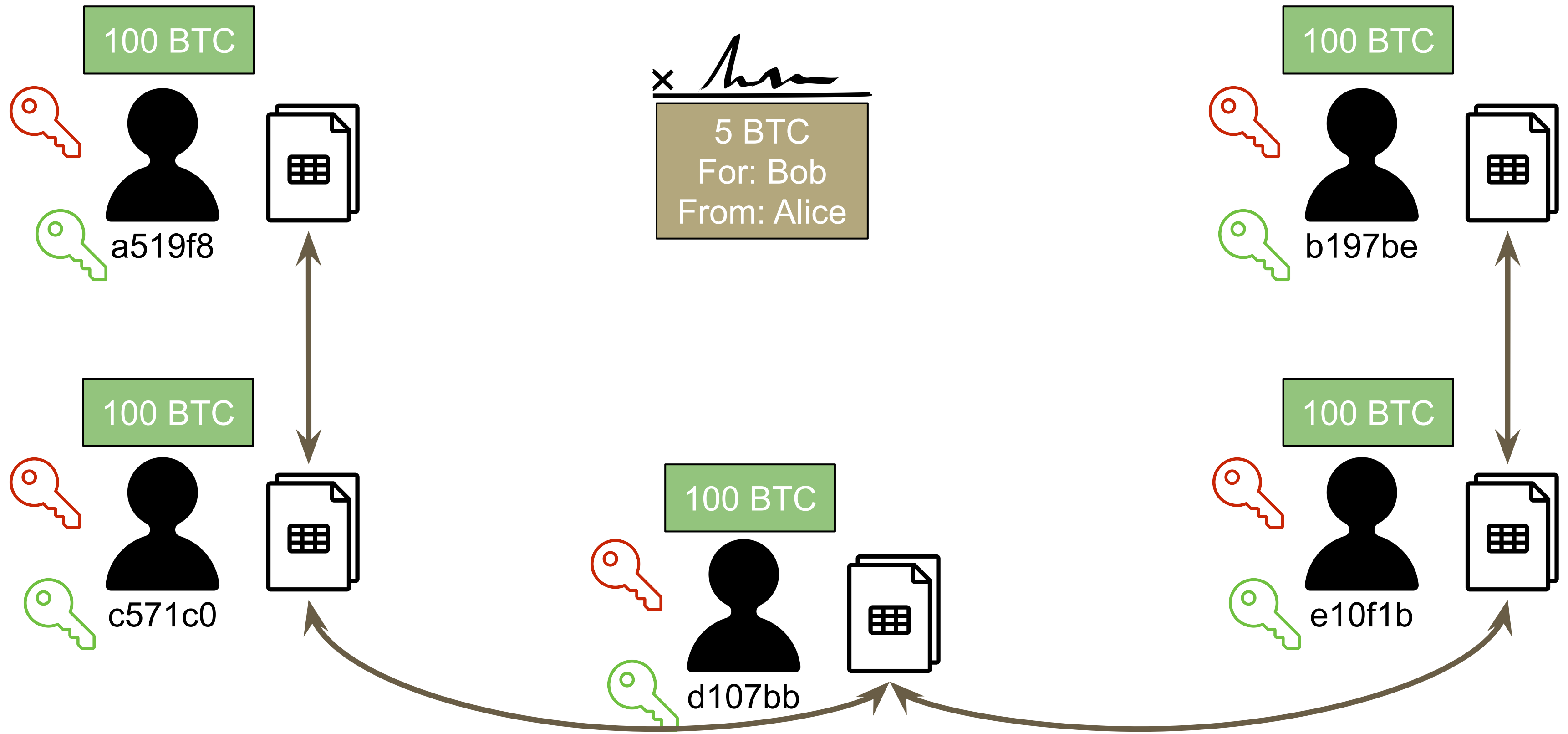
Bitcoin: a shared Blockchain (cooperative)



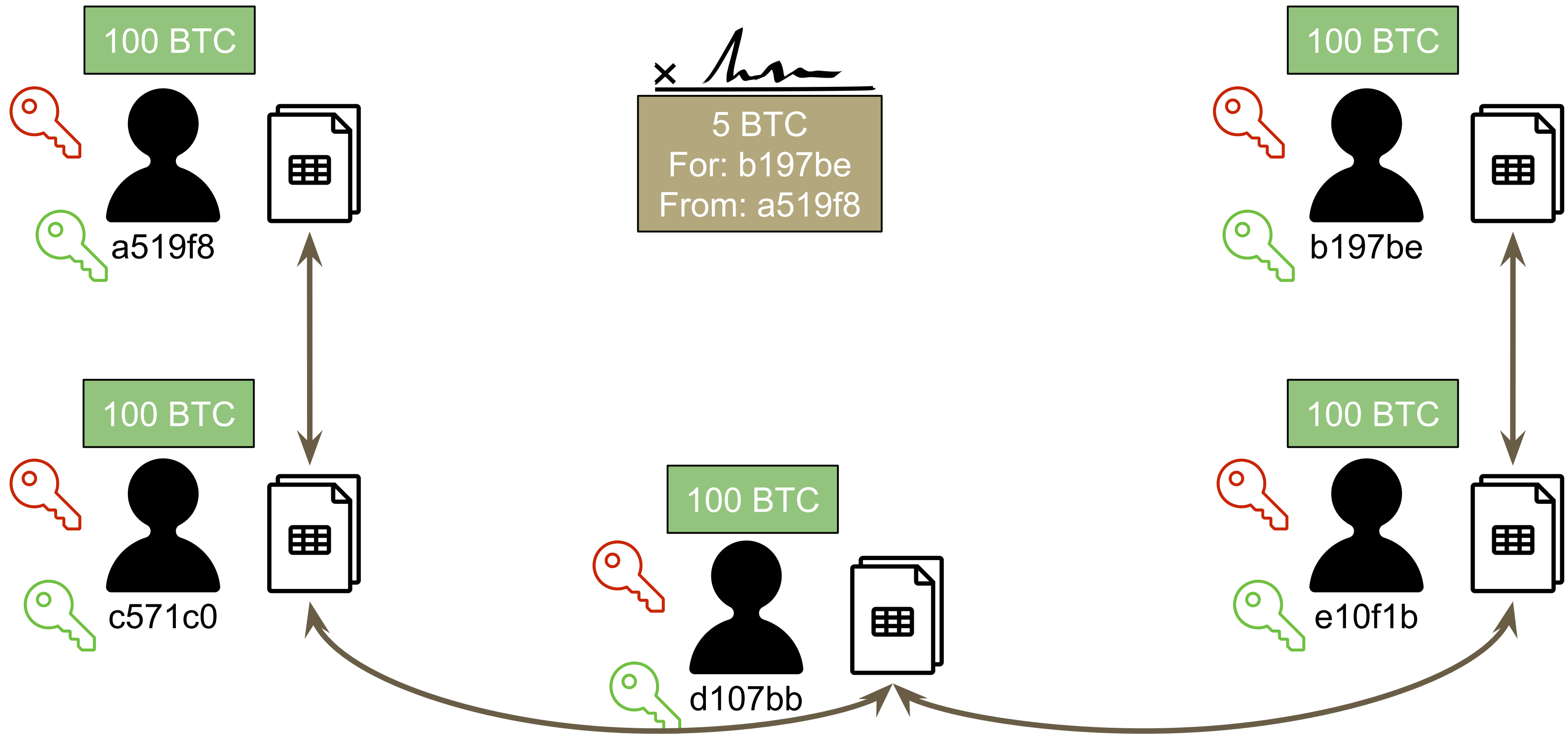
Bitcoin: change USD to Bitcoin



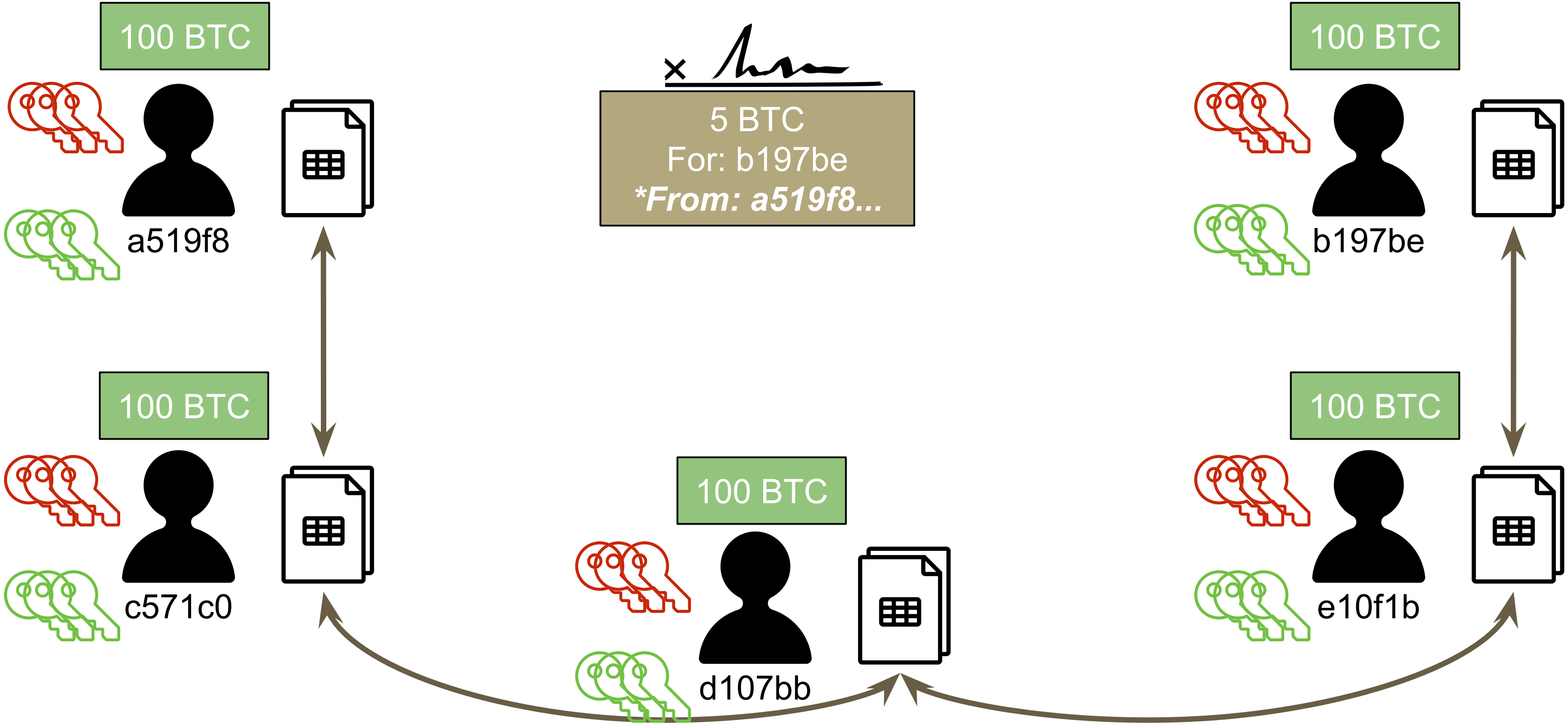
Bitcoin: no names, just (public) keys



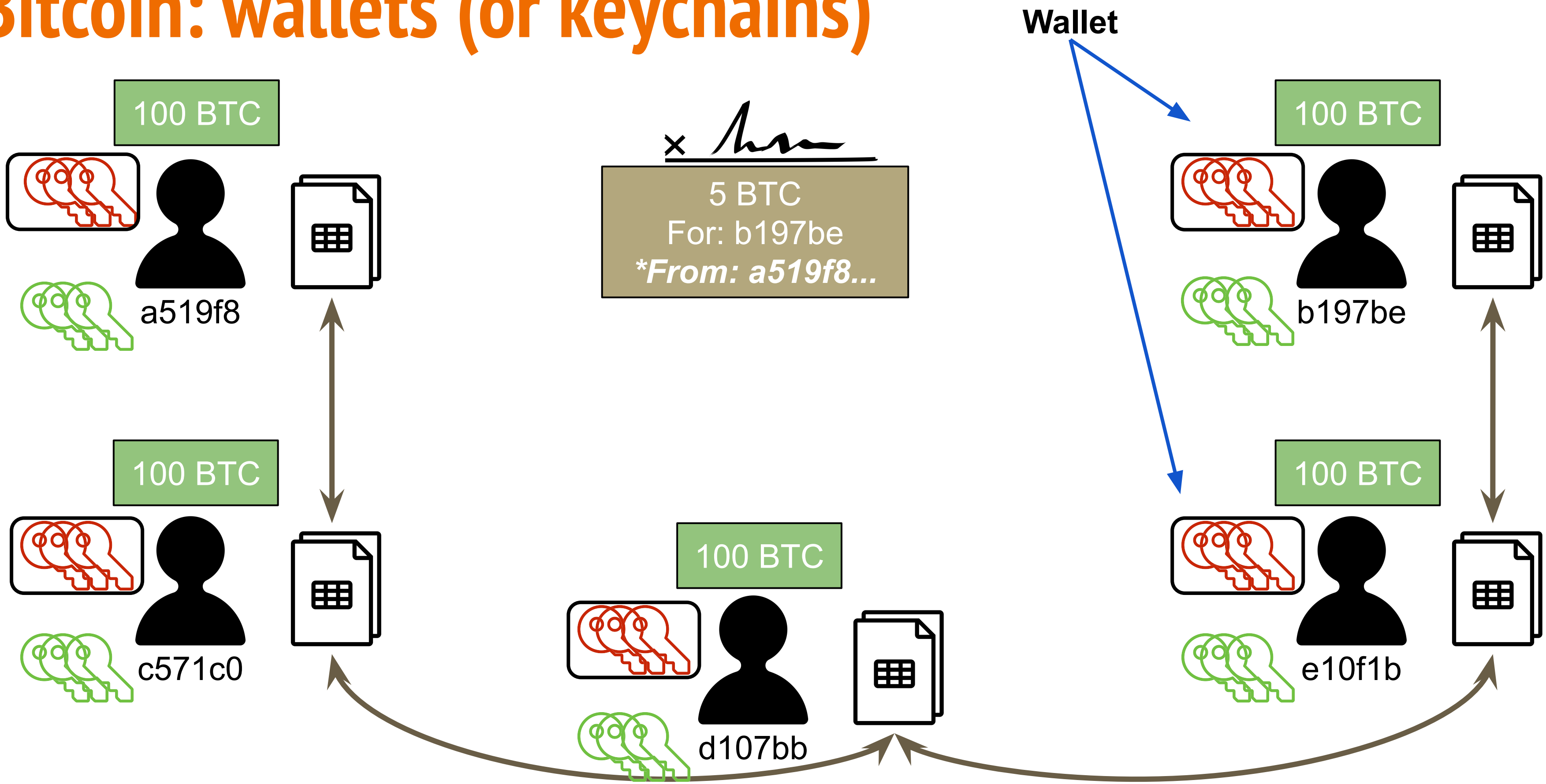
Bitcoin: keys also on the Tx's, no names



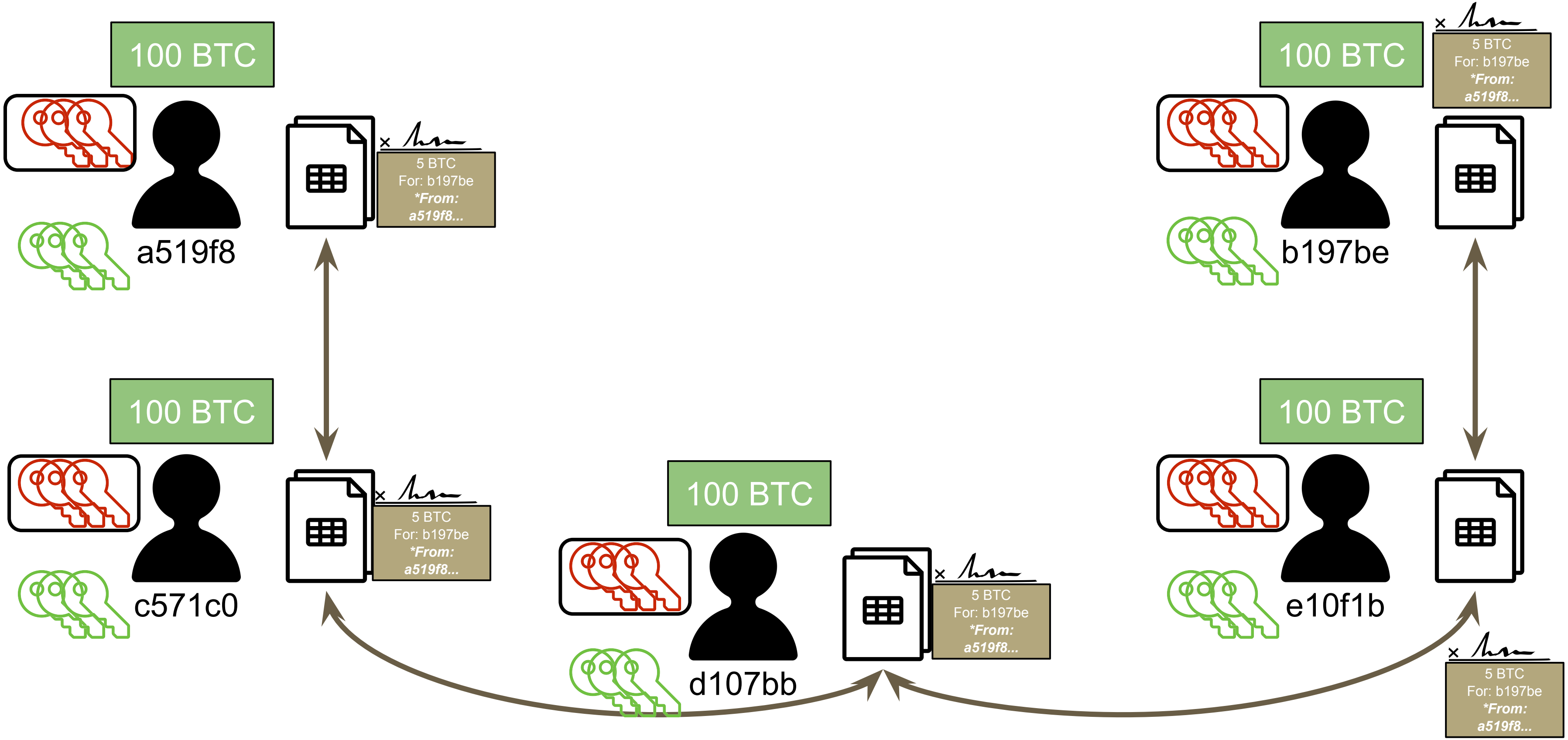
Bitcoin: multiple keys are allowed



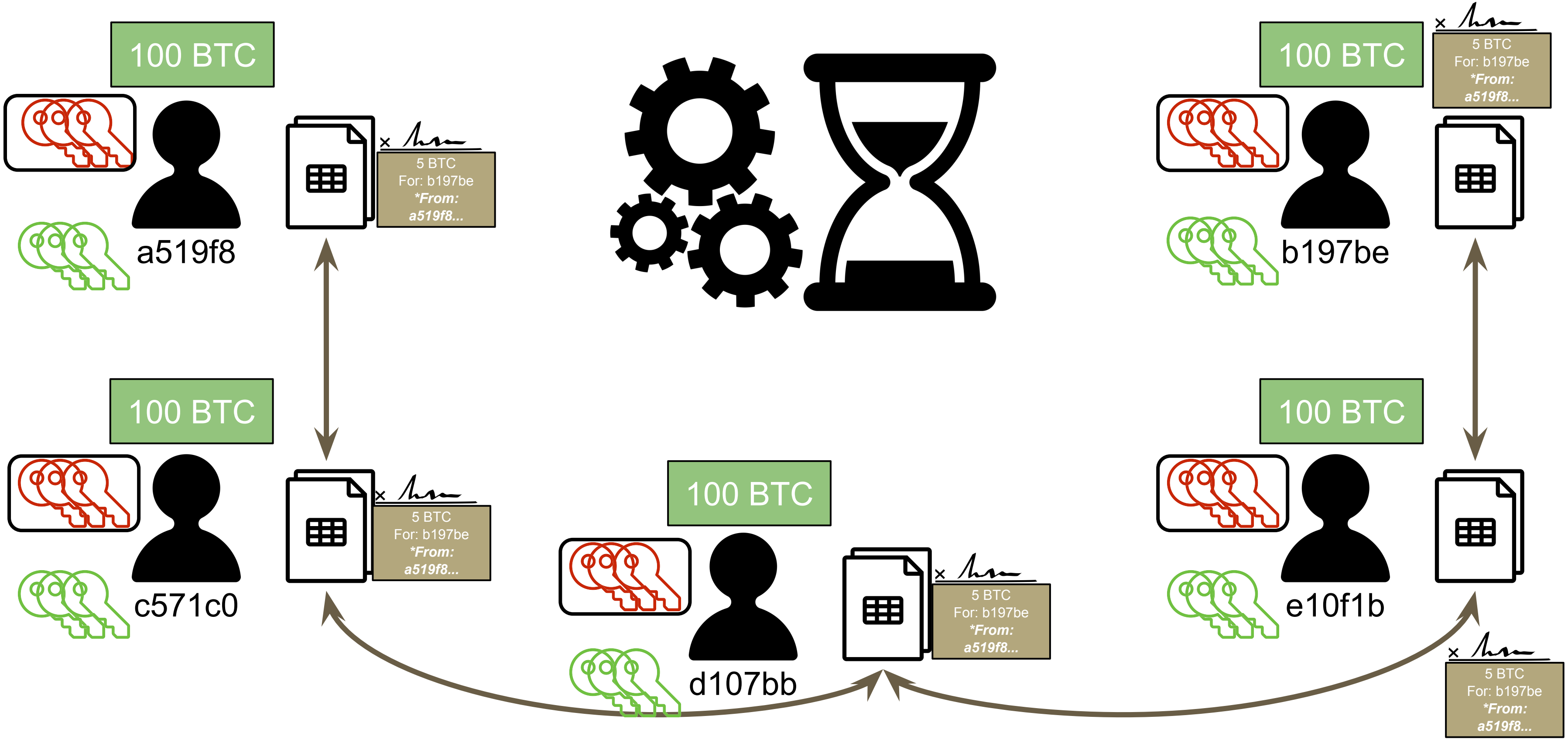
Bitcoin: wallets (or keychains)



Bitcoin: cryptographic puzzle



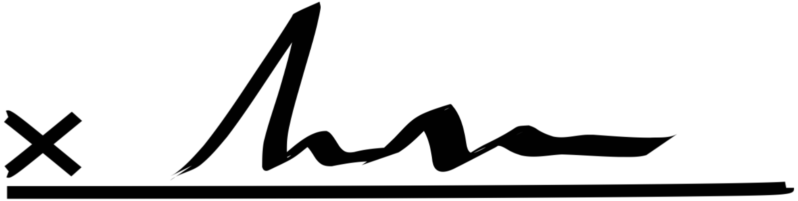
Bitcoin: "computational puzzle"



Sample attributes verified by nodes in each Tx:

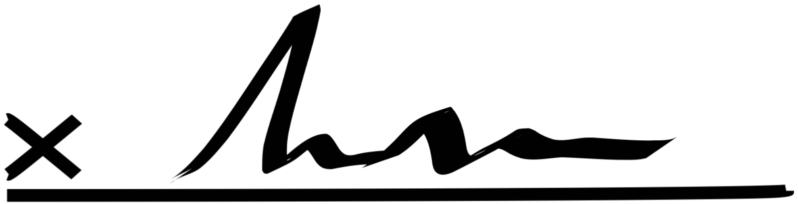
1. The transaction's syntax and data structure must be correct.
2. Neither lists of inputs or outputs are empty.
3. The transaction size in bytes is less than MAX_BLOCK_SIZE.
4. Each output value, as well as the total, must be within the allowed range of values (less than 21m coins, more than 0).
5. None of the inputs have hash=0, N=–1 (coinbase transactions should not be relayed).
6. nLockTime is less than or equal to INT_MAX.
7. The transaction size in bytes is greater than or equal to 100.
8. The number of signature operations contained in the transaction is less than the signature operation limit.
9. The unlocking script (scriptSig) can only push numbers on the stack, and the locking script (scriptPubkey) must match isStandard forms (this rejects "nonstandard" transactions).
10. A matching transaction in the pool, or in a block in the main branch, must exist.
11. For each input, if the referenced output exists in any other transaction in the pool, the transaction must be rejected.
12. For each input, look in the main branch and the transaction pool to find the referenced output transaction. If the output transaction is missing for any input, this will be an orphan transaction. Add to the orphan transactions pool, if a matching transaction is not already in the pool.
13. For each input, if the referenced output transaction is a coinbase output, it must have at least COINBASE_MATURITY (100) confirmations.
14. For each input, the referenced output must exist and cannot already be spent.
15. Using the referenced output transactions to get input values, check that each input value, as well as the sum, are in the allowed range of values (less than 21m coins, more than 0).
16. Reject if the sum of input values is less than sum of output values.
17. Reject if transaction fee would be too low to get into an empty block.
18. The unlocking scripts for each input must validate against the corresponding output locking scripts.

The Bitcoin "Puzzle"



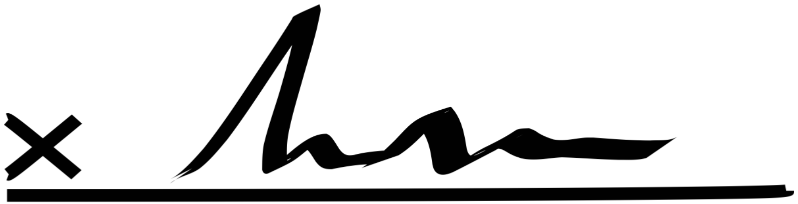
5 BTC
For: b197be
From: a519f8...

The Bitcoin "Puzzle"



Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8...

