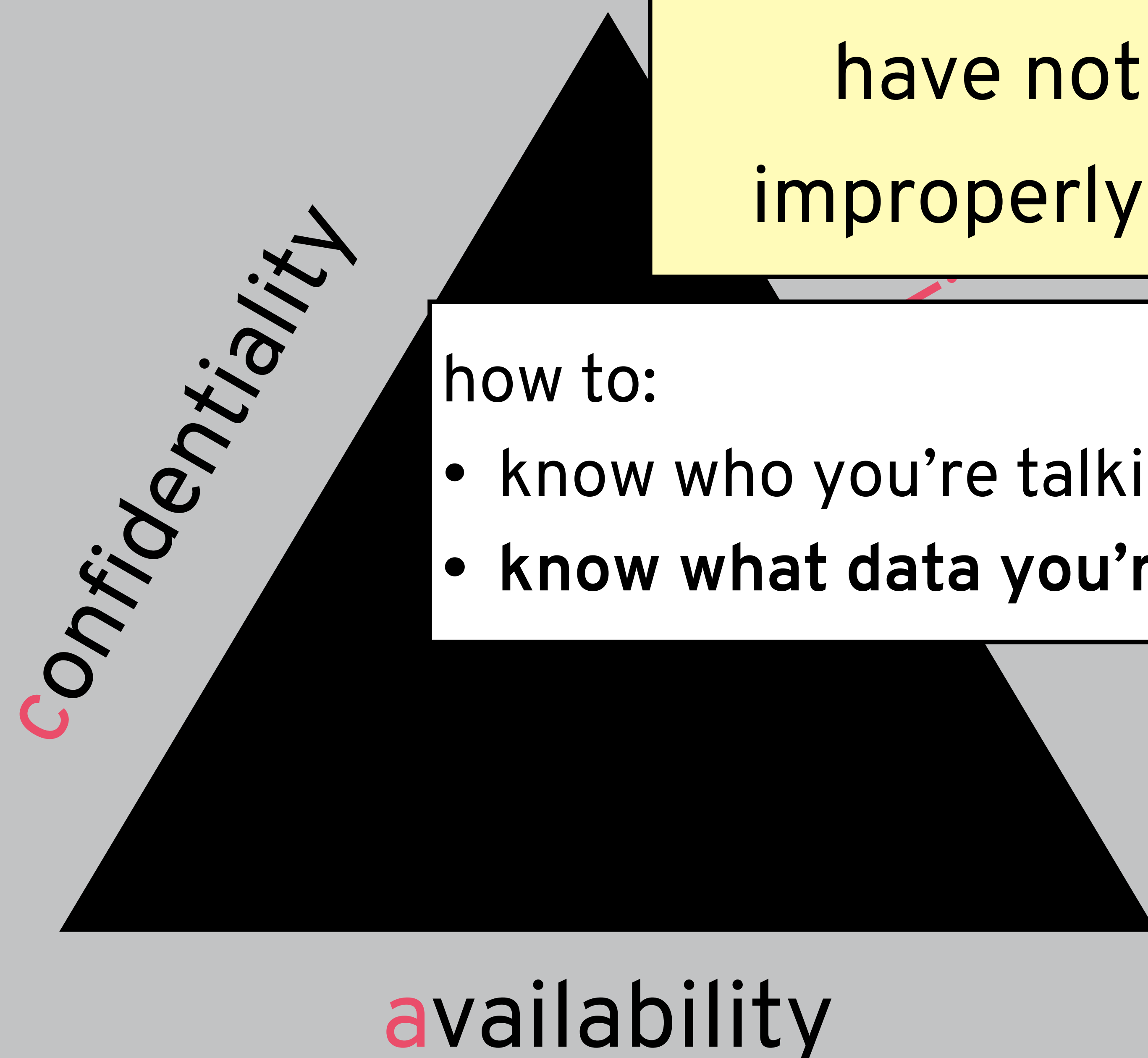
SECURITY (COMP0141): HASH FUNCTIONS

INTEGRITY