The Bitcoin "Puzzle"



Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…

Hash: ----------------------

The Bitcoin "Puzzle"

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce:
Nonce Solver:
Hash: ---------------------

The Bitcoin "Puzzle"

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce:
Nonce Solver:
Hash: ---------------------

Puzzle Solution → (Nonce)

Puzzle Solver (miner) → (Nonce Solver)

The Bitcoin "Puzzle"

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce:
Nonce Solver:
Hash: ---------------------

Puzzle Solution → Nonce

Puzzle Solver → Nonce Solver

Must be below certain value, say one leading zero → Hash

The Bitcoin "Puzzle": example of how miners mine

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 0
Nonce Solver: a519f8 (Alice)
Hash: a166137346cd32e73e

Puzzle Solution → *From: a519f8…

Puzzle Solver → Nonce Solver: a519f8 (Alice)

Must be below certain value → Hash: a166137346cd32e73e

The Bitcoin “Puzzle”

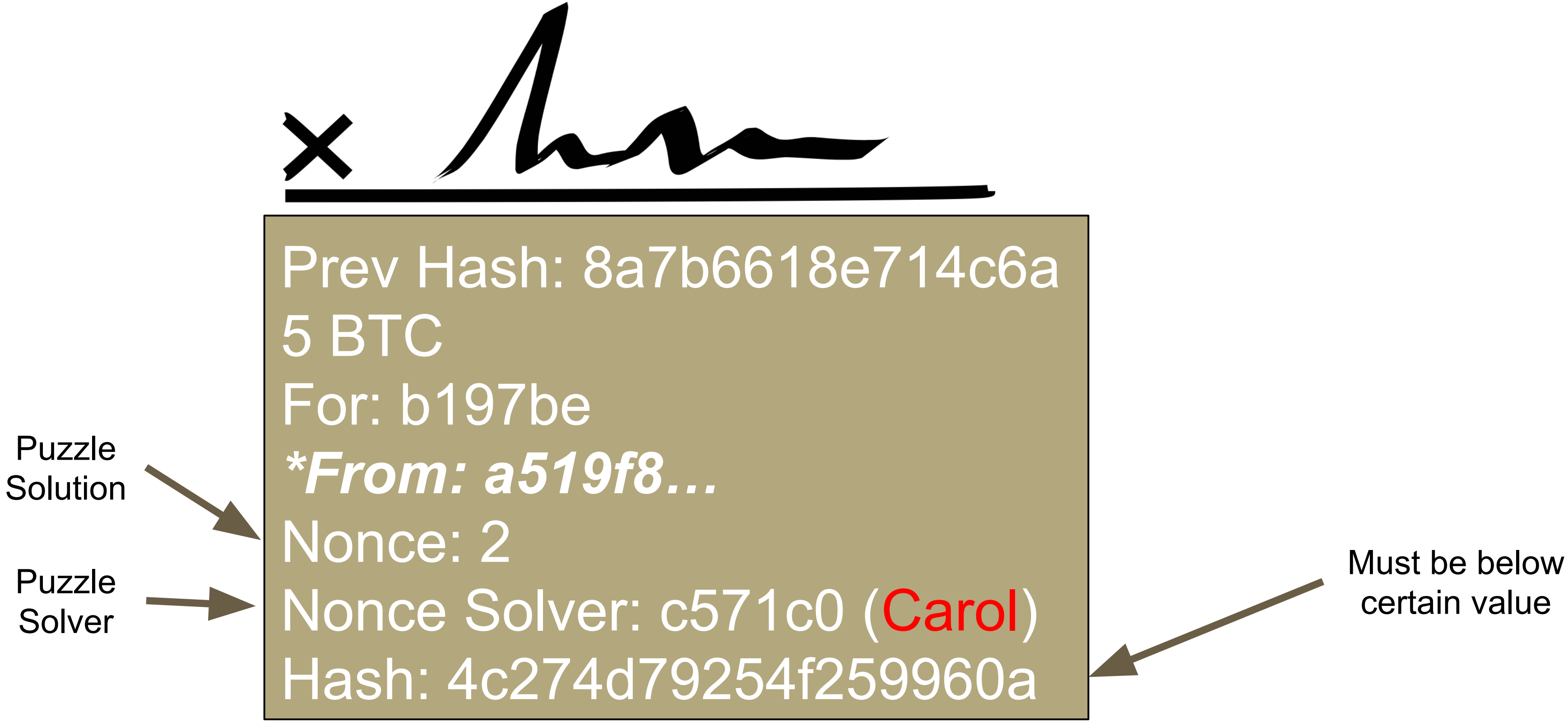
Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 1
Nonce Solver: b197be (Bob)
Hash: d59910db074b35fa9d

Puzzle Solution → *From: a519f8…

Puzzle Solver → Nonce Solver

Must be below certain value → Hash

The Bitcoin "Puzzle"



The Bitcoin "Puzzle"

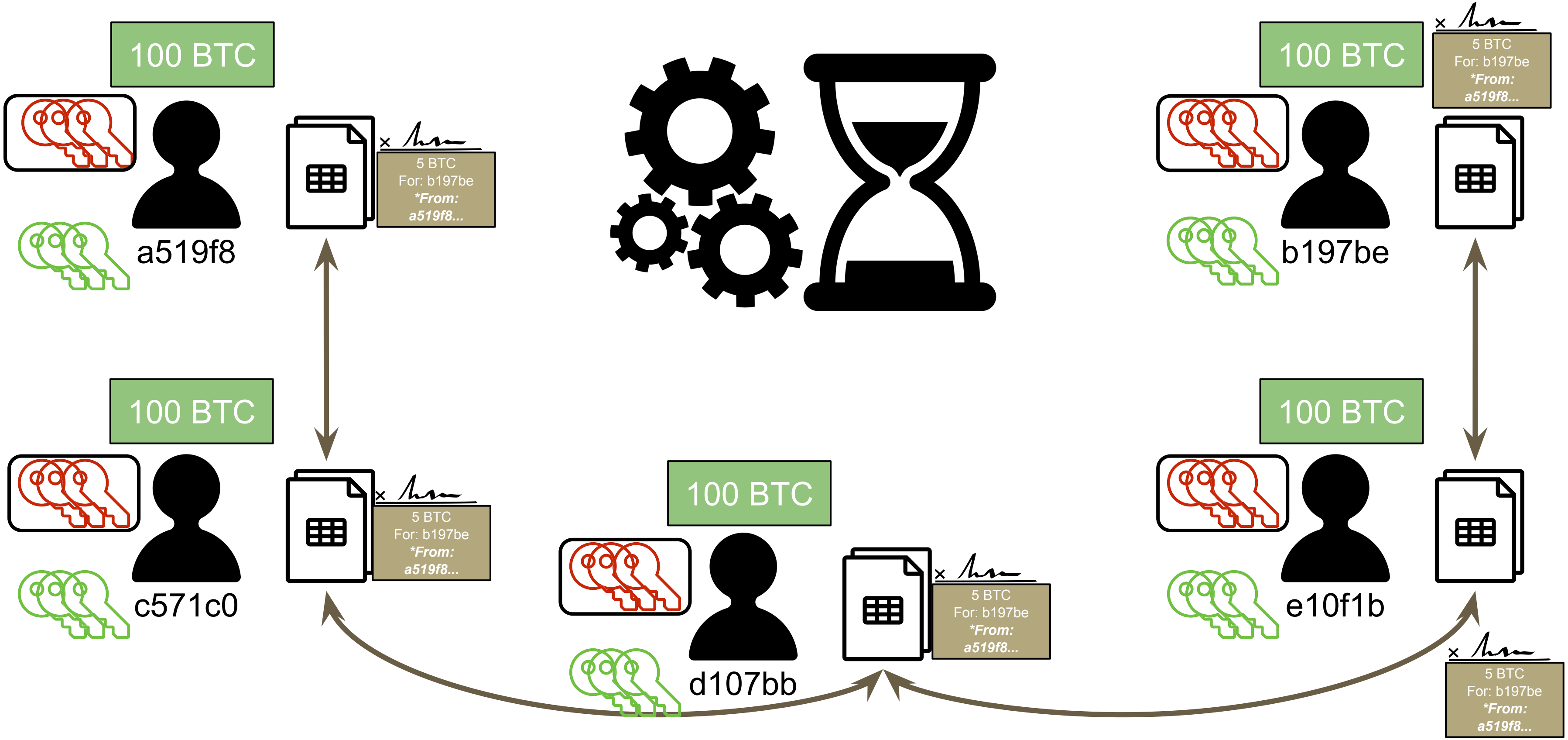
Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **03a419ef573a846f**

Puzzle Solution → *From: a519f8…

Puzzle Solver → Nonce Solver: d107bb (Dave)

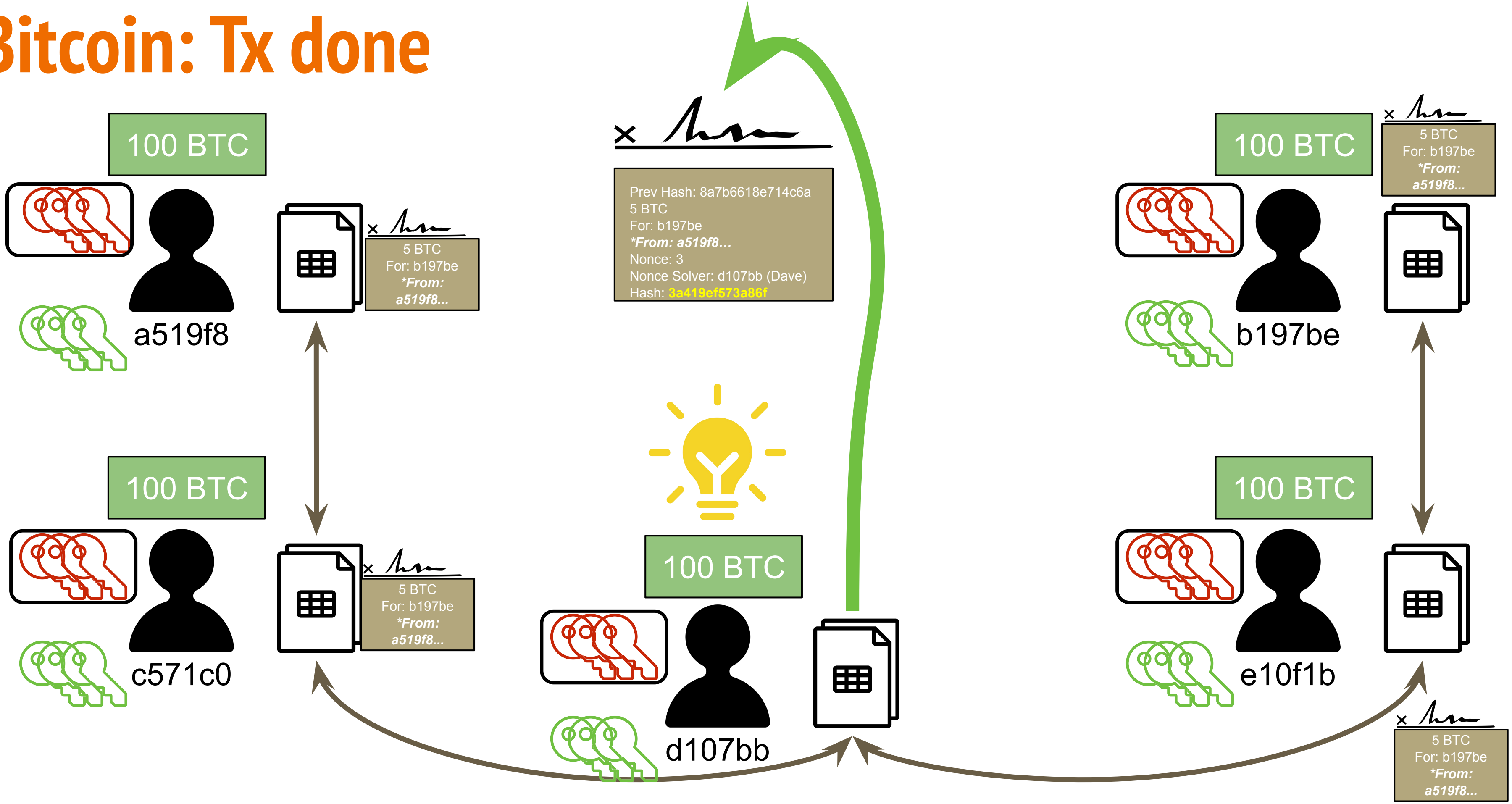
Must be below certain value → Hash: 03a419ef573a846f

Bitcoin

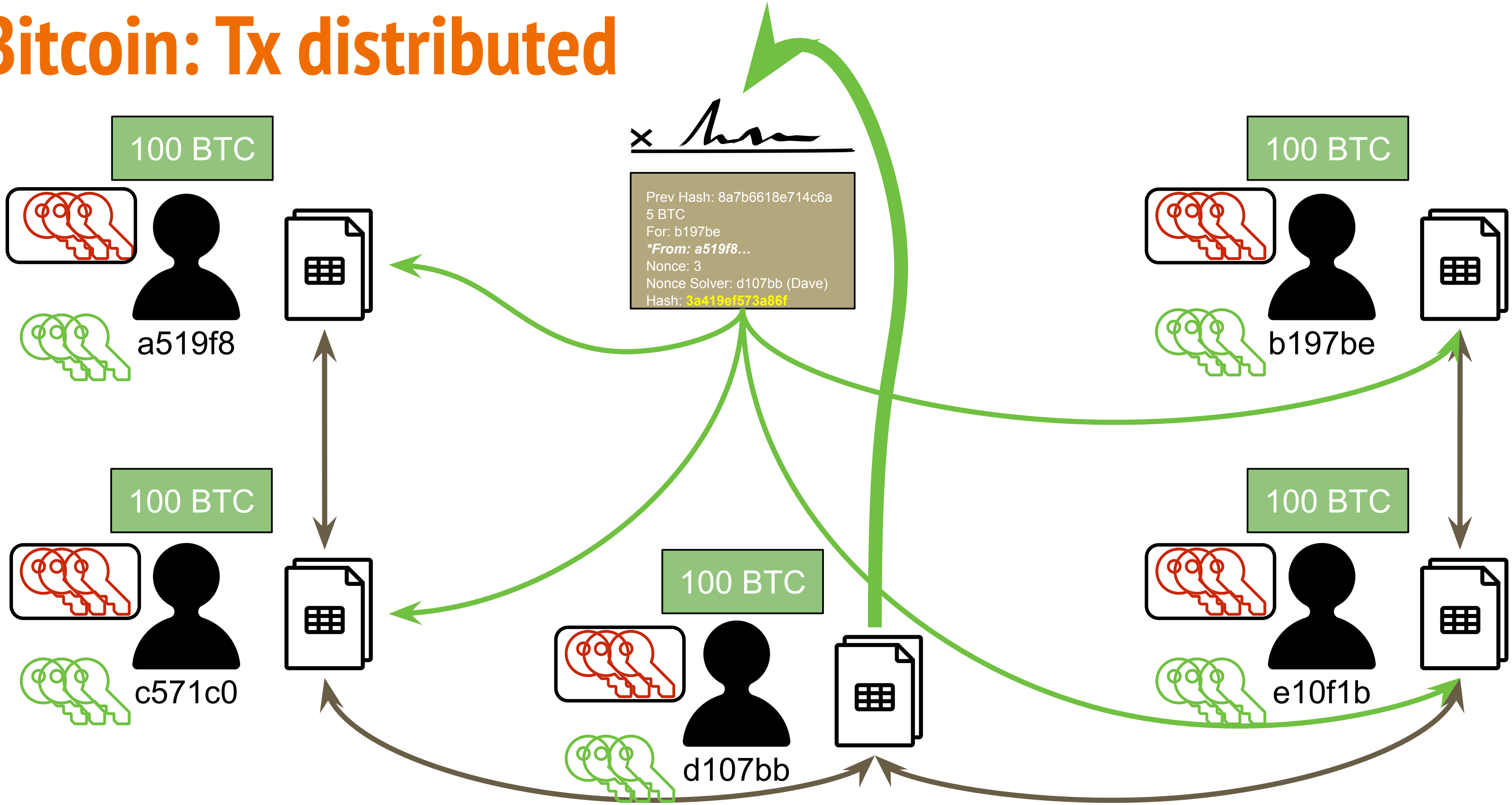


100 BTC
a519f8
5 BTC
For: b197be
*From:
a519f8...
100 BTC
b197be
5 BTC
For: b197be
*From:
a519f8...
100 BTC
c571c0
5 BTC
For: b197be
*From:
a519f8...
100 BTC
d107bb
5 BTC
For: b197be
*From:
a519f8...
100 BTC
e10f1b
5 BTC
For: b197be
*From:
a519f8...

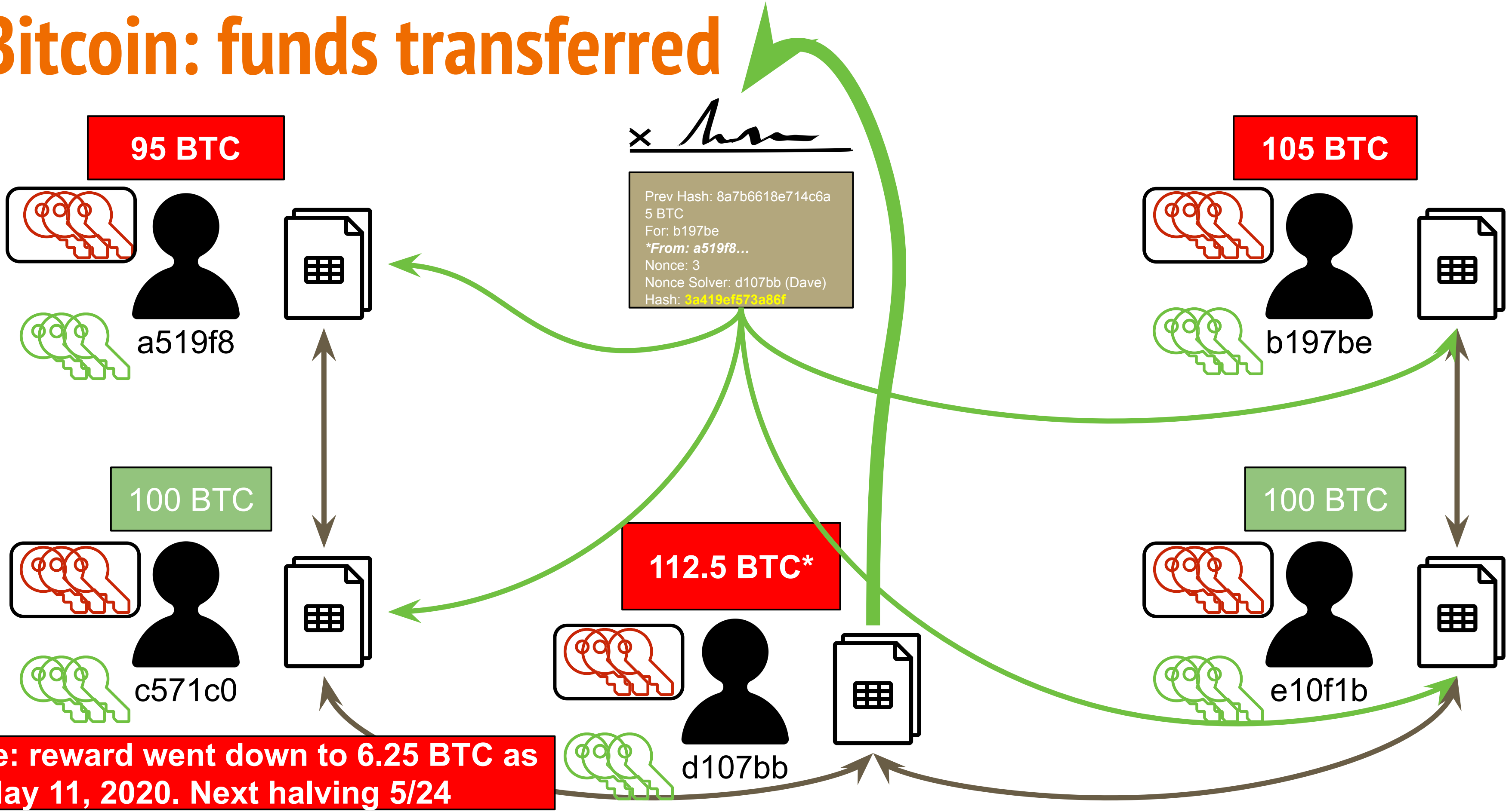
Bitcoin: Tx done



Bitcoin: Tx distributed



Bitcoin: funds transferred



Note: reward went down to 6.25 BTC as of May 11, 2020. Next halving 5/24

The Bitcoin "Puzzle": can you steal the nonce?

Puzzle Solution - nonce depends on solver too

Puzzle Solver

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **03a419ef573a86f**

Must be below certain value

The Bitcoin “Puzzle”: nonce is block-specific

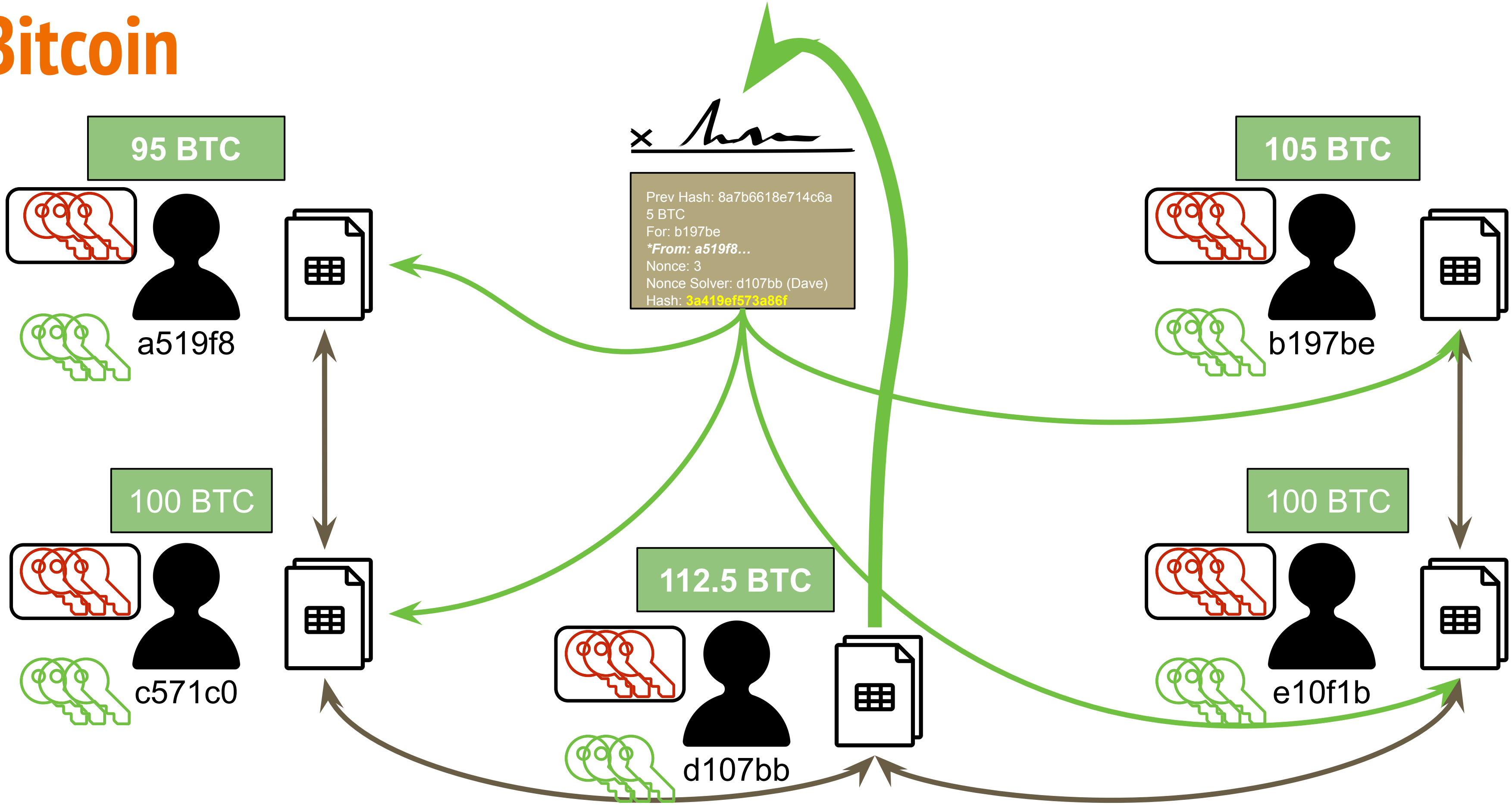
Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **03a419ef573a86f**

Puzzle Solution → From

Puzzle Solver → Nonce Solver

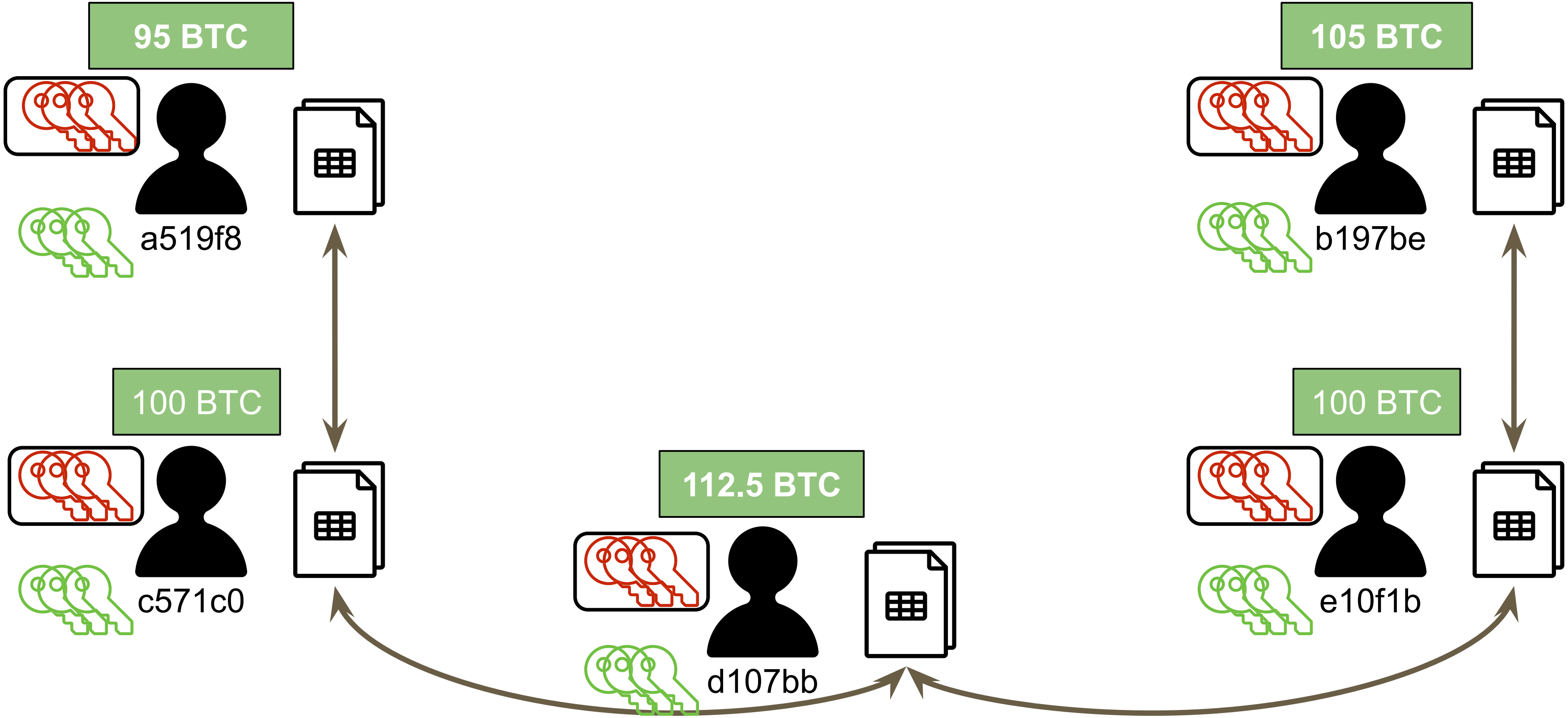
Must be below certain value → Hash

Bitcoin

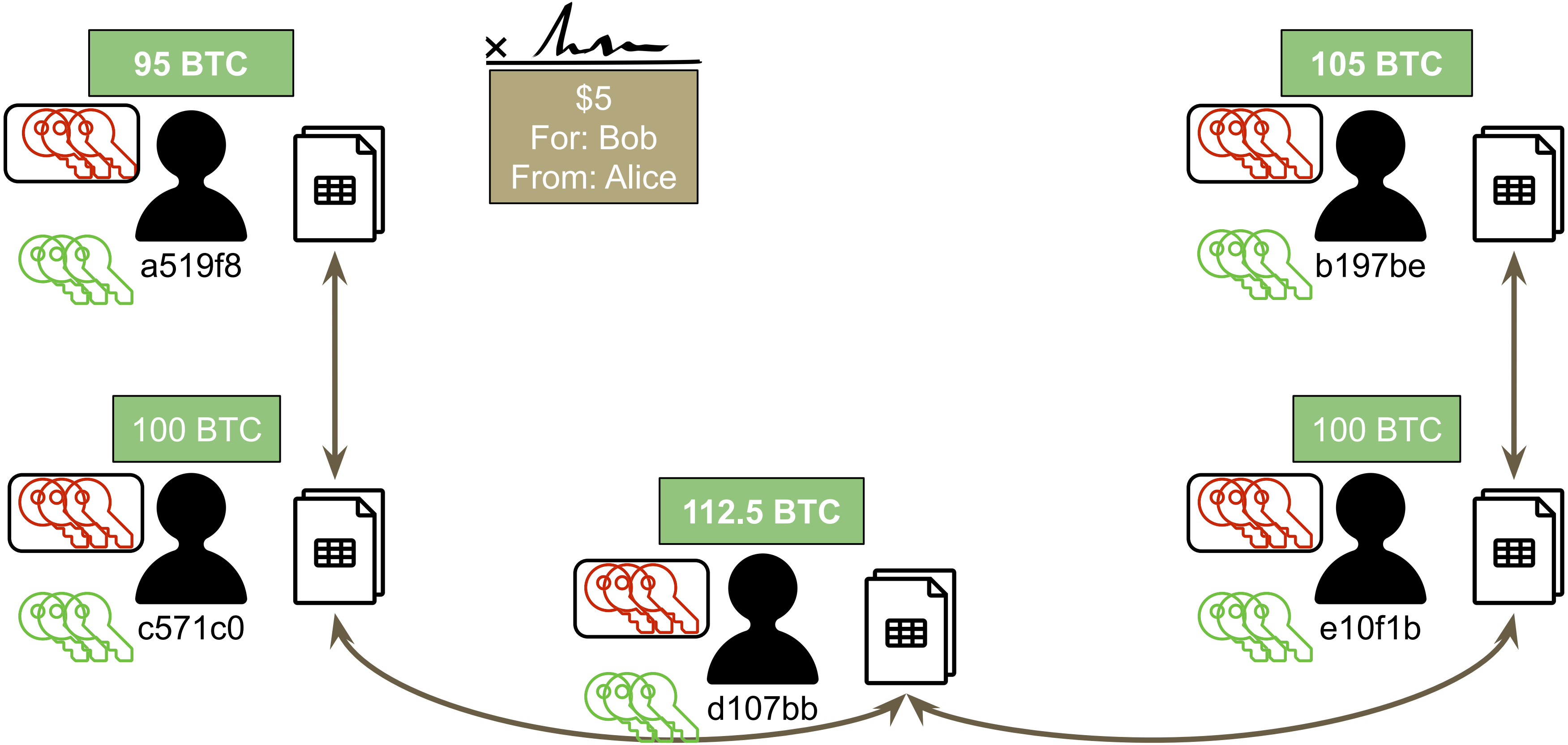


95 BTC
a519f8
Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
*From: a519f8...
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: 3a419ef573a86f
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

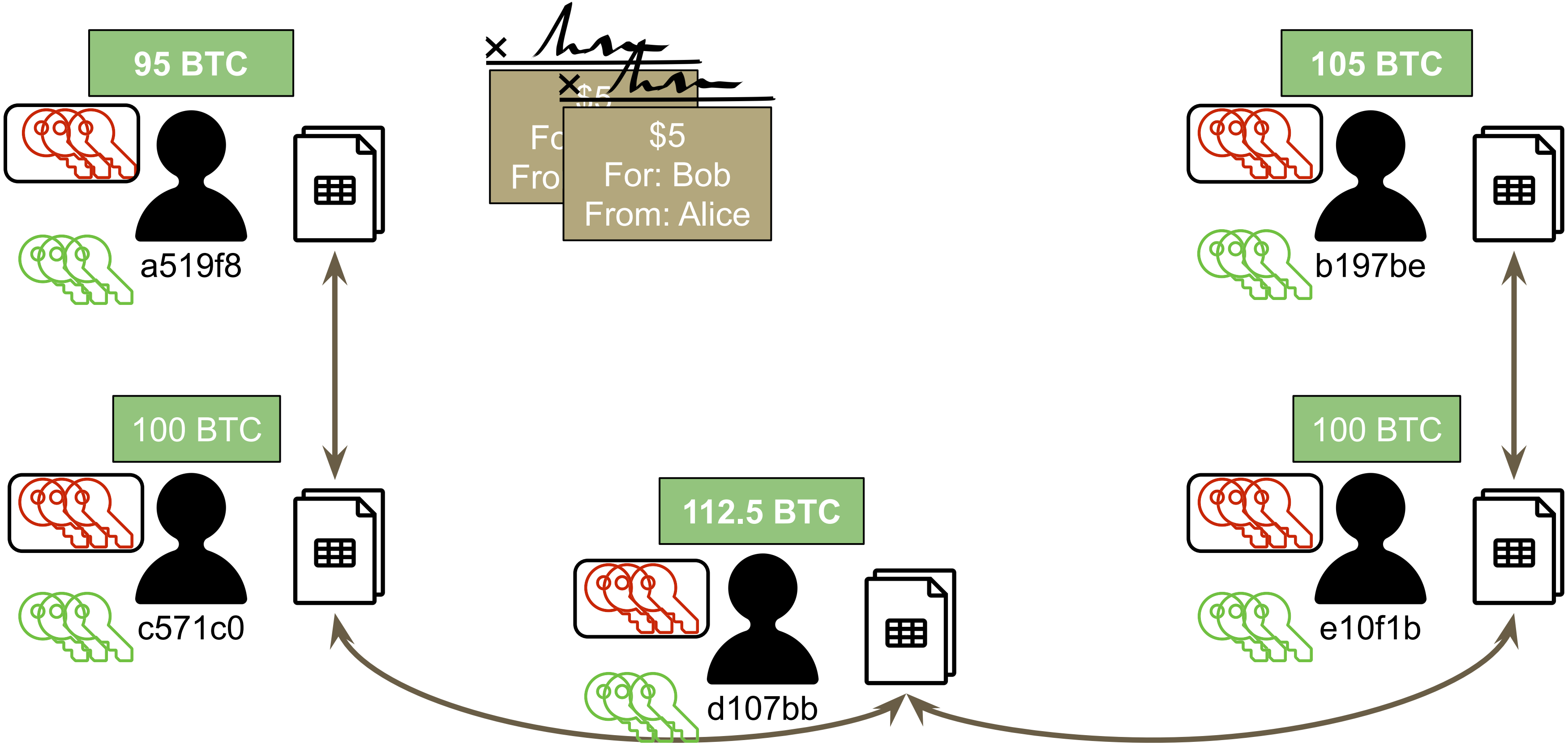
Bitcoin: one Tx per block? Not really!



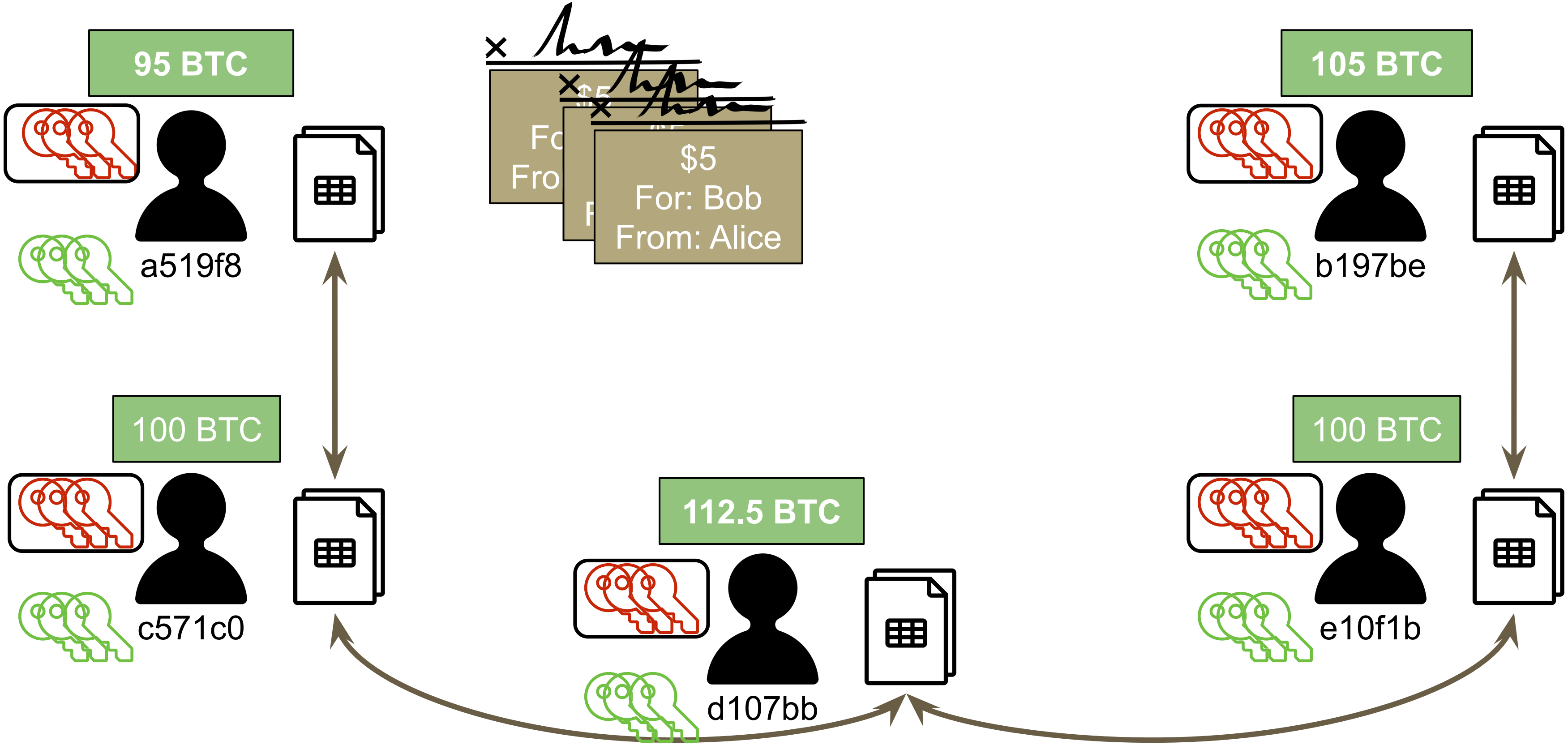
Bitcoin


95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin

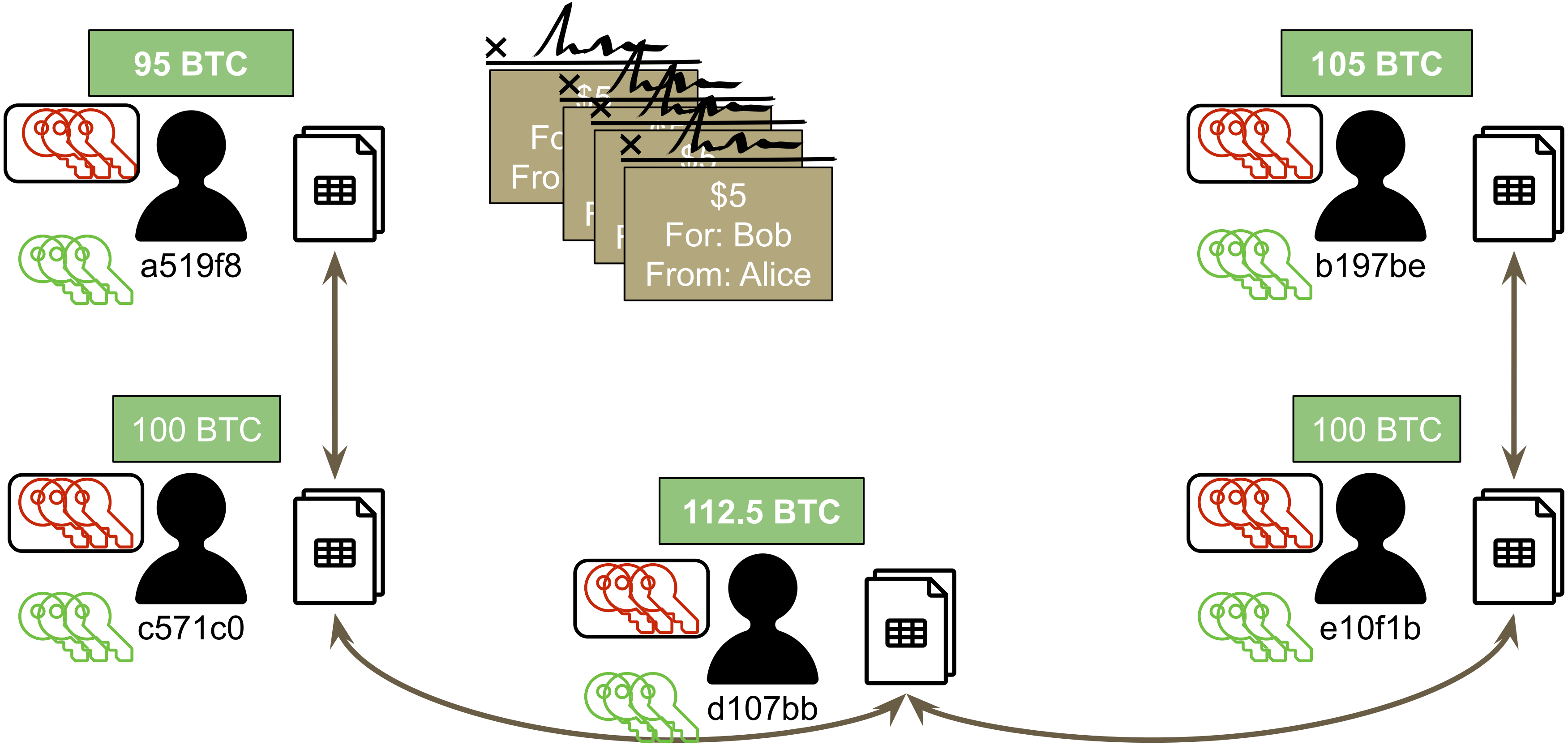

95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin

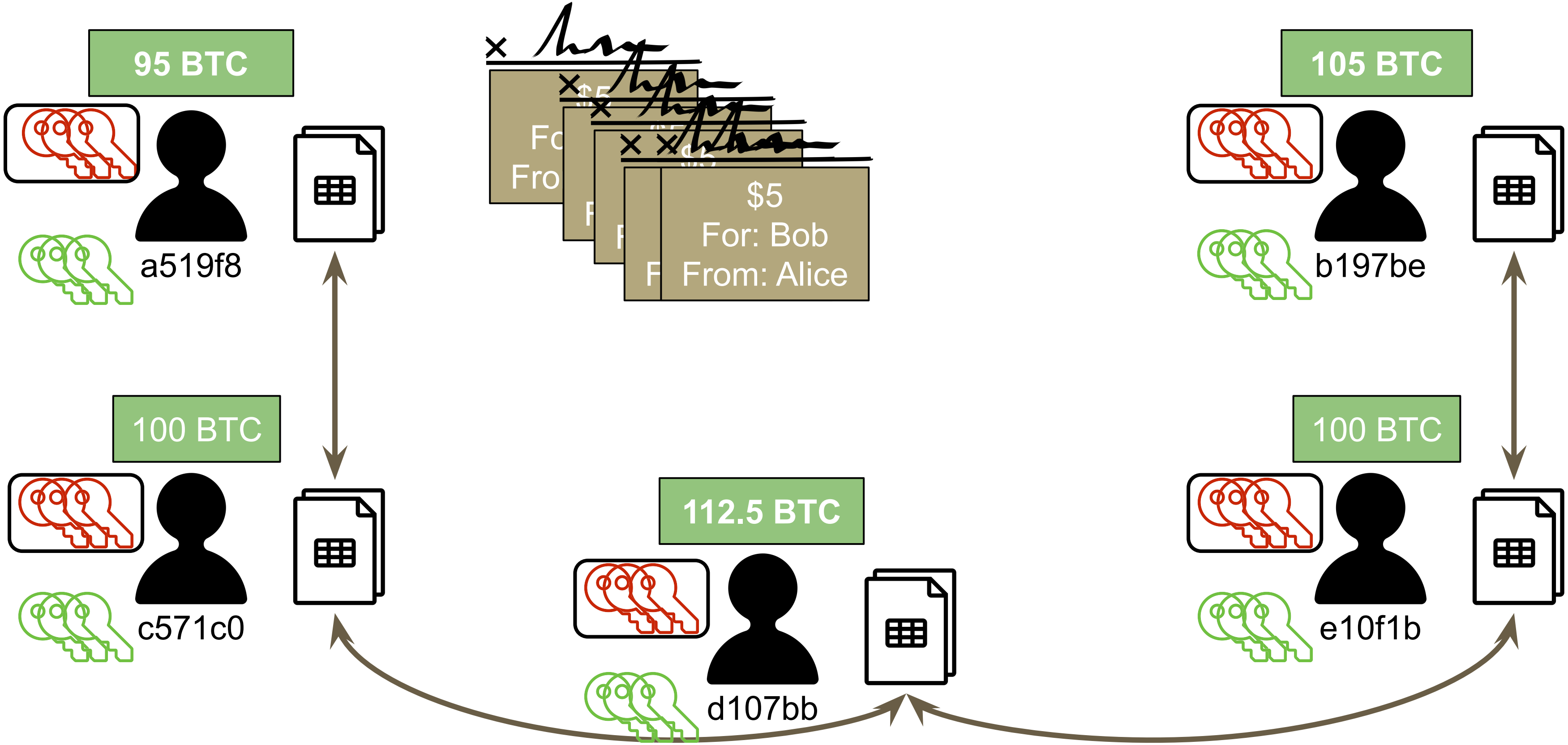


95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin

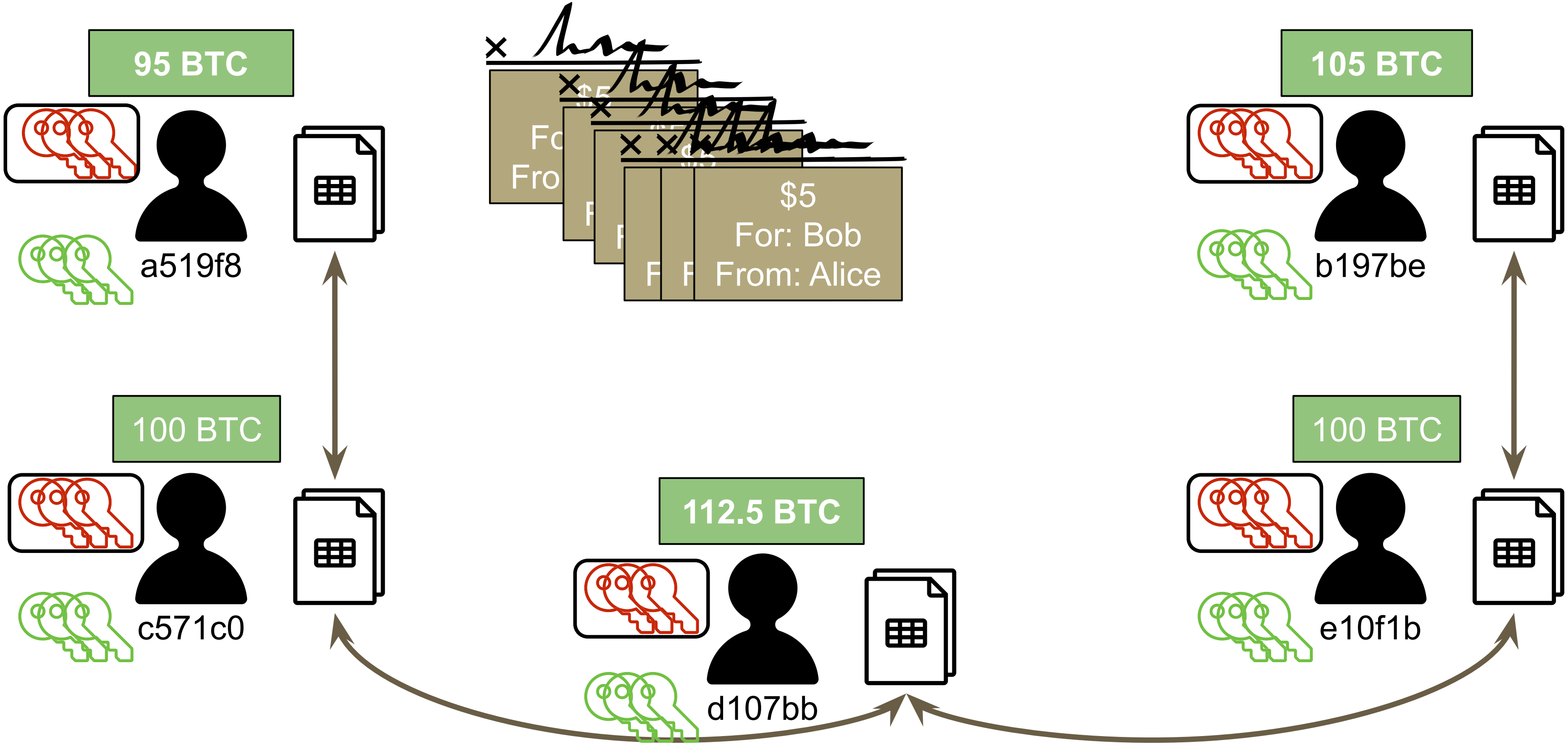

95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin



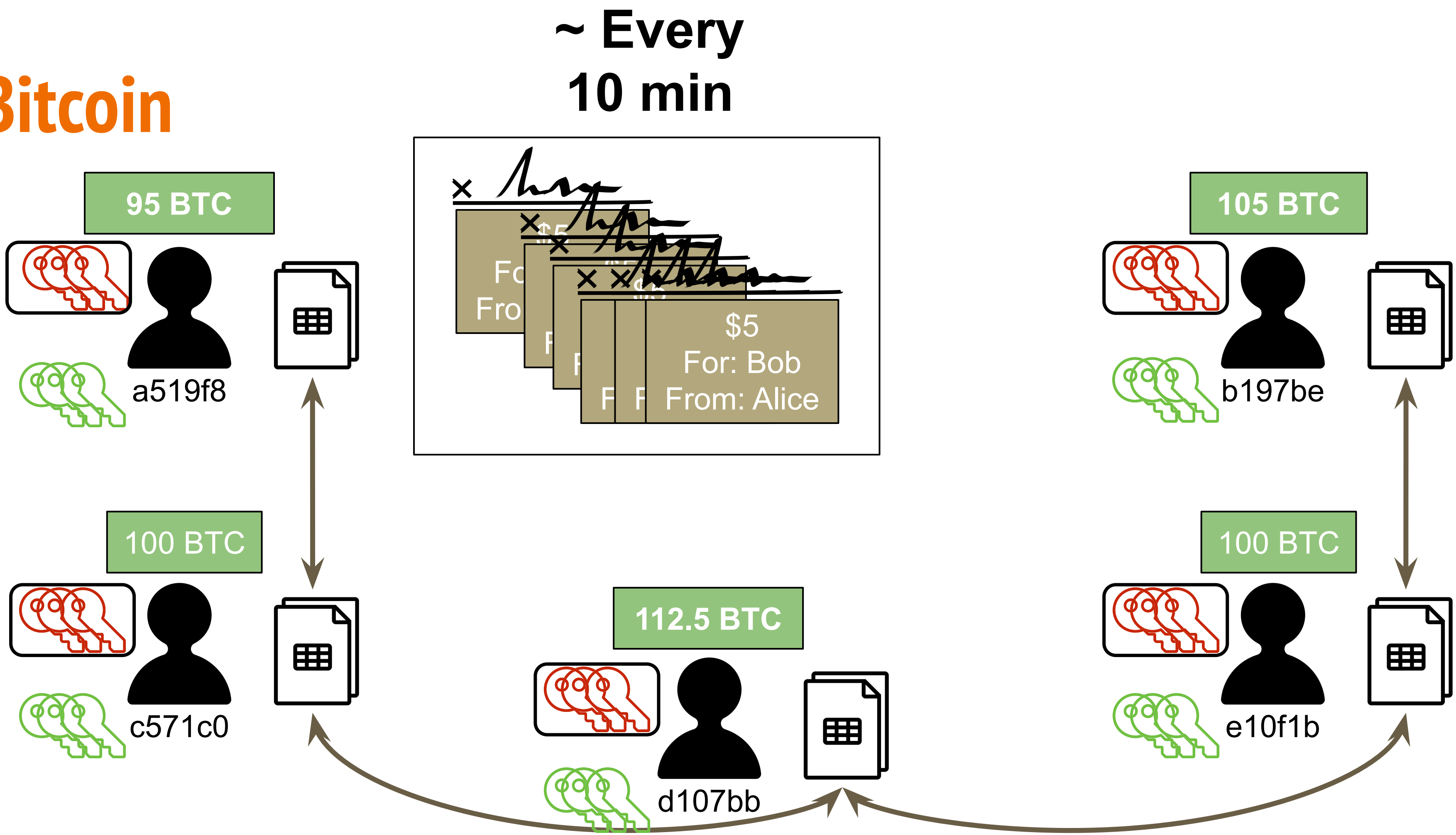
95 BTC
a519f8
105 BTC
b197be
$5
For: Bob
From: Alice
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

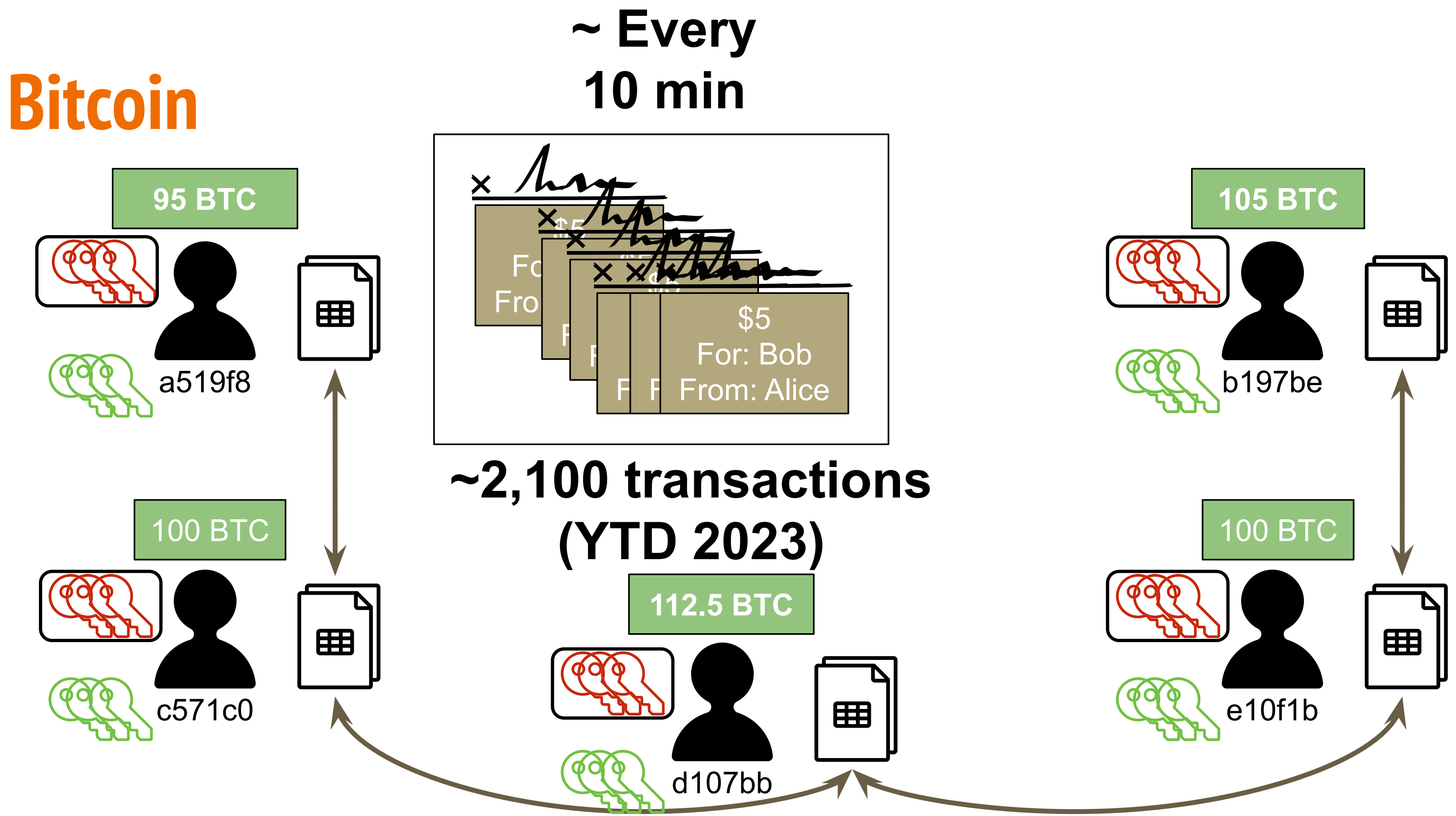
Bitcoin



95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin


~ Every
10 min
$5
For: Bob
From: Alice
95 BTC
a519f8
105 BTC
b197be
100 BTC
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b


Bitcoin
~ Every
10 min
$5
For: Bob
From: Alice
~2,100 transactions
(YTD 2023)
95 BTC
a519f8
100 BTC
c571c0
112.5 BTC
d107bb
105 BTC
b197be
100 BTC
e10f1b

Calibrating The Bitcoin “Puzzle” w/ “Difficulty”

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **3a419ef573a86f**

Puzzle Solution → *From: a519f8…

Puzzle Solver → Nonce Solver

Must be below certain value → Hash

Calibrating The Bitcoin "Puzzle" w/ Difficulty

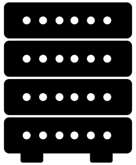
Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **3a419ef573a86f**

Puzzle Solution → Nonce: 3

Puzzle Solver → Nonce Solver: d107bb (Dave)

Must be below certain value **(DIFFICULTY)** → Hash

Calibrating The Bitcoin “Puzzle” w/ Difficulty



Few computers = low difficulty, i.e. blocks can be solved more easily

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **000a419ef573a86f**

Puzzle Solution

Puzzle Solver

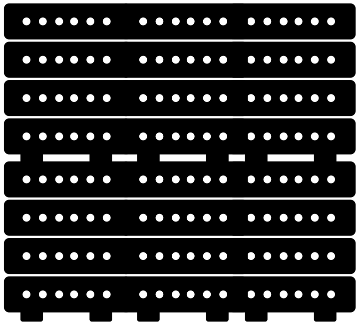
Must be below certain value **(DIFFICULTY)**

Calibrating The Bitcoin "Puzzle"...

Prev Hash: 8a7b6618e714c6a
5 BTC
For: b197be
From: a519f8…
Nonce: 3
Nonce Solver: d107bb (Dave)
Hash: **000000a419ef573a**

Puzzle Solution

Puzzle Solver



More computers = high difficulty, i.e. blocks more time-consuming to solve, but balances out because more computers working to solve the problem

Must be below certain value **(DIFFICULTY)**

Blockchain, Cryptocurrencies & Digital Tokens Demystified

Fall 2023 (EMBA)
Columbia Business School

Welcome Back to Session 3 🎉

Curriculum Roadmap

	Nov 4	Nov 18	Dec 2	Dec 9
Morning	Networks & Protocols	Hashing, Hashing Tables & One- Way Functions & a few more tech	Bitcoin + other forms of crypto payments and store of value mechanisms and media	DeFi & Other Applications (Digital Tokens, CBDC, etc.) + Speaker: Future of Finance + Discussion Forum
	Lunch	Lunch	Lunch	Lunch
Afternoon	Encryption & Cryptography (plus some math!)	**Bring it All Together:** Let's build a blockchain & discuss variety of cases	Ethereum & Other Digital Tokens + Speaker: Regulatory & Legal Considerations in Blockchain & Digital Assets	Governance, Marketplaces, NFTs & More; Final Lecture on How the Future May Play Out + Final Presentations

Class Schedule - Nov 4, Nov 18, Dec 2, Dec 9

Class Plan

Nov 4	08:30 am to 6:45 pm (K-440)	Module 1 + 2
Nov 18	08:30 am to 6:45 pm (K-440)	Module 3 + 4
Dec 2	08:30 am to 6:45 pm (K-440)	Midterm Project + 5 & 6 + Guest Speaker
Dec 9	08:30 am to 6:45 pm (K-440)	Module 7 & 8 + Guest Speaker + final presentations

Daily Schedule

8:30-9:45 am	**Lecture**
9:45-10:00 am	*Break*
10:00-11:15 pm	**Lecture**
11:15 am-12:30 pm	Lunch (1h15min) - Kravis 2nd floor (Smith Dining)
12:30-2:00 pm	**Lecture**
2:00-2:15 pm	*Break*
2:15-3:30 pm	**Lecture**
3:30-3:45 pm	*Break*
3:45-5:00 pm	**Lecture**
5:00-5:15 pm	*Break*
5:15-6:45 pm	**Lecture**

Important Admin Items for the Day

- Note last class is on Dec 9 (next week, not in two weeks)
- Final projects assigned already
- Details on your final projects (presentation & paper)
- Final presentations next week
- Final papers due on Dec 18
- Thoughts on “Blockchain Killer App” for today and/or 4
- Discussion Forum next class
- Watch lecture recordings and email me for office hours

THE MOST Important Admin Item for the Day

THE MOST Important Admin Item for the Day

Catering today is by **Dig Inn**:

- Brown Rice
- Lemon & Herb Farro
- Maple Glazed Crispy Tofu
- Herb Roasted Chicken
- Beef & Chicken Meatballs
- Wild Salmon
- Broccoli
- Brussels Sprouts
- Sweet Potatoes

THE MOST Important Admin Item for the Day

Catering today is by **Dig Inn**:

- Brown Rice
- Lemon & Herb **Farro**
- Maple Glazed Crispy Tofu
- Herb Roasted Chicken
- Beef & Chicken Meatballs
- Wild Salmon
- Broccoli
- Brussels Sprouts
- Sweet Potatoes

THE MOST Important Admin Item for the Day

Catering today is by **Dig Inn**:

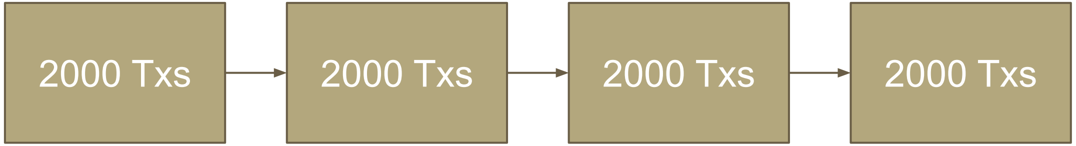
- Brown Rice
- Lemon & Herb **Farro (Farrokhnia!!)**
- Maple Glazed Crispy Tofu
- Herb Roasted Chicken
- Beef & Chicken Meatballs
- Wild Salmon
- Broccoli
- Brussels Sprouts
- Sweet Potatoes

Before we begin, any interesting points or lessons from our prior session you'd like to share?

Let's start our Session 3

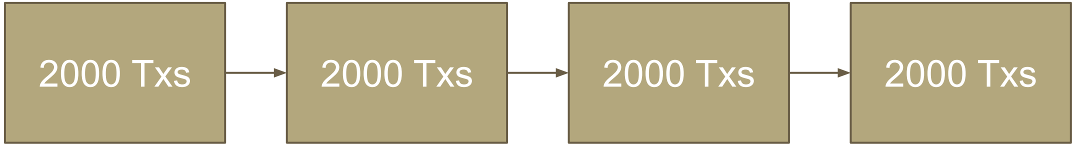
Why the Puzzle?

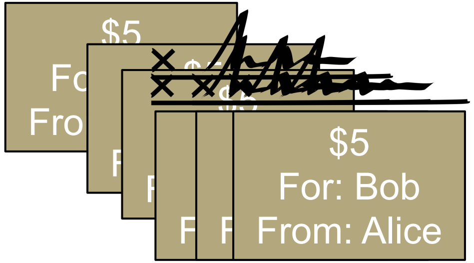
Normal Miner’s Blockchain:



Why the Puzzle?

Normal Miner's Blockchain:

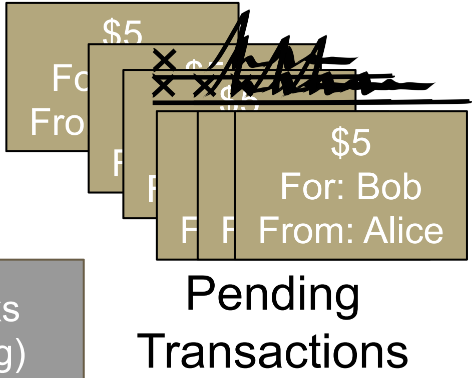




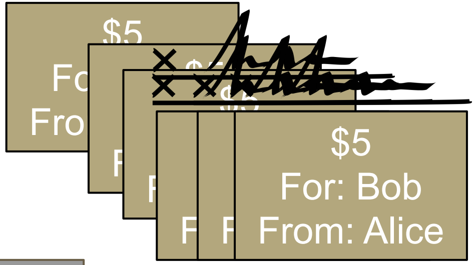
Pending Transactions

Why the Puzzle?

Normal Miner’s Blockchain:



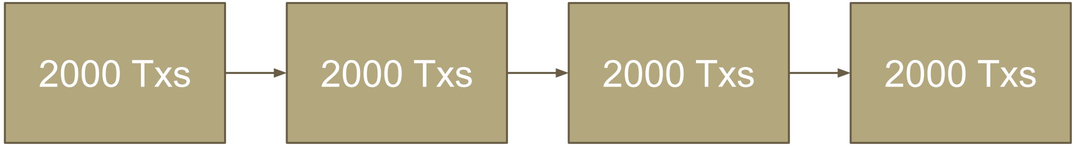
Why the Puzzle?



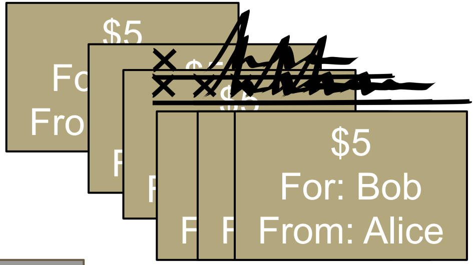
Pending Transactions

Normal Miner’s Blockchain:

Malicious Miner’s Blockchain:



Why the Puzzle? Let's spam!



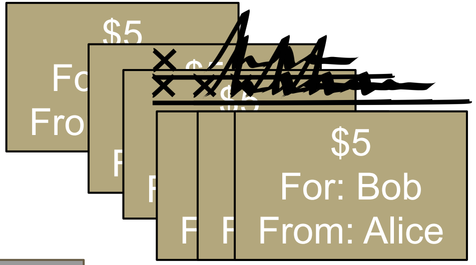
Pending Transactions

Normal Miner's Blockchain:

Malicious Miner's Blockchain:

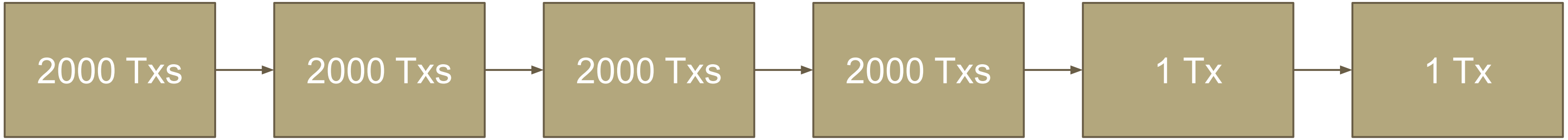
Why the Puzzle?

Normal Miner's Blockchain:

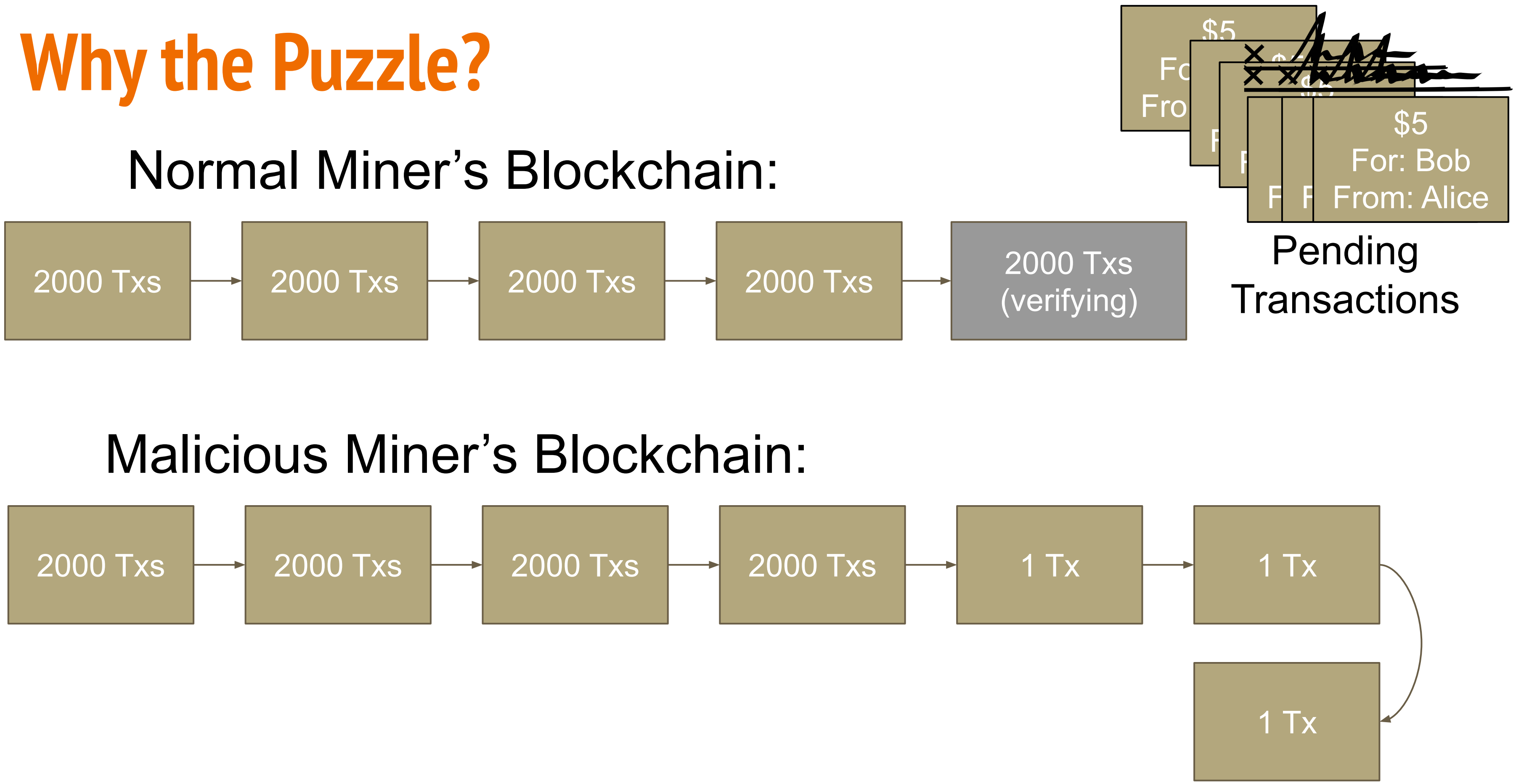


Pending Transactions

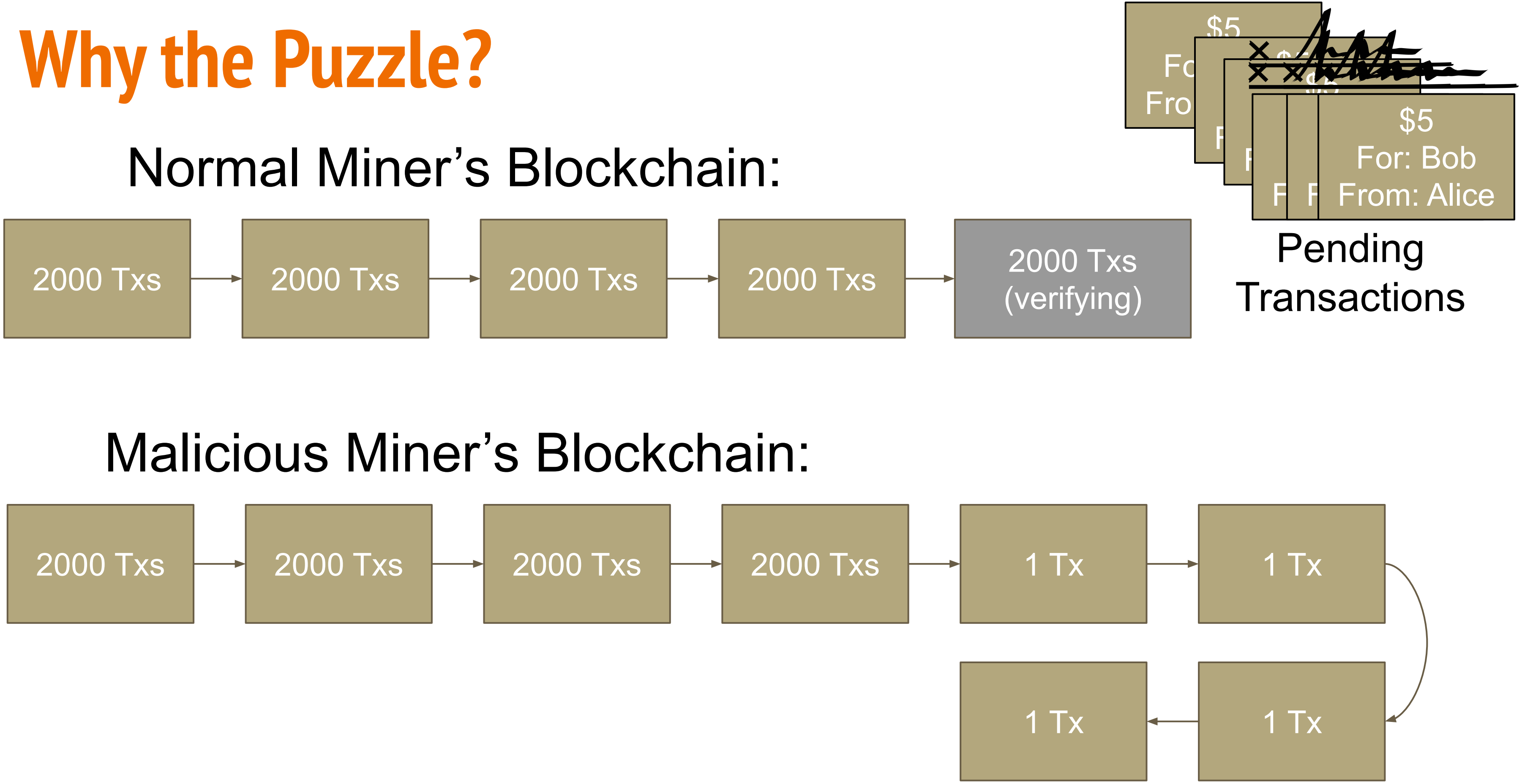
Malicious Miner's Blockchain:



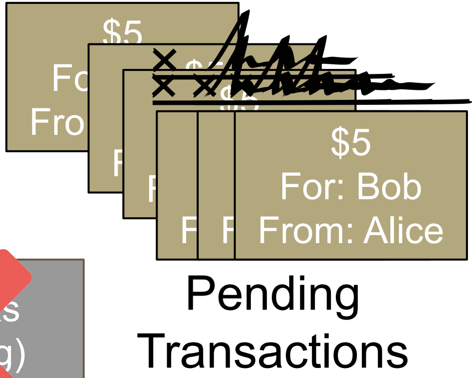
Why the Puzzle?



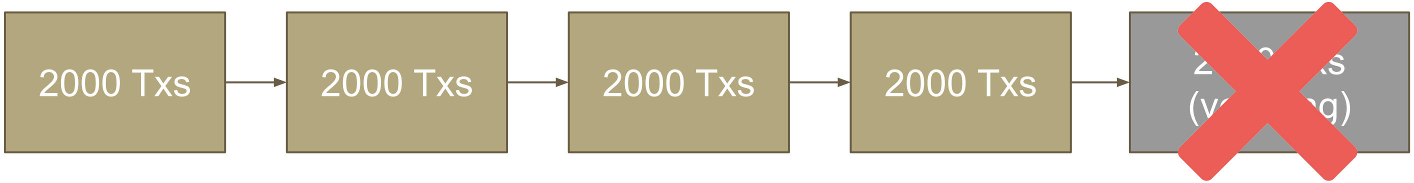
Why the Puzzle?



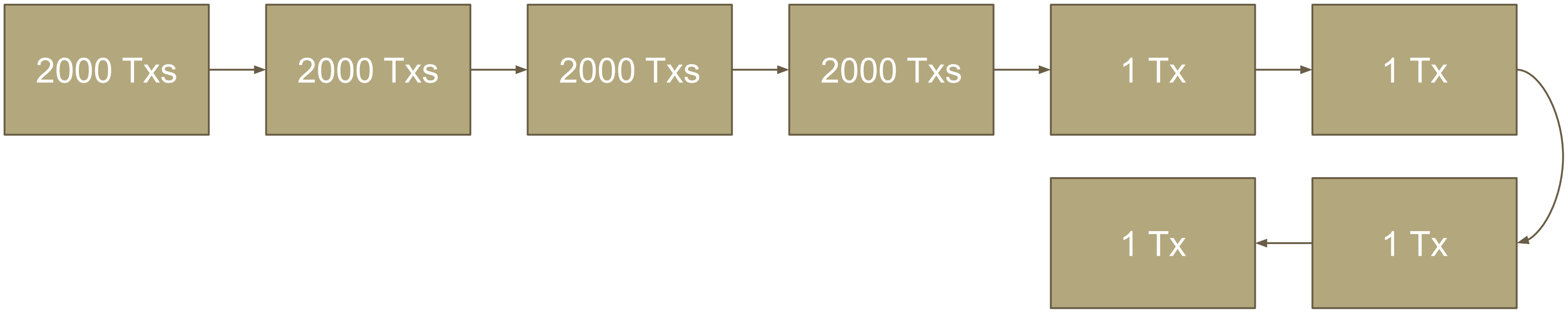
Why the Puzzle?



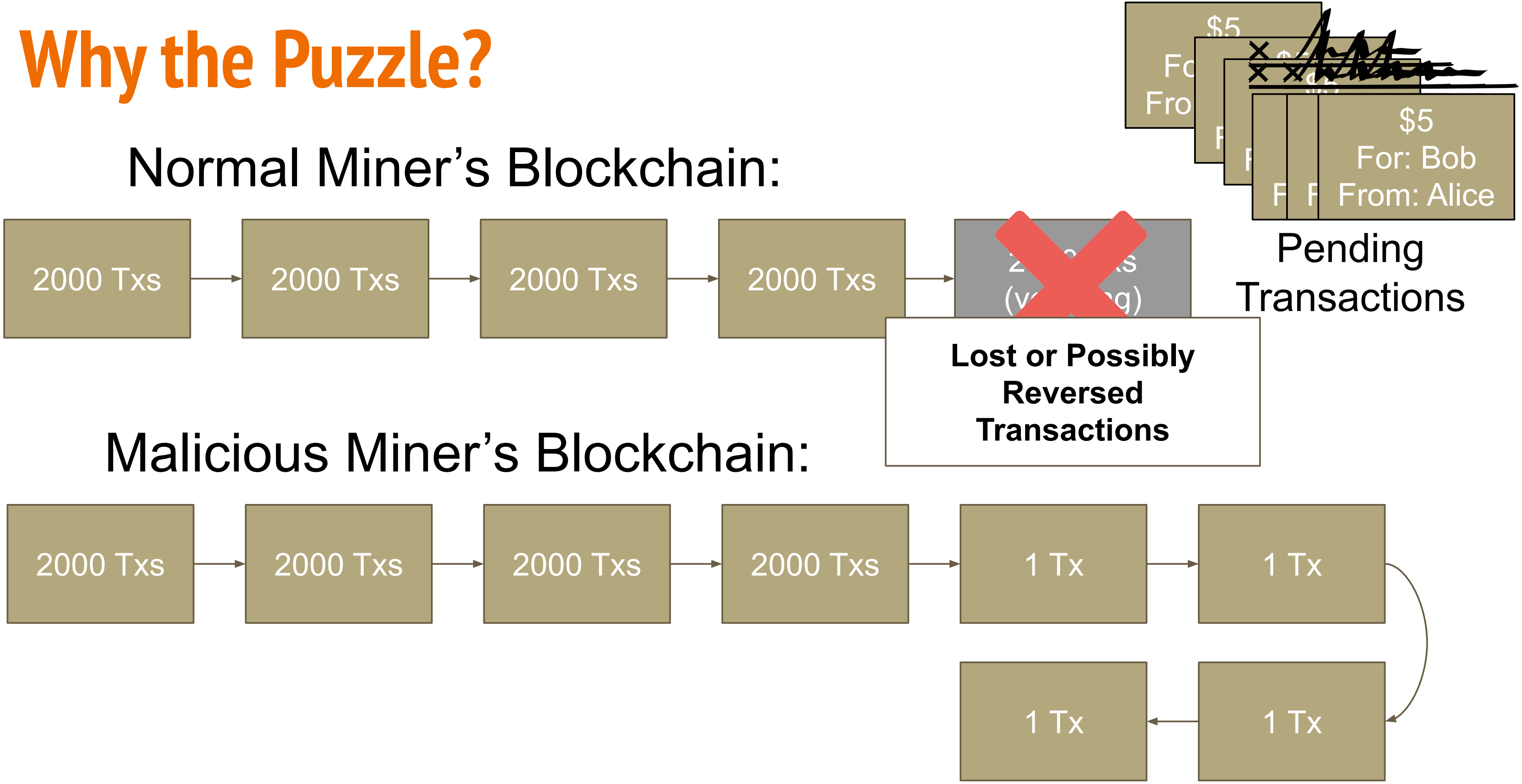
Normal Miner's Blockchain:



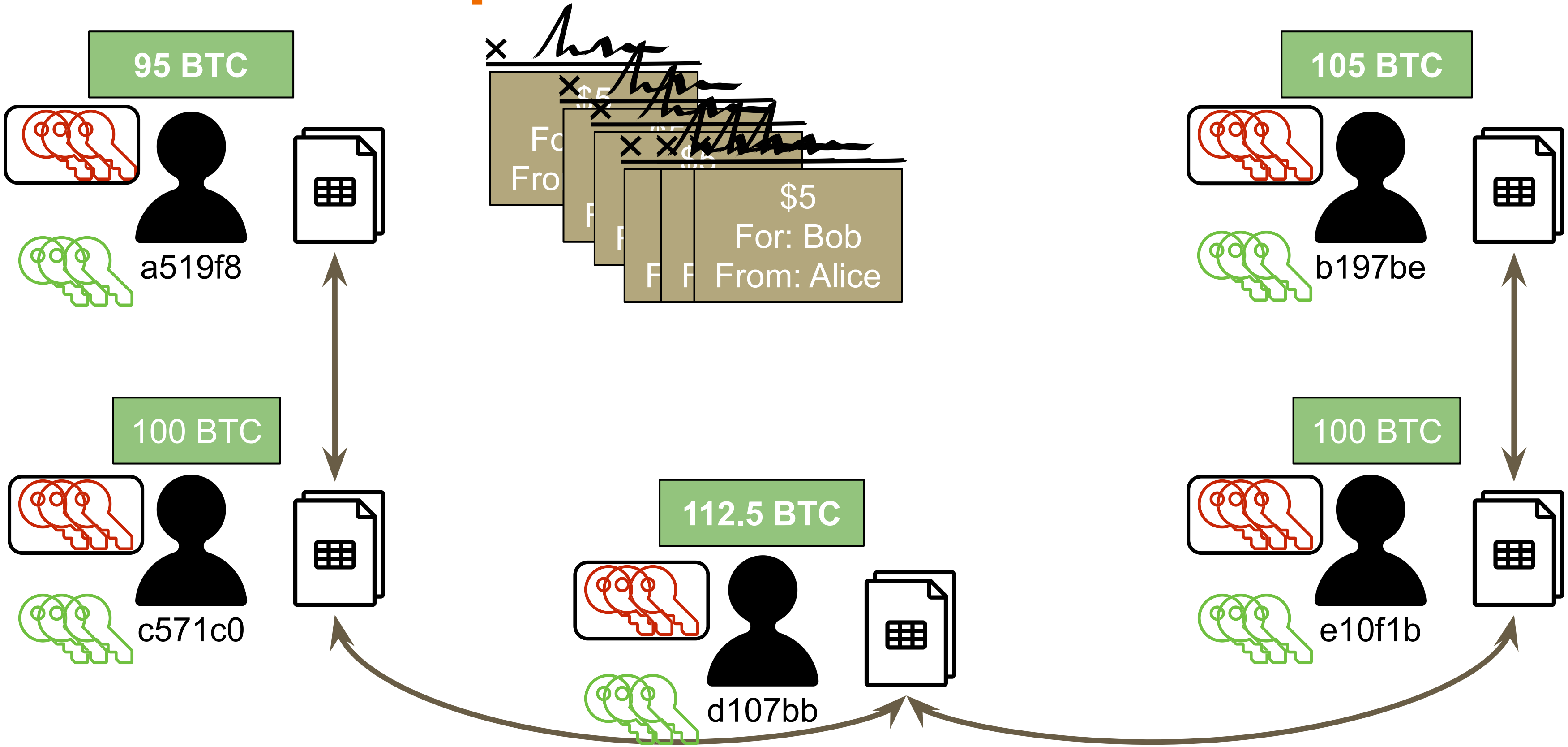
Malicious Miner's Blockchain:



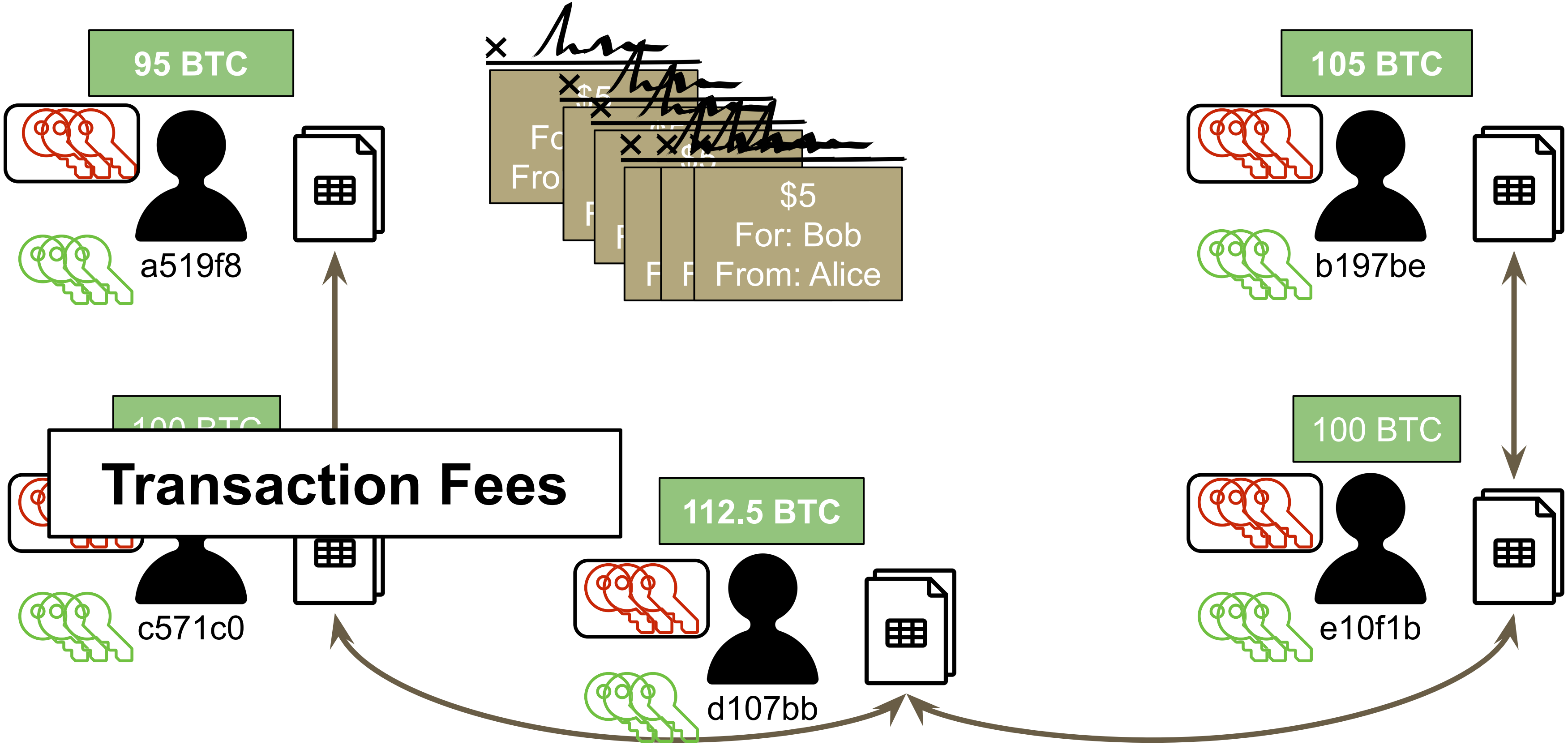
Why the Puzzle?



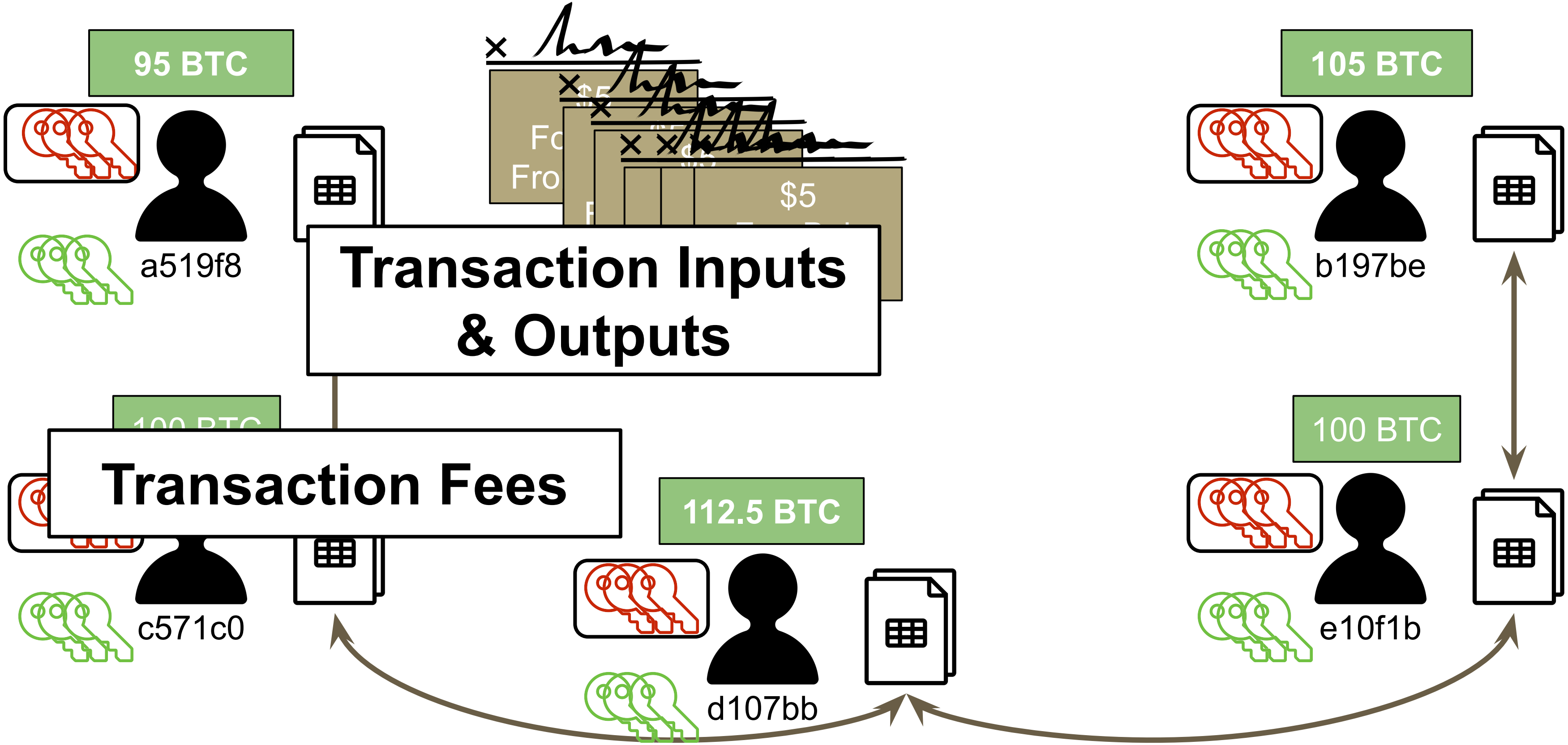
Bitcoin: other topics



Bitcoin

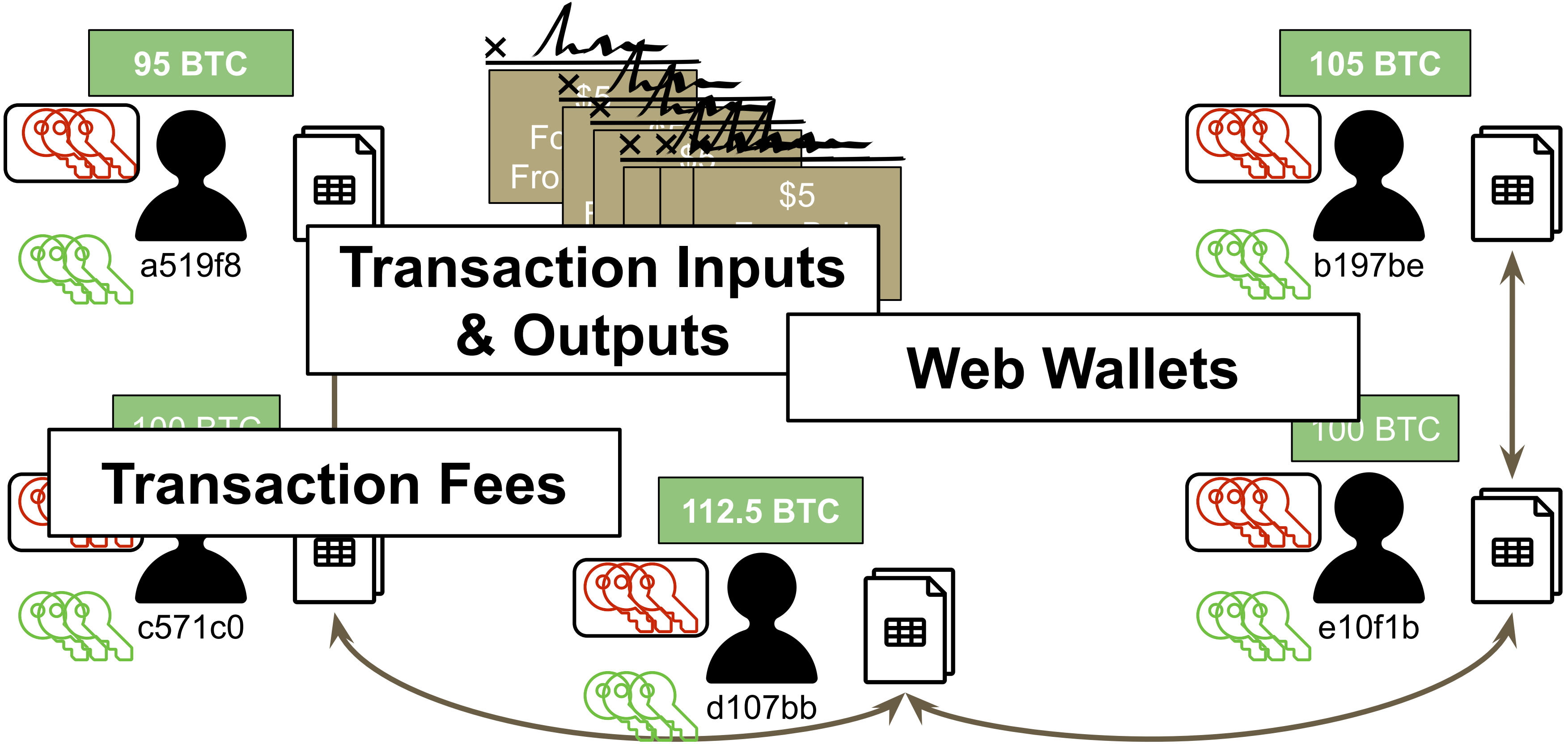

95 BTC
a519f8
$5
For: Bob
From: Alice
105 BTC
b197be
100 BTC
Transaction Fees
112.5 BTC
c571c0
d107bb
100 BTC
e10f1b

Bitcoin

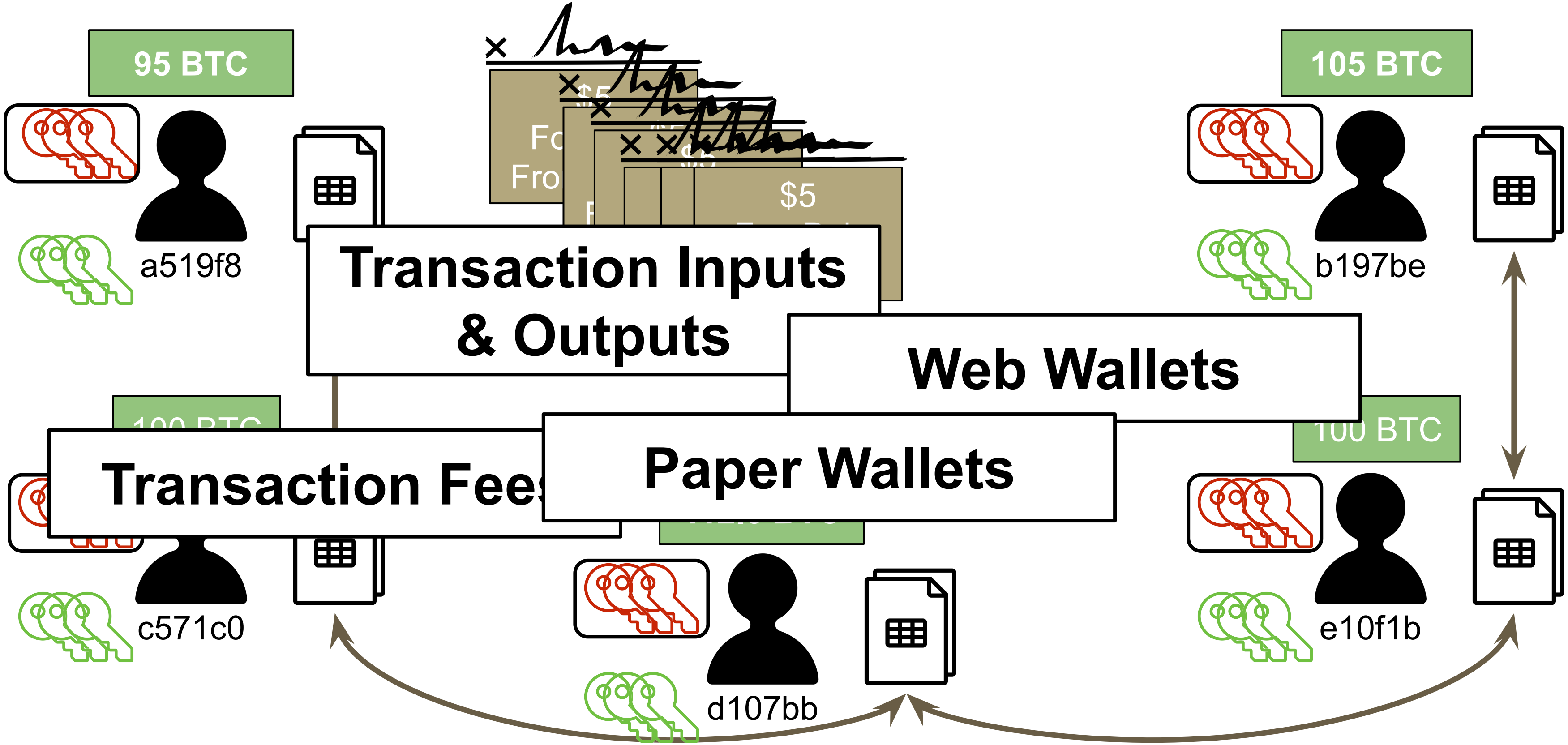


95 BTC
a519f8
$5
Transaction Inputs & Outputs
105 BTC
b197be
100 BTC
e10f1b
Transaction Fees
c571c0
112.5 BTC
d107bb

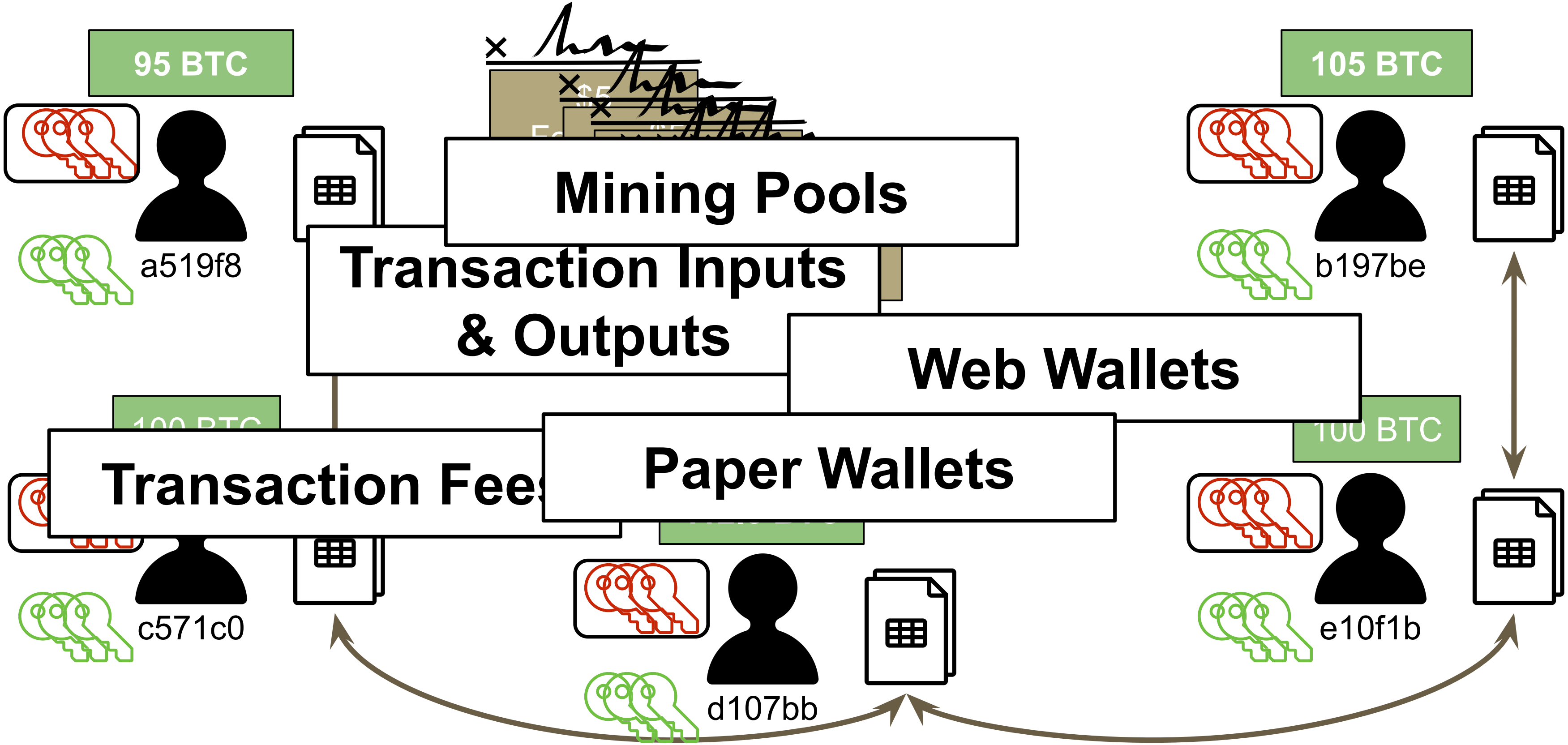
Bitcoin


95 BTC
a519f8
105 BTC
b197be
$5
Transaction Inputs & Outputs
Web Wallets
Transaction Fees
c571c0
112.5 BTC
d107bb
100 BTC
e10f1b

Bitcoin

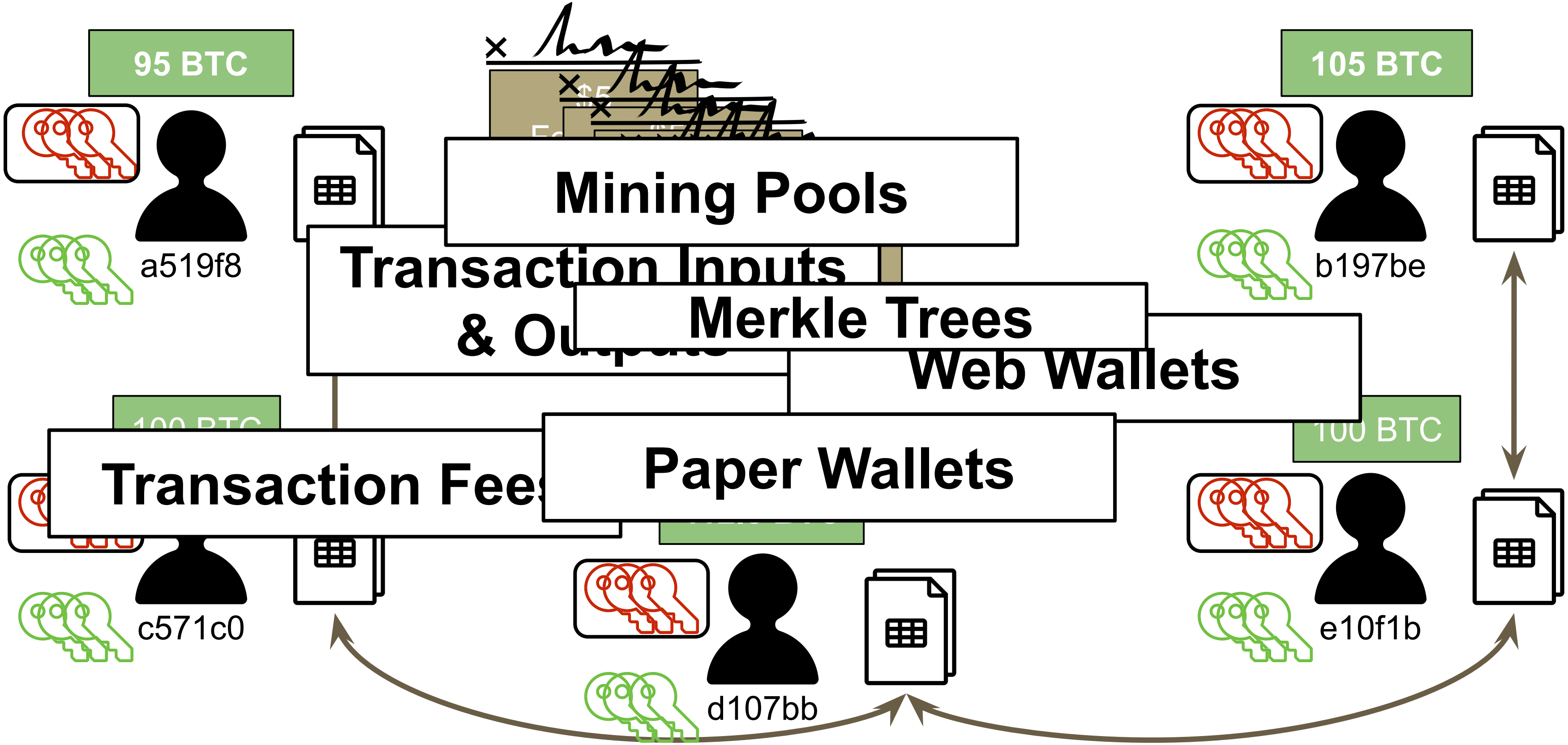

95 BTC
a519f8
105 BTC
b197be
$5
Transaction Inputs & Outputs
Web Wallets
100 BTC
Transaction Fees
Paper Wallets
c571c0
d107bb
e10f1b

Bitcoin



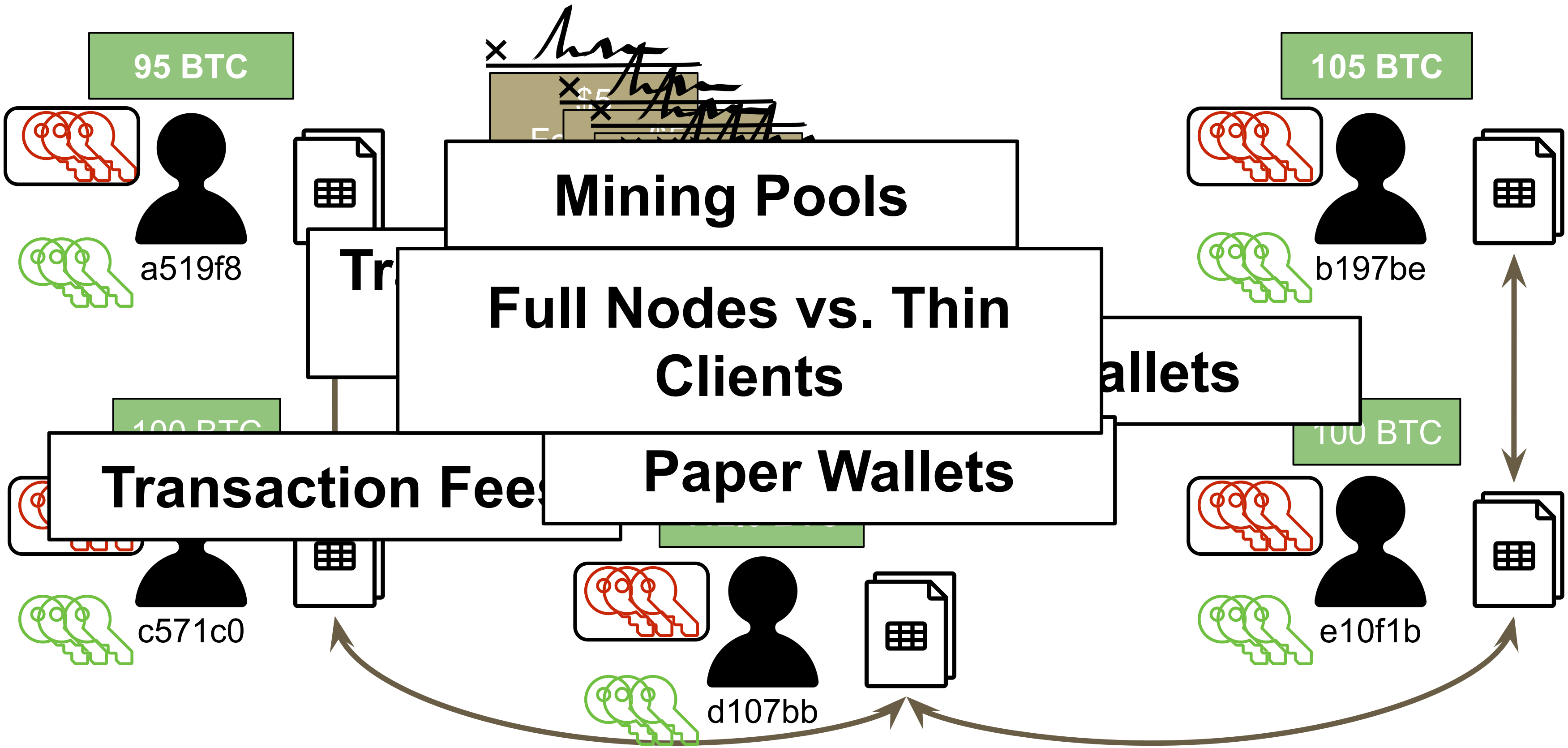
95 BTC
a519f8
105 BTC
b197be
Mining Pools
Transaction Inputs & Outputs
Web Wallets
Transaction Fees
Paper Wallets
100 BTC
c571c0
d107bb
e10f1b

Bitcoin



95 BTC
a519f8
105 BTC
b197be
Mining Pools
Transaction Inputs
& Ou
Merkle Trees
Web Wallets
100 BTC
Transaction Fees
Paper Wallets
c571c0
d107bb
e10f1b

Bitcoin


95 BTC
a519f8
105 BTC
b197be
c571c0
100 BTC
e10f1b
d107bb
Mining Pools
Full Nodes vs. Thin Clients
Transaction Fees
Paper Wallets
allets

Funny Story: guard your wallet (Dec 2013)!!

Bitcoin Questions

- Is Bitcoin anonymous?
- Is Bitcoin really invulnerable to compromise?

Bitcoin Questions

- **Is Bitcoin anonymous?**

- Is Bitcoin really invulnerable to compromise?

BLOCKCHAIN info

Home | Charts | Stats | Markets | API | Wallet | English

Home Welcome to Blockchain

More...

Height	Age	Transactions	Total Sent	Relayed By	Size (kB)
453057	10 minutes	1884	15,911.31 BTC	Bitcoin.com	998.11
453056	16 minutes	1834	27,998.50 BTC	ViaBTC	999.16
453055	38 minutes	2331	17,512.90 BTC	BitFury	998.18
453054	48 minutes	2524	17,116.92 BTC	F2Pool	999.91
453053	59 minutes	2321	15,615.56 BTC	AntPool	998.09
453052	1 hour 15 minutes	2096	9,727.30 BTC	BTCC Pool	998.12

Latest Transactions

9d92666f7028abe9860f7235f...	< 1 minute	27.50688943 BTC
d3939c0ce5b3521cd26d16d73...	< 1 minute	0.006798 BTC

Search

You may enter a block height, address, block hash, transaction hash, hash160, or ipv4 address...

Address / ip / SHA hash | Search

Block #453054

Summary	
Number Of Transactions	2524
Output Total	17,116.9190252 BTC
Estimated Transaction Volume	3,595.87663859 BTC
Transaction Fees	1.24502972 BTC
Height	453054 (Main Chain)
Timestamp	2017-02-14 17:06:39
Received Time	2017-02-14 17:06:39
Relayed By	F2Pool
Difficulty	422,170,566,883.84
Bits	402823865
Size	999.913 KB

Hashes	
Hash	000000000000000001a13b341900b61b36ad8664ceae30da3cc0c52d9faa0b99
Previous Block	000000000000000000734158f091f9918677ccdc9e50281794c5f4f433ec582a
Next Block(s)	000000000000000000ee1b60d8fde428589e7ac37ec35b3c7de9119040047043
Merkle Root	759f9e4d7c7266410e5b40a7f245e757b9eb69873ecdf7f0e3f45a25b2467467

Network Propagation

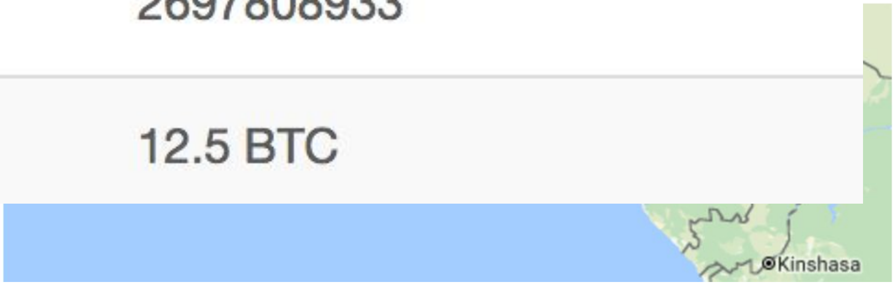
Block #453054

Summary		Hashes

Bits	402823865
Size	999.913 KB
Version	0x20000000
Nonce	2697808933
Block Reward	12.5 BTC

Nu a0b99
Ou 582a
Es
Tra 47043
He 467
Tir
Re
Re
Dif
Bit

Size	999.913 KB

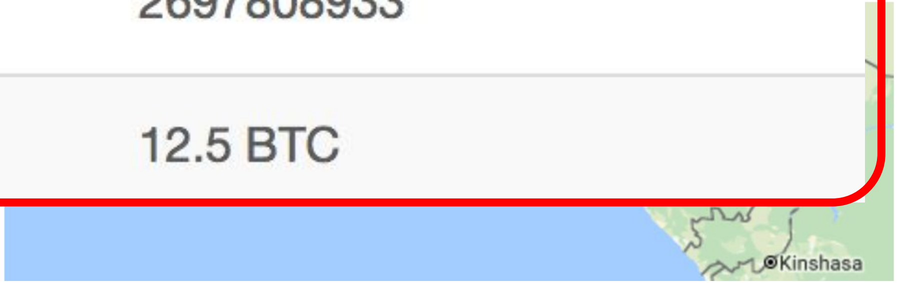


Block #453054

Summary	Hashes

Bits	402823865
Size	999.913 KB
Version	0x20000000
Nonce	2697808933
Block Reward	12.5 BTC

Size	999.913 KB

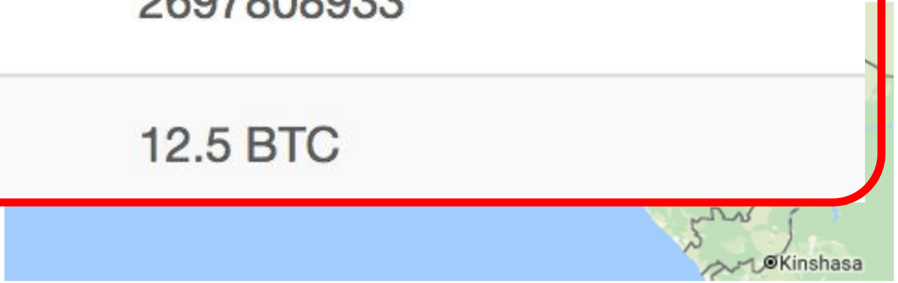


Block #453054

Summary	Hashes

Bits	402823865
Size	999.913 KB
Version	0x20000000
Nonce	2697808933
Block Reward	12.5 BTC

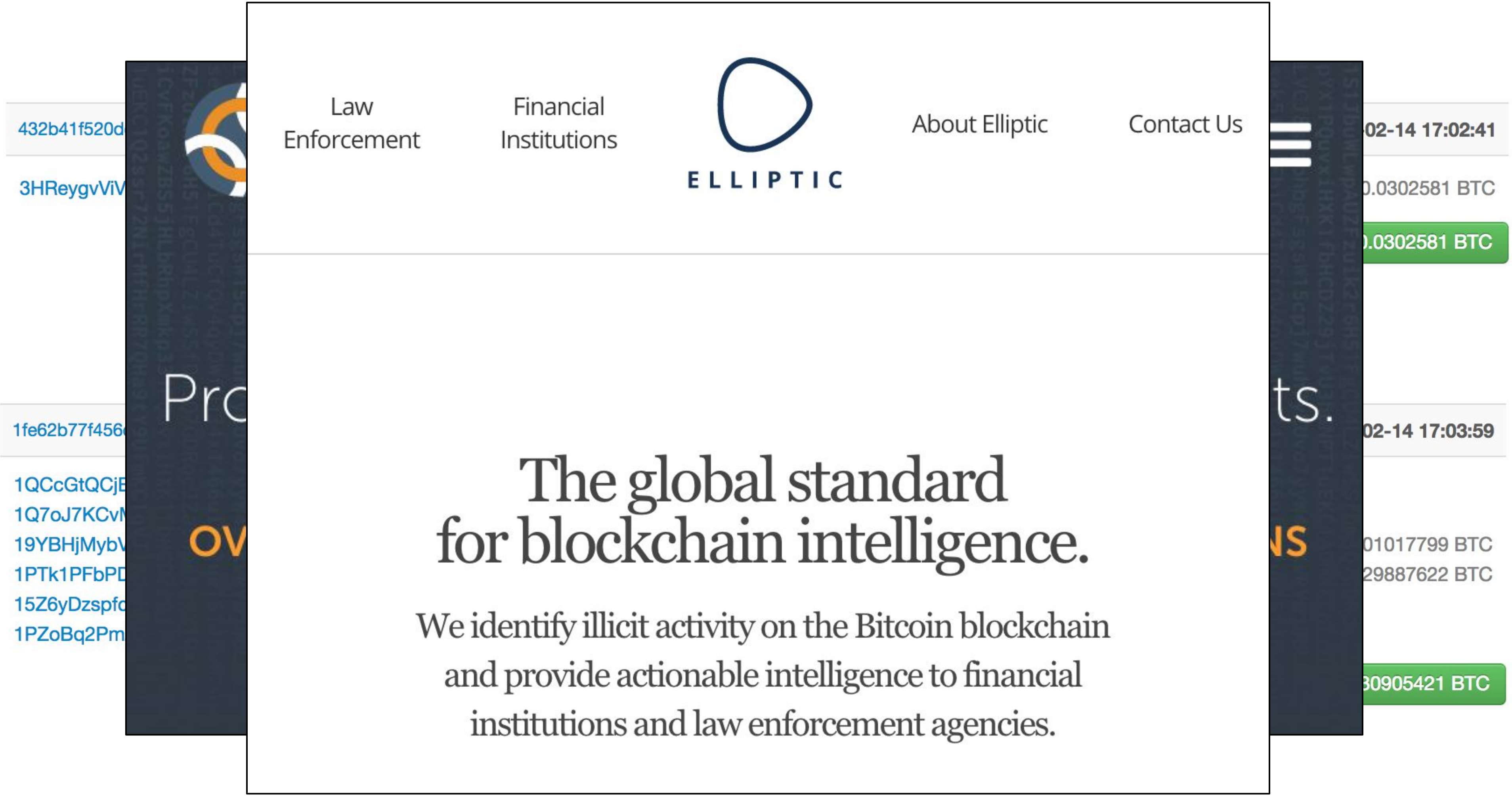
Size	999.913 KB



432b41f520dd8806531db5bcd1bc418e9cfdbe9653f16d6579fa9f26962f6215			**2017-02-14 17:02:41**
3HReygvViVEPKq81yNfvHGpFwvZiLarWhf	→	13MhE199nVsxVDaH23PZMFdjzHXBZ8emQc	0.0302581 BTC
			0.0302581 BTC

1fe62b77f456d8cf180e9da499fc393db383bcf8d4d1317abe2e7349eff93d18			**2017-02-14 17:03:59**
1QCcGtQCjBCV6N4xrCmM36osTBdvuY4Dyj 1Q7oJ7KCvM3s2TnuhLkpYgaHHcmGUSoTt5 19YBHjMybV86puPbaY4mWJZ417pVuzY2aq 1PTk1PFbPDCdhD6fySExnZ72bGrDYttbrn 15Z6yDzspfcouWFt51hfiMRuXf5HgUag1M 1PZoBq2PmZTmDLVBUaqdNBGTbNYdJGZSou	→	1LzqQ7oj49pwr6pDbNUT4usxt9C4qxAr7v 1NfC2rPsdMUfabCZ7D1VjfYJp5crqt1F6f	0.01017799 BTC 0.29887622 BTC
			0.30905421 BTC


CHAINALYSIS
Protecting the integrity of digital assets.
OVER $15 BILLION WORTH OF BITCOIN TRANSACTIONS CHECKED BY CHAINALYSIS ON BEHALF OF OUR CUSTOMERS
432b41f520d
02-14 17:02:41
3HReygvViV
0.0302581 BTC
1fe62b77f456
02-14 17:03:59
1QCcGtQCjE
1Q7oJ7KCvM
19YBHjMybV
1PTk1PFbPD
15Z6yDzspfc
1PZoBq2Pm
01017799 BTC
29887622 BTC
30905421 BTC


Law Enforcement
Financial Institutions
ELLIPTIC
About Elliptic
Contact Us
The global standard for blockchain intelligence.
We identify illicit activity on the Bitcoin blockchain and provide actionable intelligence to financial institutions and law enforcement agencies.

BLOCKCHAIN info

Home | Charts | Stats | Markets | API | Wallet

Search

Bitcoin Nodes Log

List of bitcoin nodes blockchain.info has connected to in the past.

Total Unique Ip Addresses: 16,043

Ip Address or Hostname

IP	Port	Last Connected	Location	Hostname
50.159.122.115	8333	2017-02-13 12:48:15	US	c-50-159-122-115.hsd1.wa.comcast.net
66.175.217.124	8333	2017-02-13 12:48:15	US (Absecon)	li512-124.members.linode.com
212.164.233.103	8333	2017-02-13 12:48:14	RU (Novosibirsk)	212.164.233.103
52.62.57.222	8333	2017-02-13 12:48:14	US (Wilmington)	ec2-52-62-57-222.ap-southeast-2.compute.amazonaws.com
195.67.36.89	8333	2017-02-13 12:48:13	SE (Vimmerby)	195-67-36-89.customer.telia.com
89.142.75.86	8333	2017-02-13 12:48:13	SI (Polzela)	BSN-142-75-86.dynamic.siol.net

Bitcoin Questions

- **Is Bitcoin anonymous?**

- Is Bitcoin really invulnerable to compromise?

Bitcoin Questions

- **Is Bitcoin anonymous?**

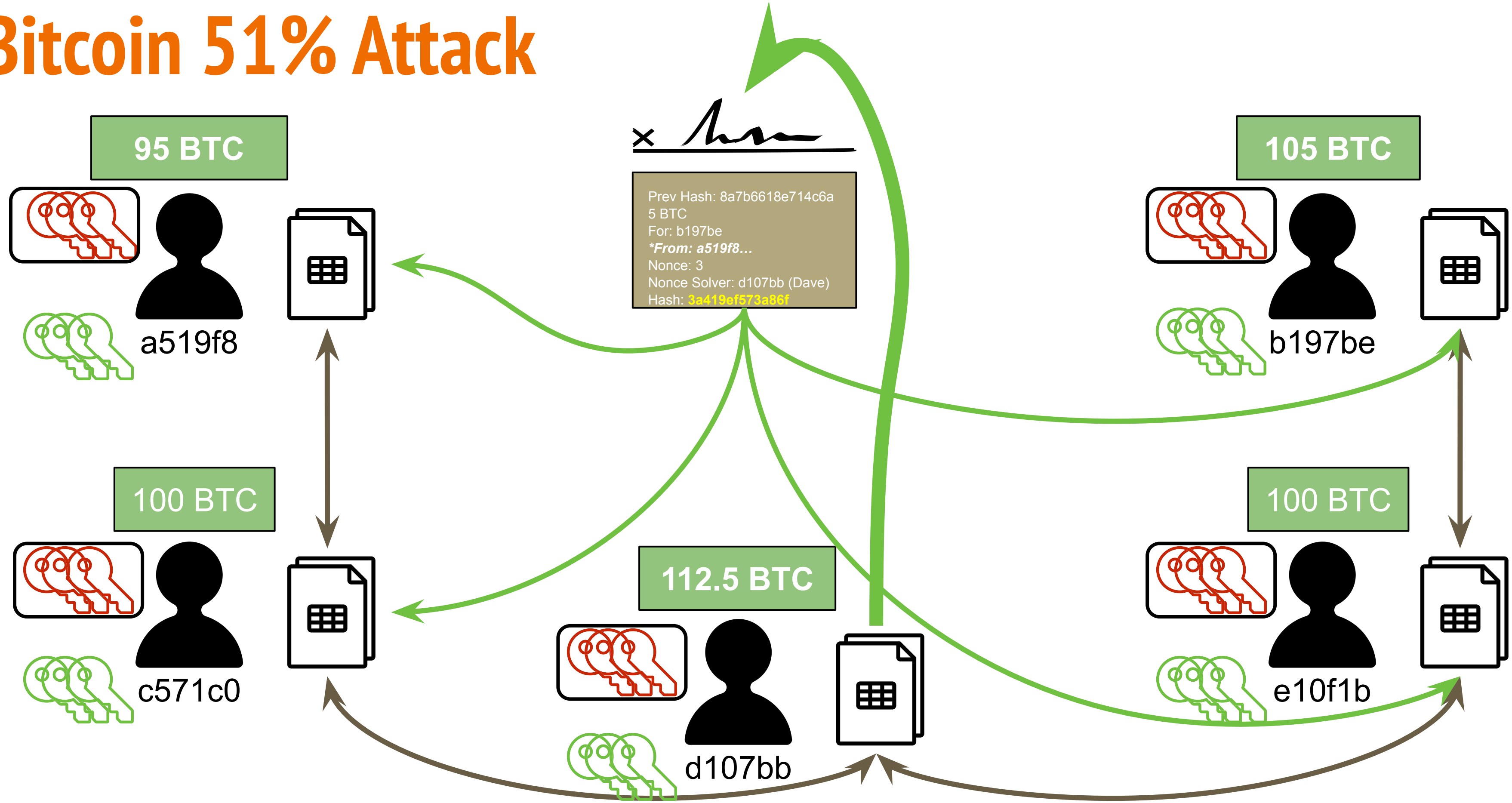
  Yes and No! It is pseudonymous.
- Is Bitcoin really invulnerable to compromise?

Bitcoin Questions

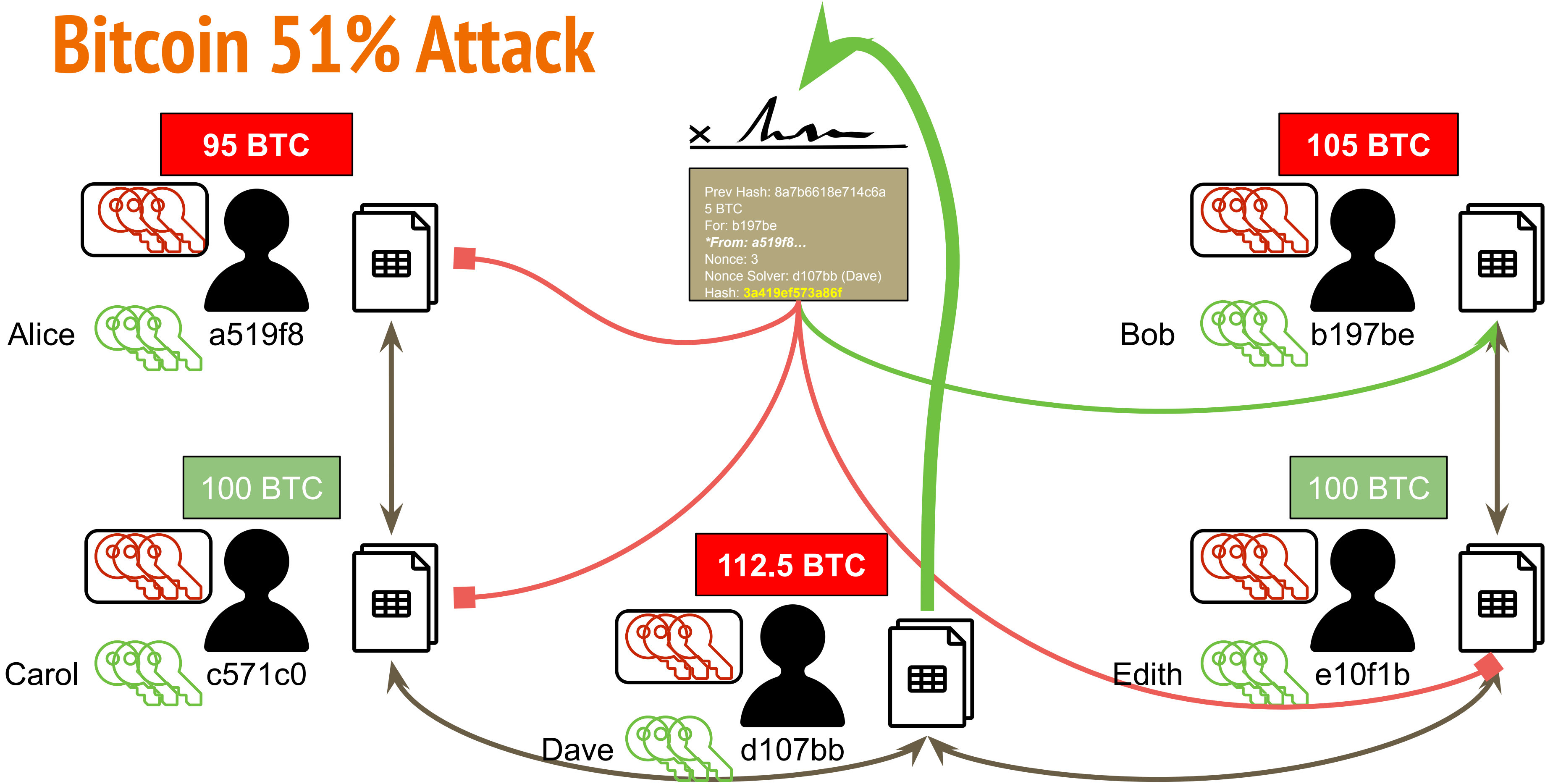
- Is Bitcoin anonymous?

  Yes and No! It is pseudonymous.
- **Is Bitcoin really invulnerable to compromise**?

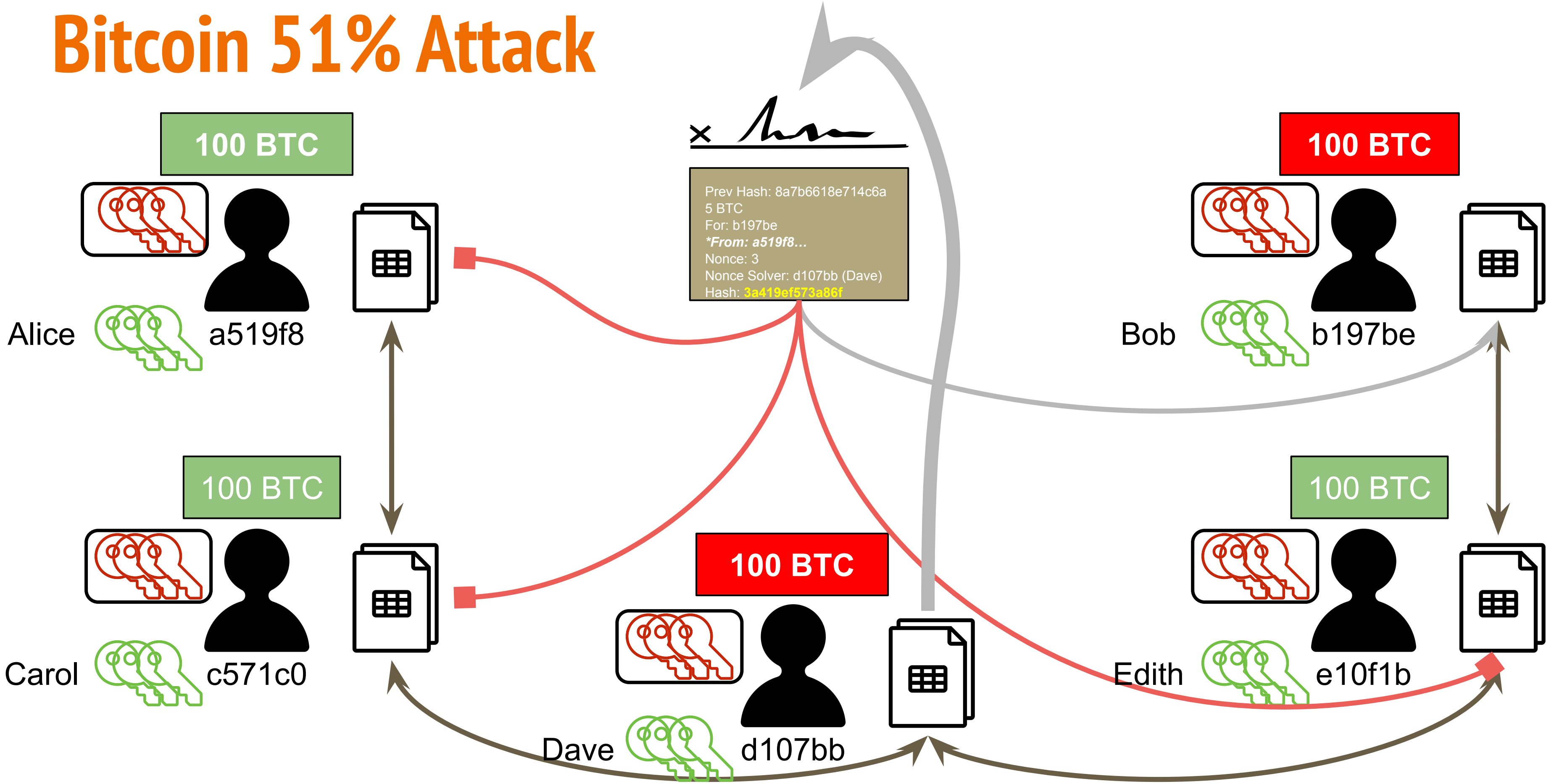
Bitcoin 51% Attack



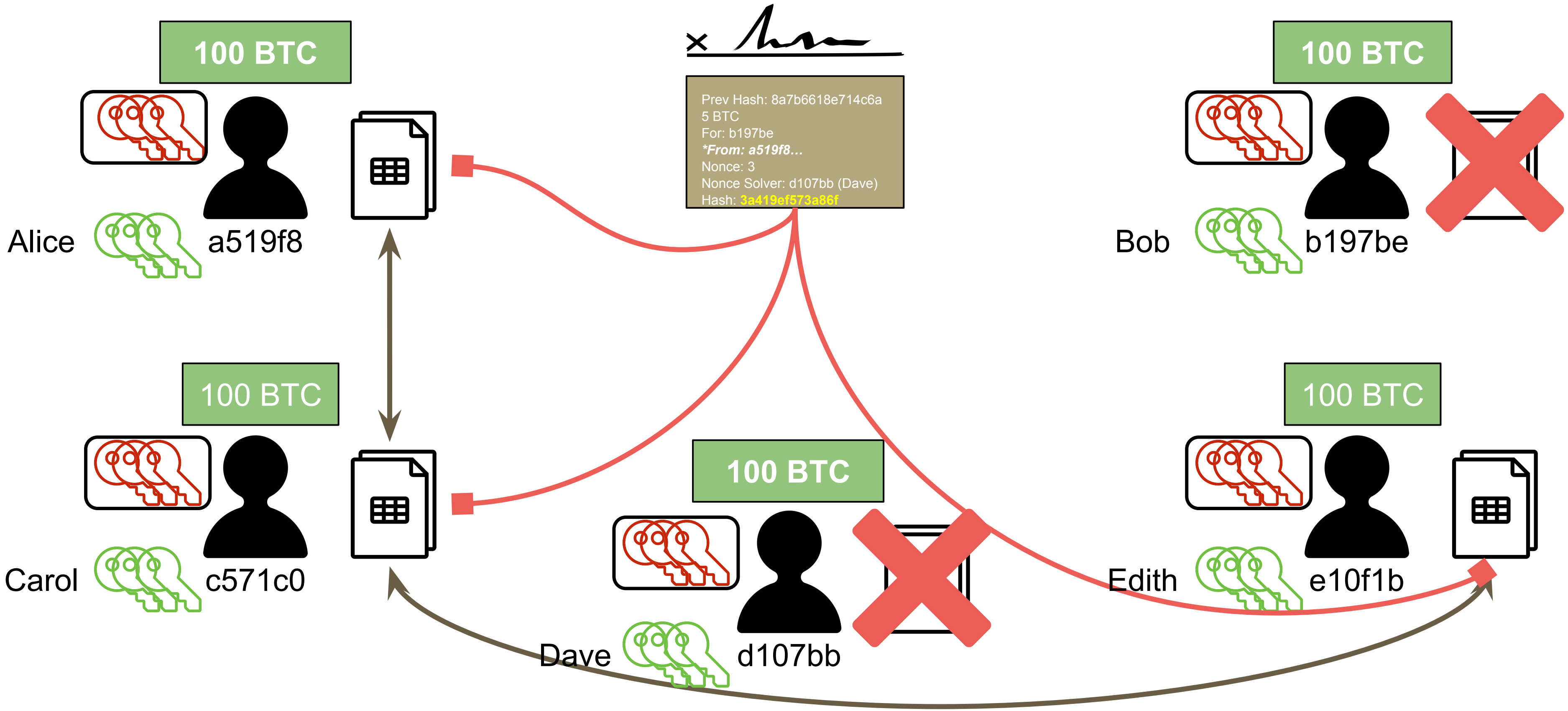
Bitcoin 51% Attack



Bitcoin 51% Attack



Bitcoin 51% Attack



Could this actually happen?

Could this actually happen?

ars TECHNICA SIGN IN

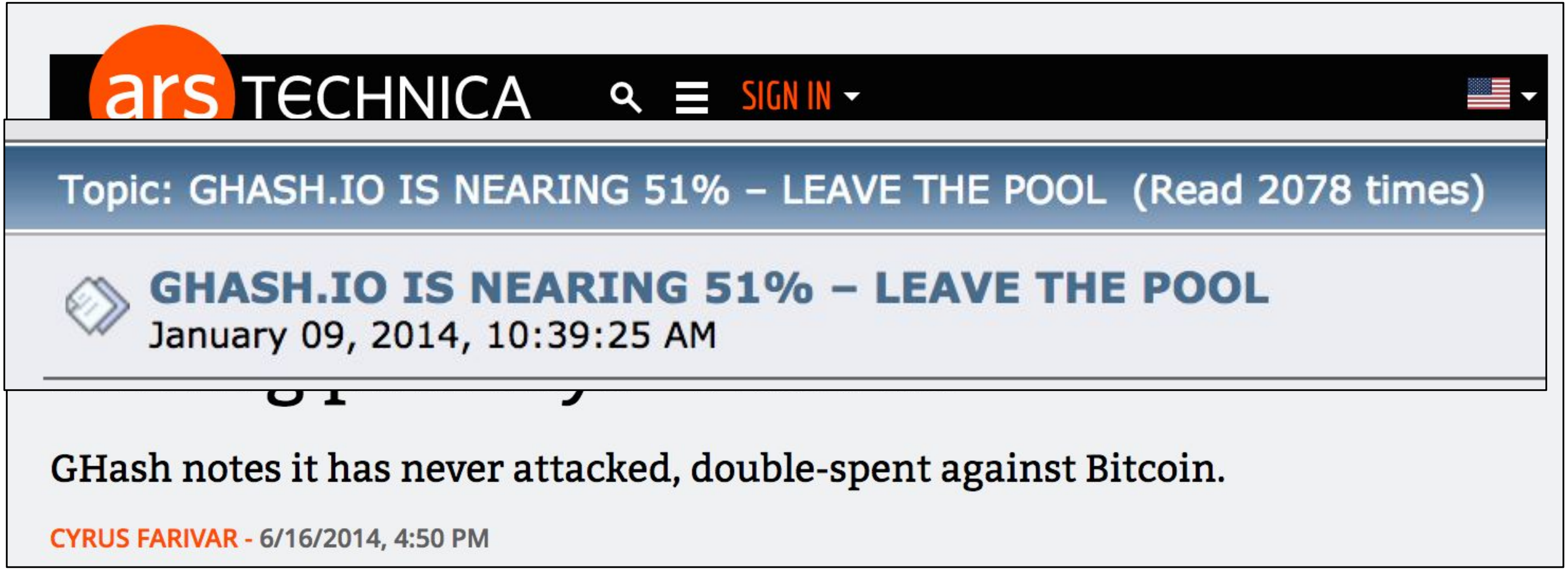
RISK ASSESSMENT —

After reaching 51% network power, Bitcoin mining pool says "trust us"

GHash notes it has never attacked, double-spent against Bitcoin.

CYRUS FARIVAR - 6/16/2014, 4:50 PM

Could this actually happen?



ars TECHNICA SIGN IN

Topic: GHASH.IO IS NEARING 51% – LEAVE THE POOL (Read 2078 times)

GHASH.IO IS NEARING 51% – LEAVE THE POOL

January 09, 2014, 10:39:25 AM

GHash notes it has never attacked, double-spent against Bitcoin.

CYRUS FARIVAR - 6/16/2014, 4:50 PM

Could this actually happen?

Could this actually happen?

Quantum Computing & Cryptography

Newsweek

TECH & SCIENCE

IS BITCOIN DOOMED?

BY ANTHONY CUTHBERTSON ON 10/12/16 AT 10:08 AM

Quantum Computing & Cryptography

Nev

WIRED SECURITY

Security

The quantum clock is ticking on encryption – and your data is under threat

Quantum computers pose a major threat to the security of our data. So what can be done to keep it safe?

By **NICOLE KOBIE**

04 Oct 2016

Quantum Computing & Cryptography

REVIEWS NEWS VIDEO HOW TO SMART HOME CARS DEALS DOWNLOAD

NSA working on quantum computer to break any encryption

The spy agency is reportedly in a race to build its own quantum computer to stay ahead of others seeking to own the mother of all decryption machines.

Bitcoin Questions

- Is Bitcoin anonymous?

  Yes and No! It is pseudonymous.
- **Is Bitcoin really invulnerable to compromise?**

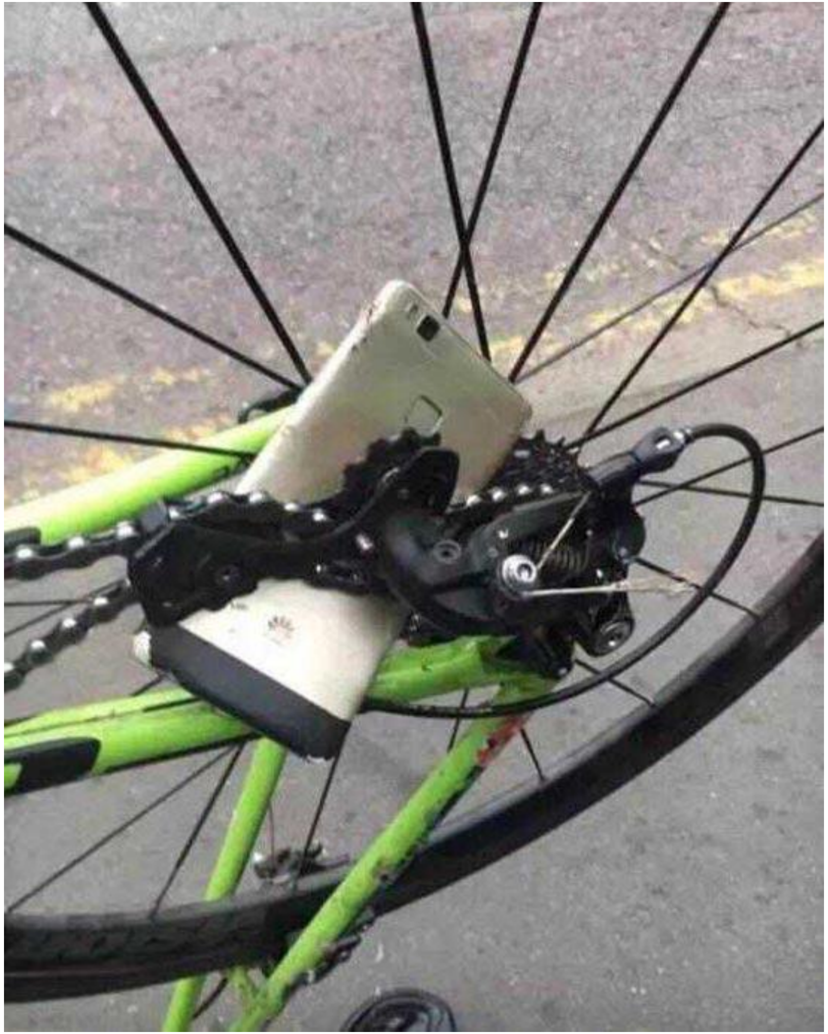
Bitcoin Questions

- Is Bitcoin anonymous?

  Yes and No! It is pseudonymous.
- **Is Bitcoin really invulnerable to compromise?**

  Probably not, at least in the short term.


LATE NIGHT
WITH SETH MEYERS



Latest blocks

View more block

Height	Hash	Mined	Miner	Size
588944	0000000000000000001be255f513d686d195ebf...	17:30 PM	BTC.com	1,274,162 bytes
588943	00000000000000000017db20ea351d2d0e425fb...	17:18 PM	BTC.com	1,104,880 bytes
588942	000000000000000000008208c98e373a2db0542...	17:14 PM	AntPool	1,335,648 bytes
588941	0000000000000000000051c28e6ec6e1b7b30920...	17:04 PM	F2Pool	1,249,497 bytes
588940	00000000000000000004ca36580eaee738367d...	16:59 PM	F2Pool	1,366,541 bytes
588939	000000000000000000067173d7fe7c2ba7bcdec...	16:37 PM	Unknown	1,076,107 bytes
588938	0000000000000000003ea92f022801fce2965d...	16:37 PM	SlushPool	1,223,835 bytes
588937	0000000000000000015aaaef25cd813d574781e...	16:24 PM	AntPool	1,212,669 bytes
588936	000000000000000000f793419624af43f22c9d...	16:24 PM	Unknown	1,237,063 bytes
588935	00000000000000000004349f5e9d4140247cae6...	16:22 PM	AntPool	1,268,144 bytes
588934	0000000000000000000ea4d6189877fe9048da7...	16:18 PM	Unknown	1,242,545 bytes
588933	00000000000000000004092599b25e760c74b6...	16:08 PM	Unknown	1,179,208 bytes
588932	00000000000000000003011df1e9bc109685206...	16:05 PM	Unknown	1,188,151 bytes
588931	0000000000000000000e066ff0d28f32d86478...	16:02 PM	Unknown	1,248,546 bytes
588930	000000000000000000090e0bef88087be6b787a...	15:58 PM	BTC.TOP	1,181,402 bytes

VI. Beyond Bitcoin

What applications does the blockchain have beyond cryptocurrencies like Bitcoin?

Other Cryptocurrencies

Other Cryptocurrencies

Other Cryptocurrencies

litecoin

Primecoin

Other Cryptocurrencies

litecoin

Primecoin

namecoin

VI. Beyond Bitcoin

Ethereum

Other Cryptocurrencies

ethereum

What is Ethereum?

- Simply put, it is an “open-source and globally decentralized computing infrastructure that executes programs called Smart Contracts. It uses blockchain to synchronize and store system’s state changes, using a cryptocurrency called Ether (ETH) to meter [or measure] and constrain execution resource costs.”

**Derived from definition put forth by Antonopoulos & Wood*

What is Ethereum?

- It shares many similarities & common elements with Bitcoin or other cryptocurrencies (P2P network connecting participants, Byzantine Fault Tolerant consensus algos, proofs, hashes, sigs)
- But it's also different in other aspects, esp in having Utility Functions ("world computer, virtual machine") + "general purpose blockchain" & Decentralized Applications (dApps or DApps!)

“Smart” Contracts: records of prog. agreements

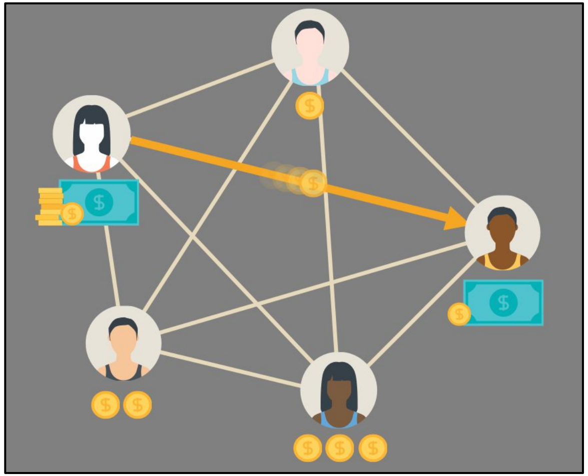
- Ethereum contracts are programs that control money, running inside Ethereum VM
- Once created, they have an Ethereum address, just like wallets (say, belonging to a person)
- Transactions sent to an address may have ether, data, or both → ethers get “deposited” to the contract balance; data can specify a named functions (in the contract) and call it

“Smart” Contracts

ethereum

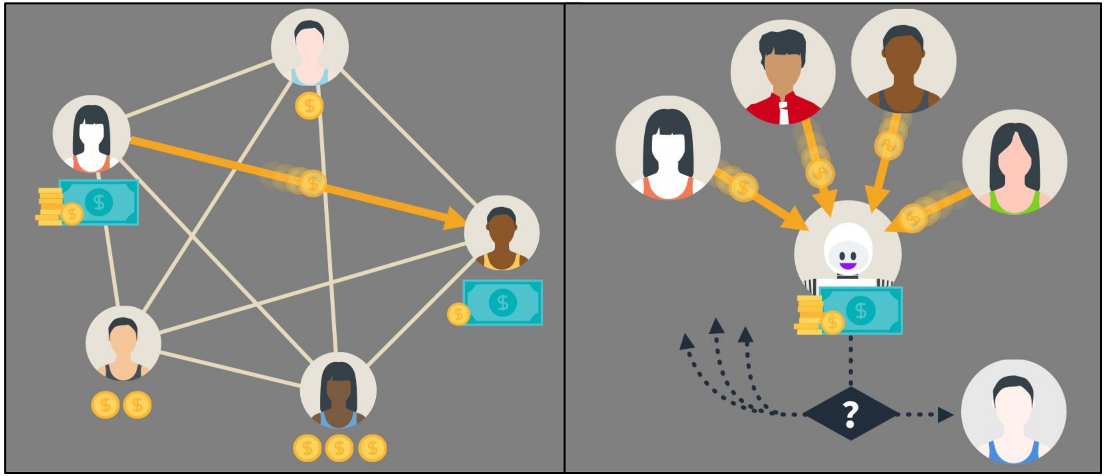
“Smart” Contracts

ethereum



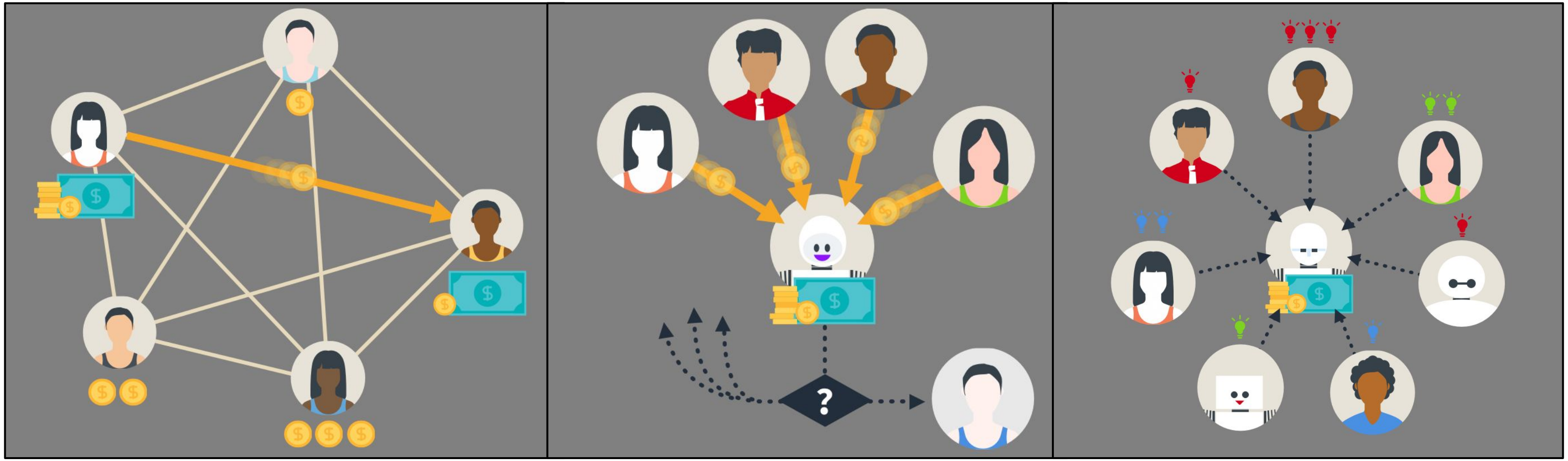
“Smart” Contracts

ethereum

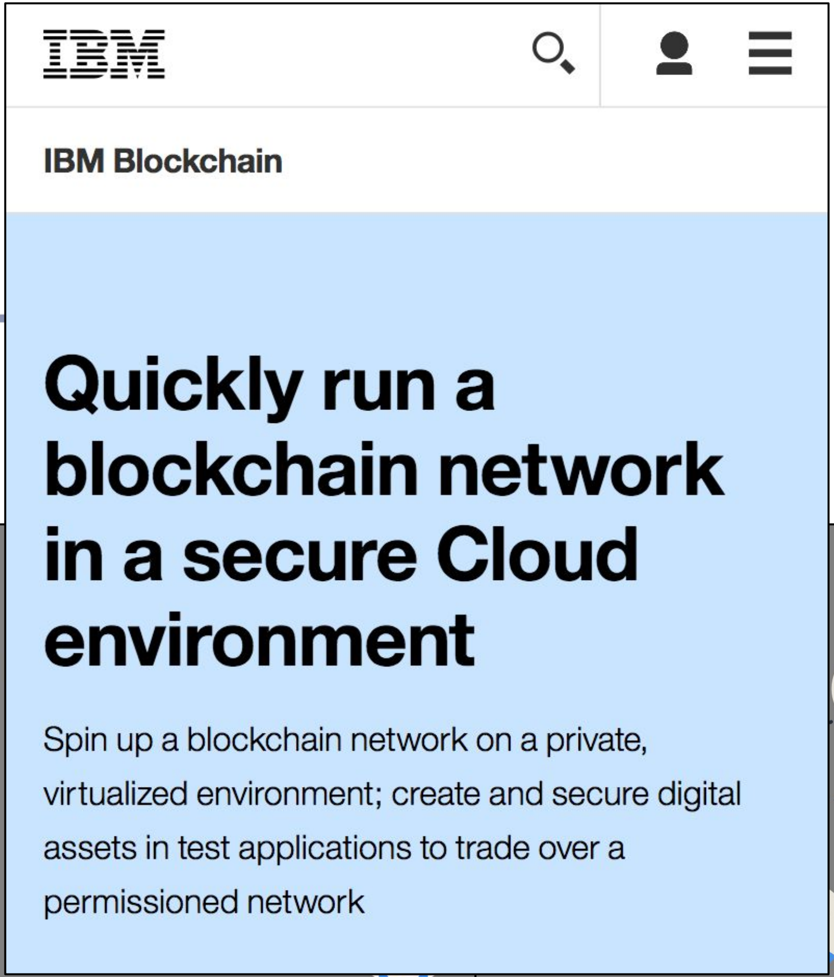


“Smart” Contracts

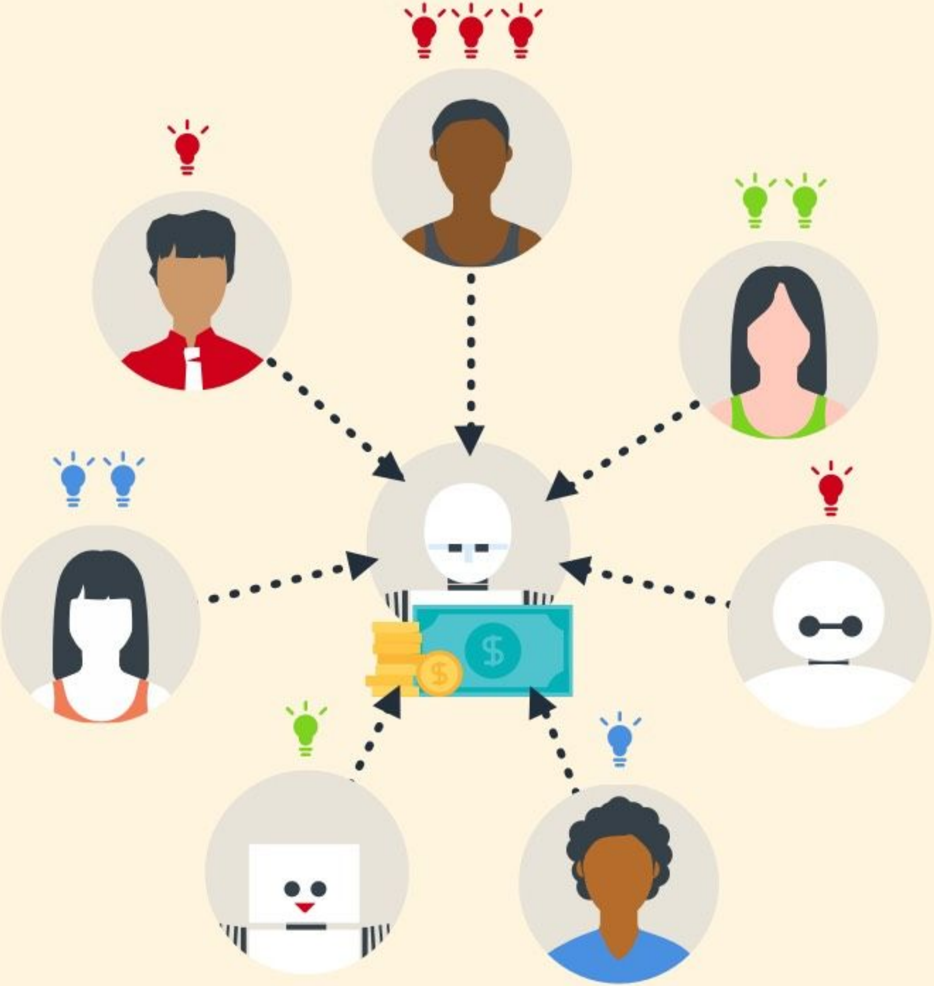
ethereum

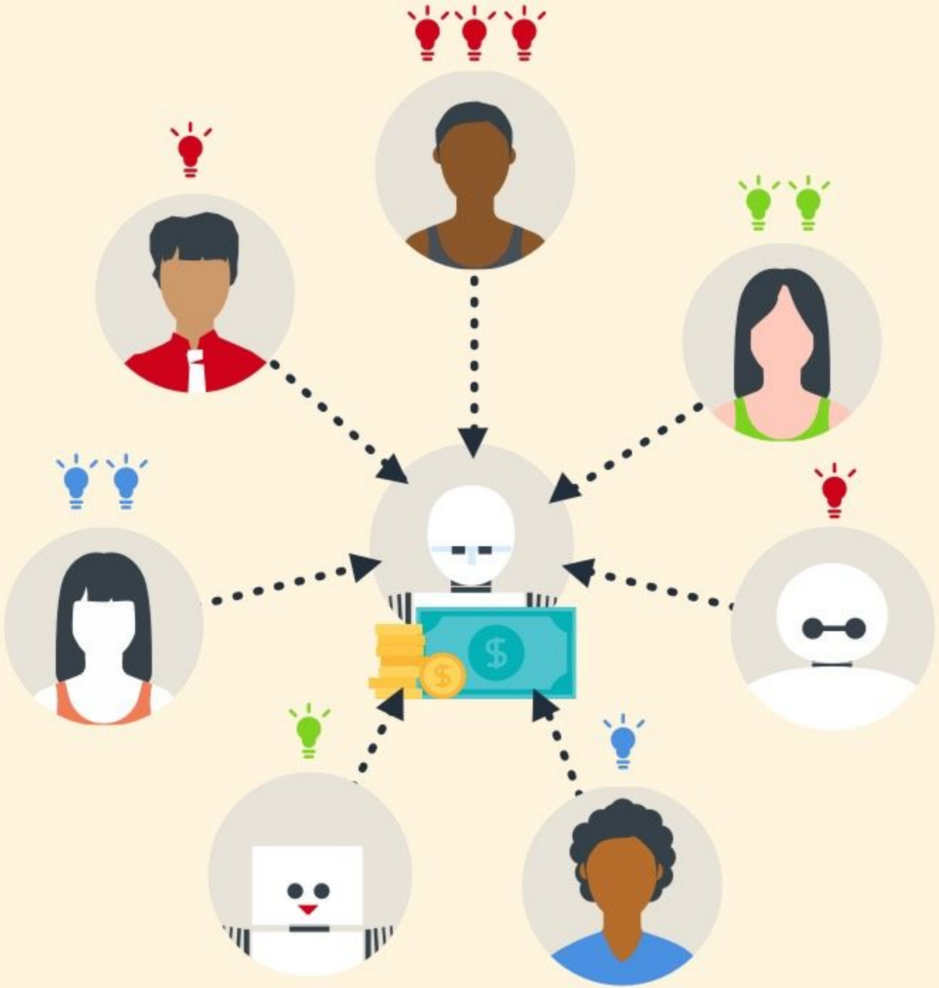
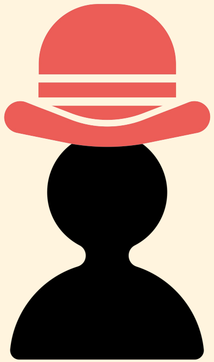


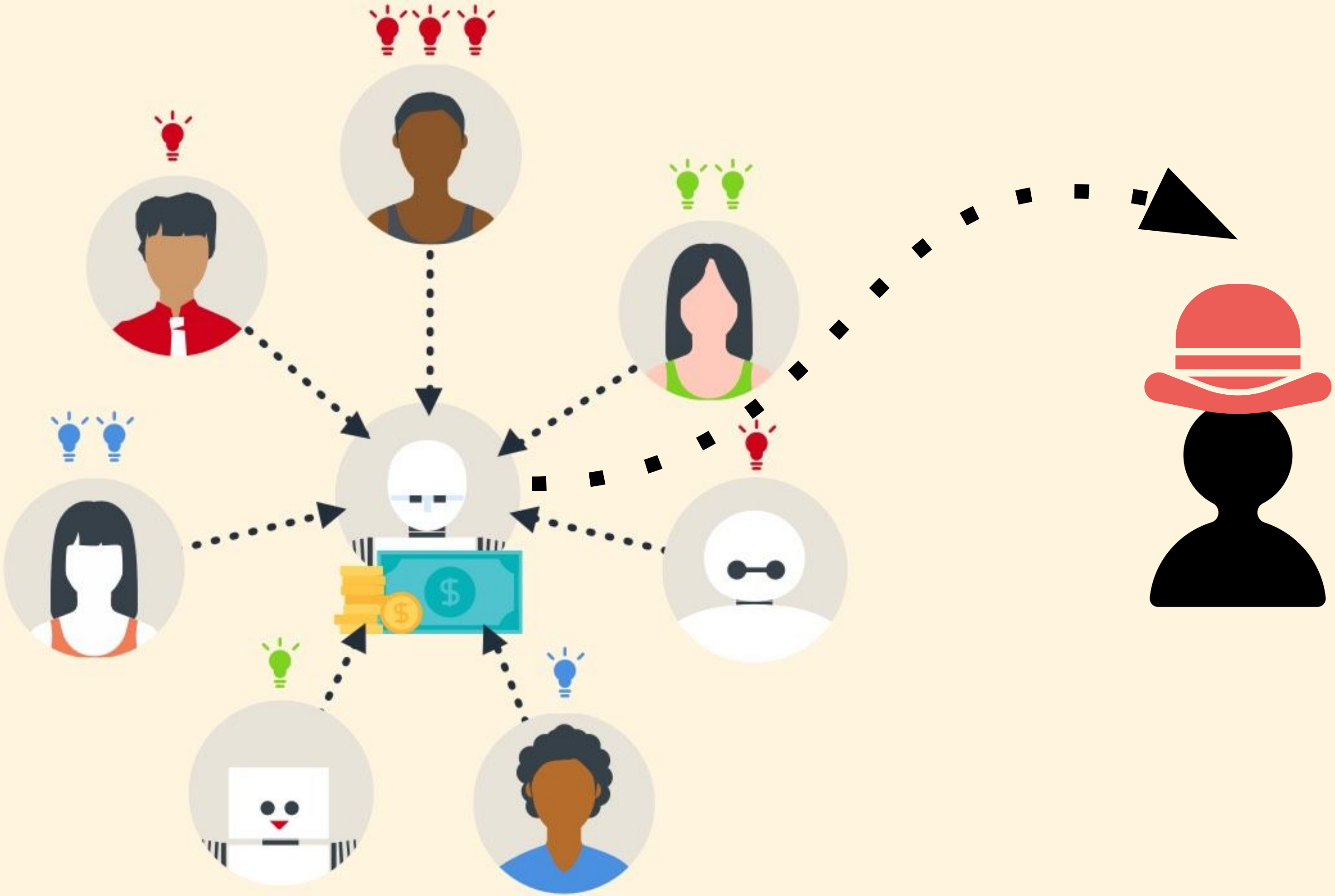
“Smart” Contracts

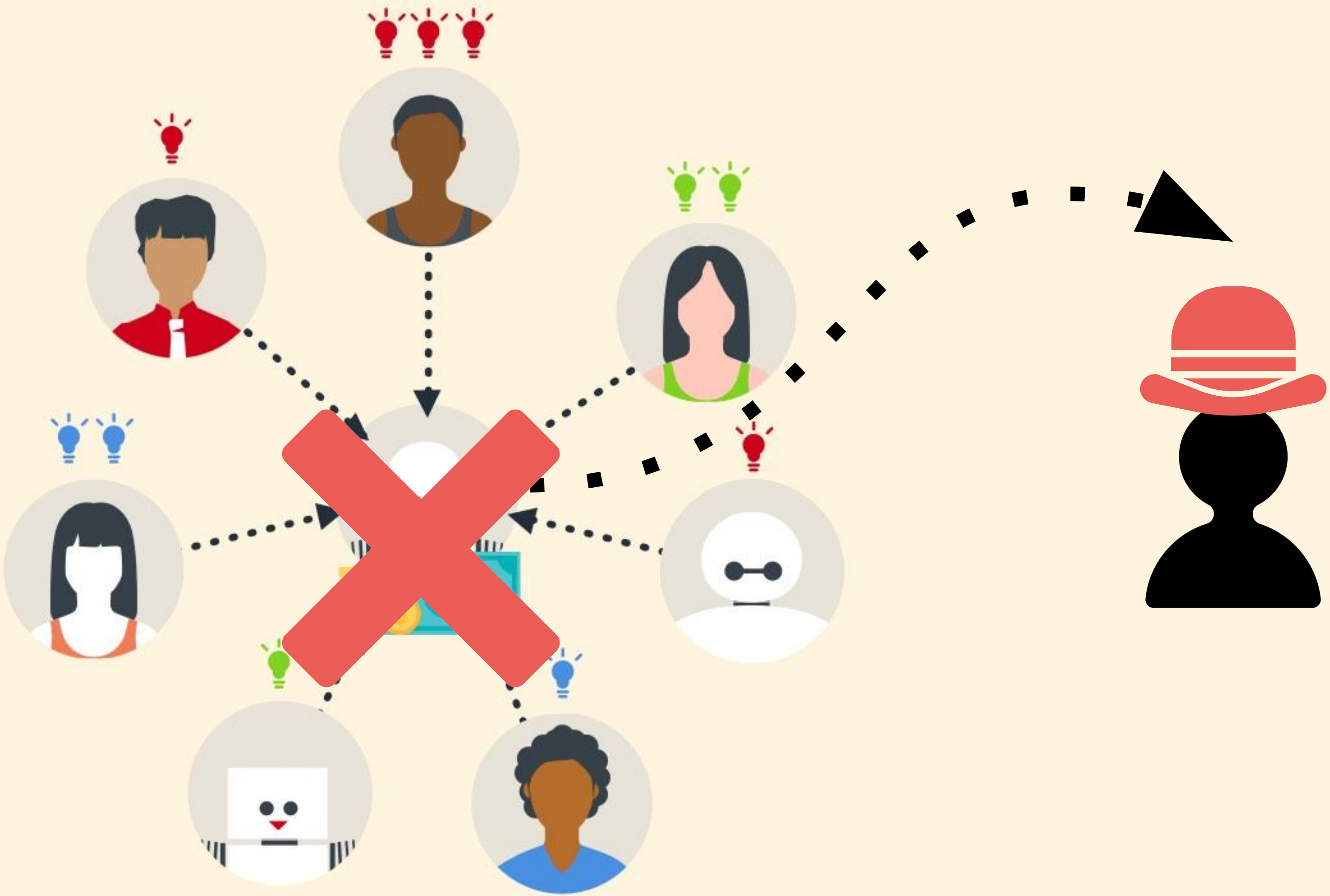


The DAO

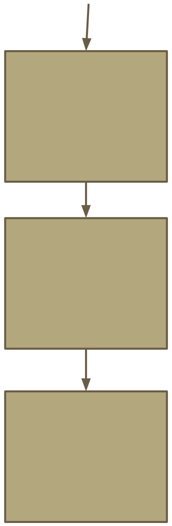


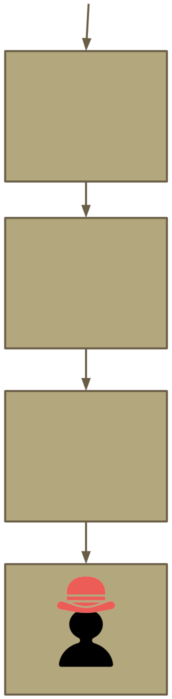




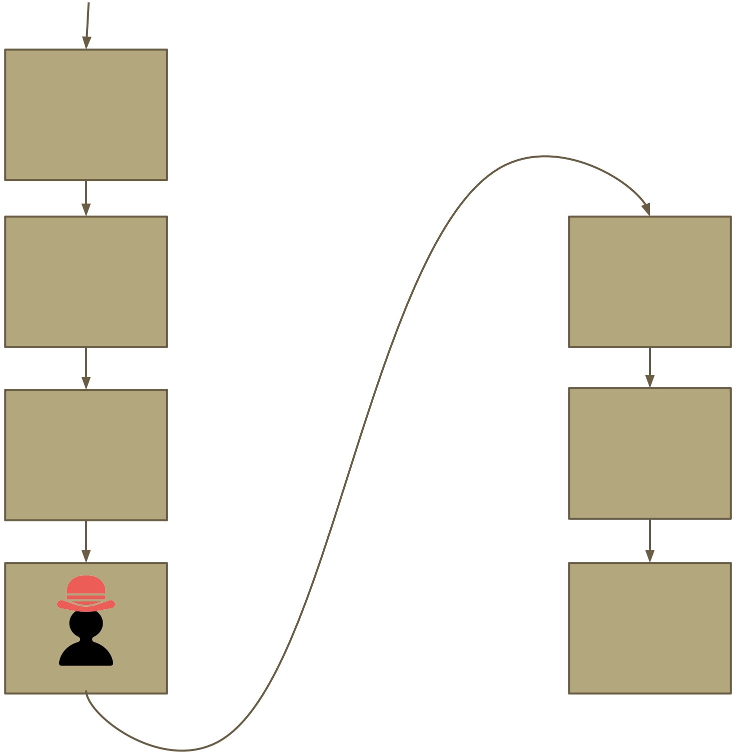
The DAO & “Forking”



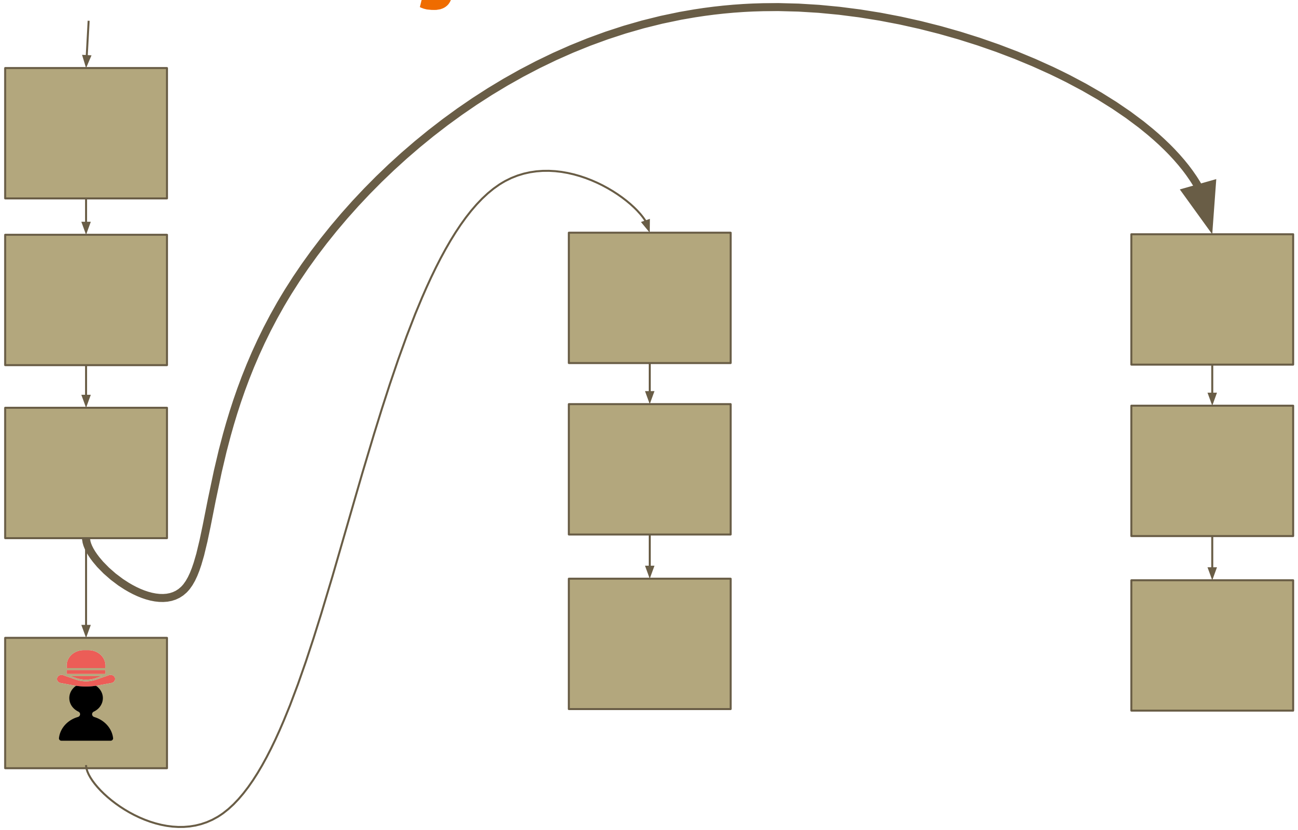
The DAO & “Forking”



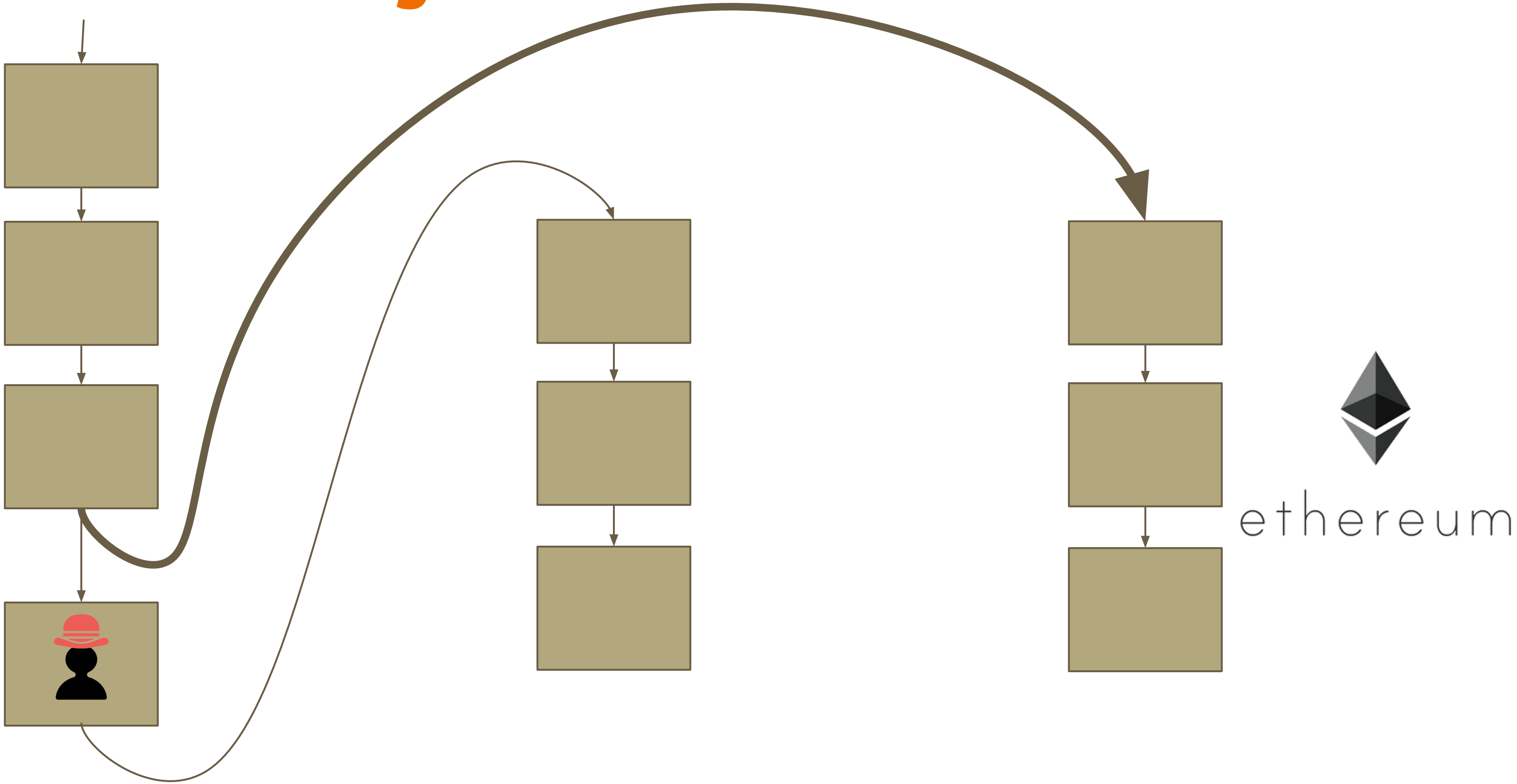
The DAO & “Forking”



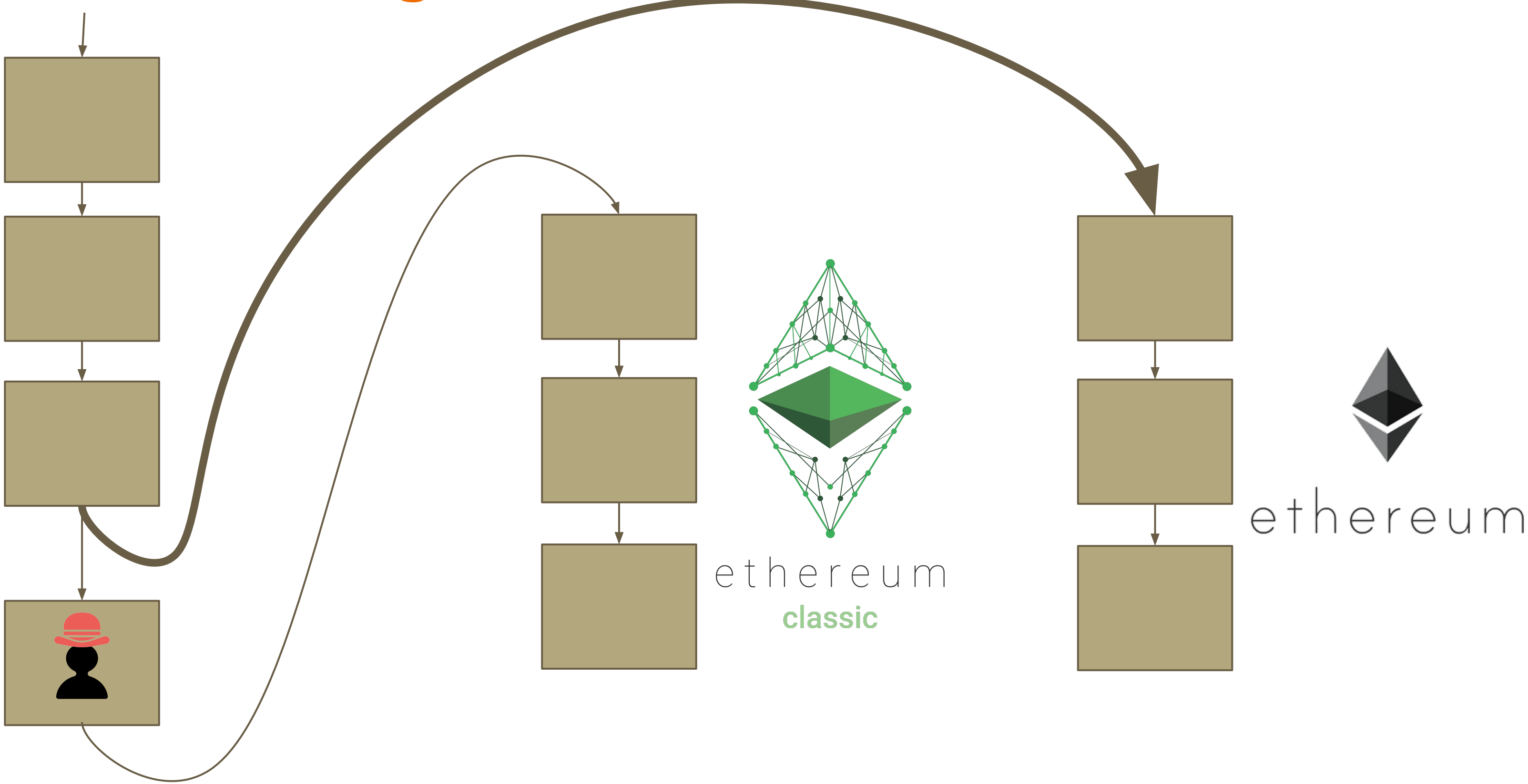
The DAO & “Forking”



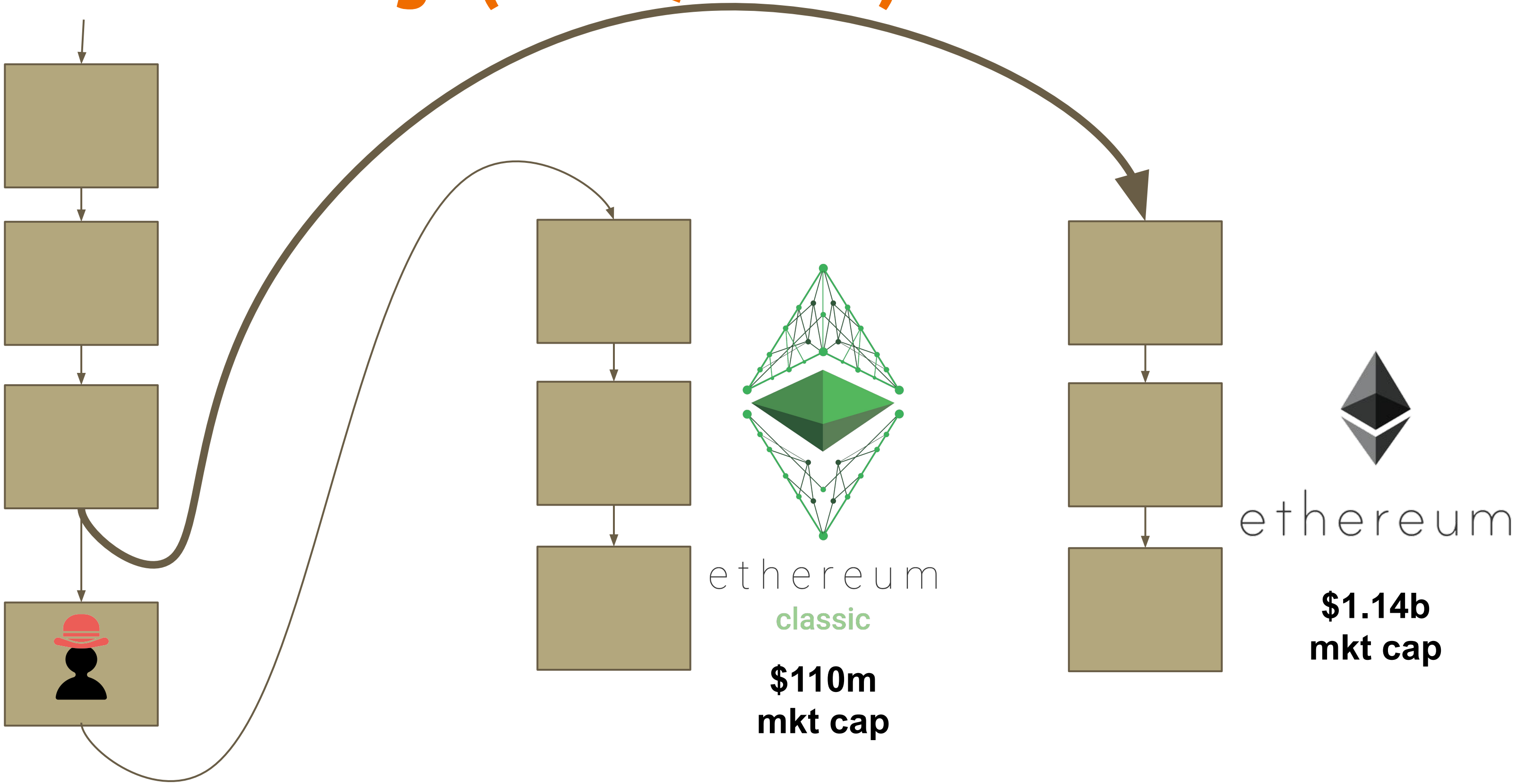
The DAO & “Forking”



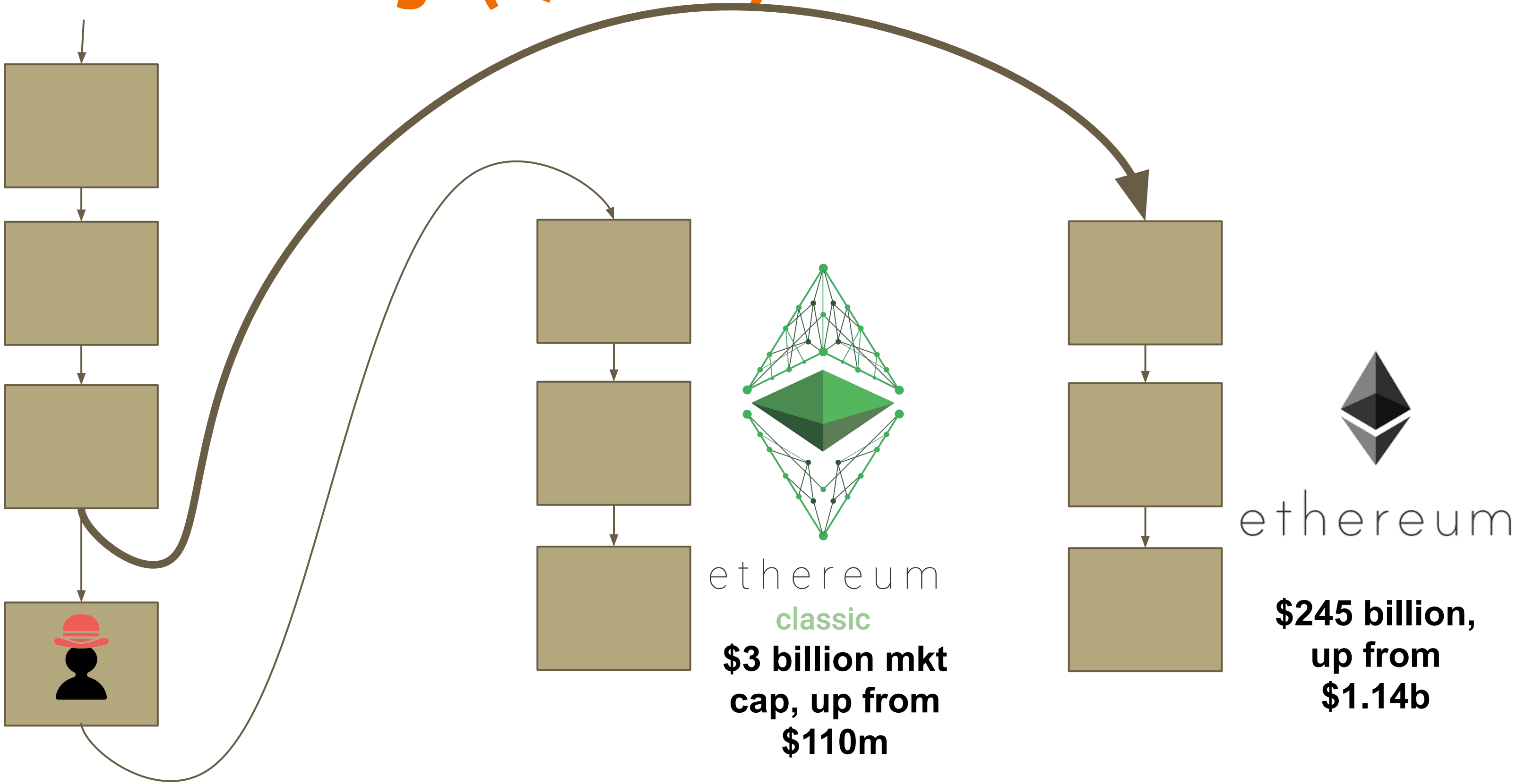
The DAO & “Forking”



The DAO & "Forking" (2016/2017)



The DAO & "Forking" (4/2023)



A quick regulatory lesson

The “Howey Test”

- It is an investment of money
- There is an expectation of profits from the investment
- The investment of money is in a common enterprise
- Any profit comes from the efforts of a promoter or third party

Class Discussion

Easter Egg … for those with a Mac ;-)

Open either 1) Finder OR 2) click on Go, then Computer …
then click on Macintosh HD at the bottom of the window,
then System ⟶ Library ⟶ Image Capture ⟶ Devices.
Once there, right click on VirtualScanner.app and choose "Show Package Contents." Open Resources, and click on "simpledoc.pdf."
What do you see?!

Congratulations!
You made it to the end of slides … almost!
We still have a few more days to go 😎

Thank You!

End of Slides