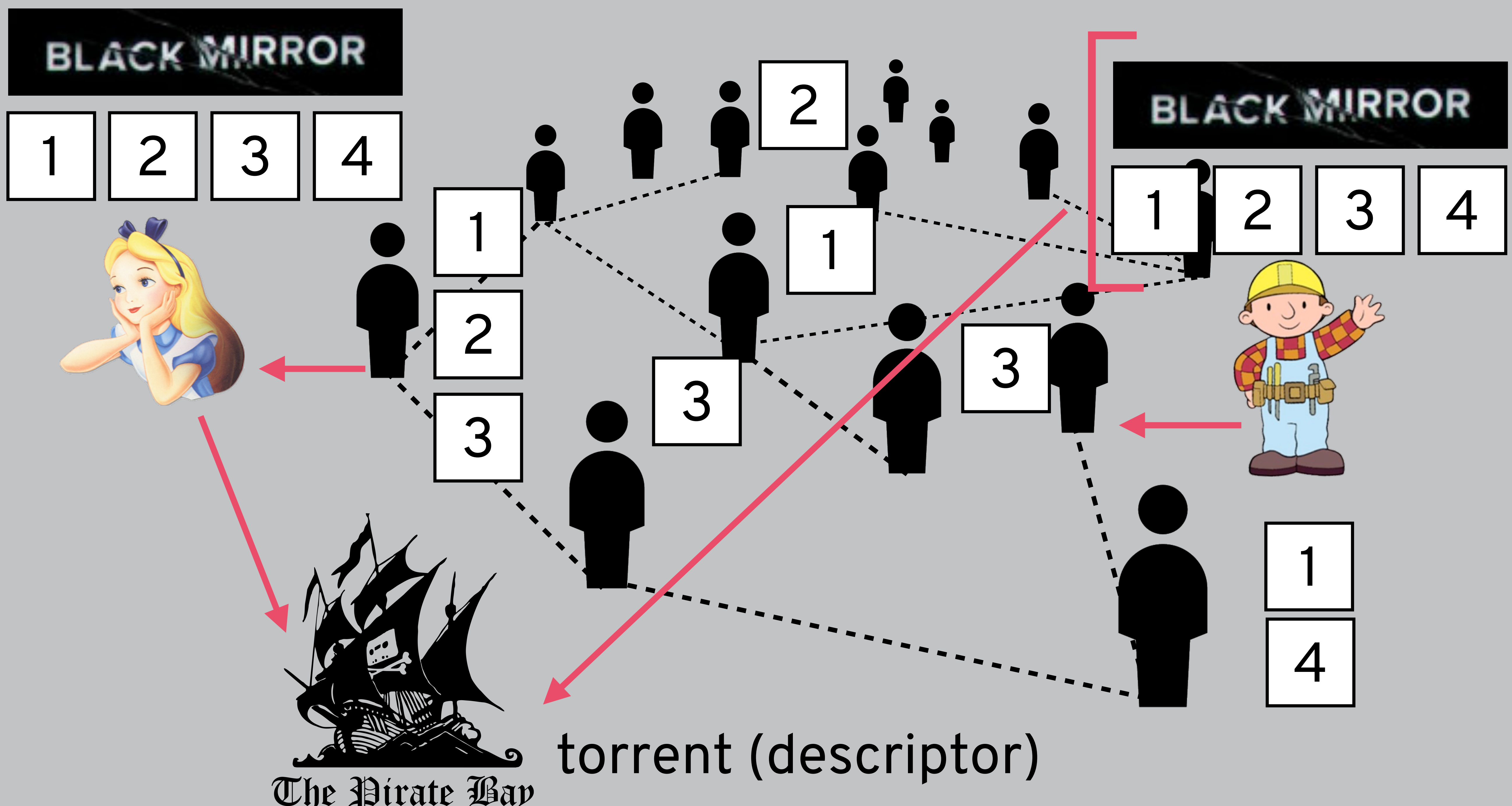


> “system and data have not been improperly altered”

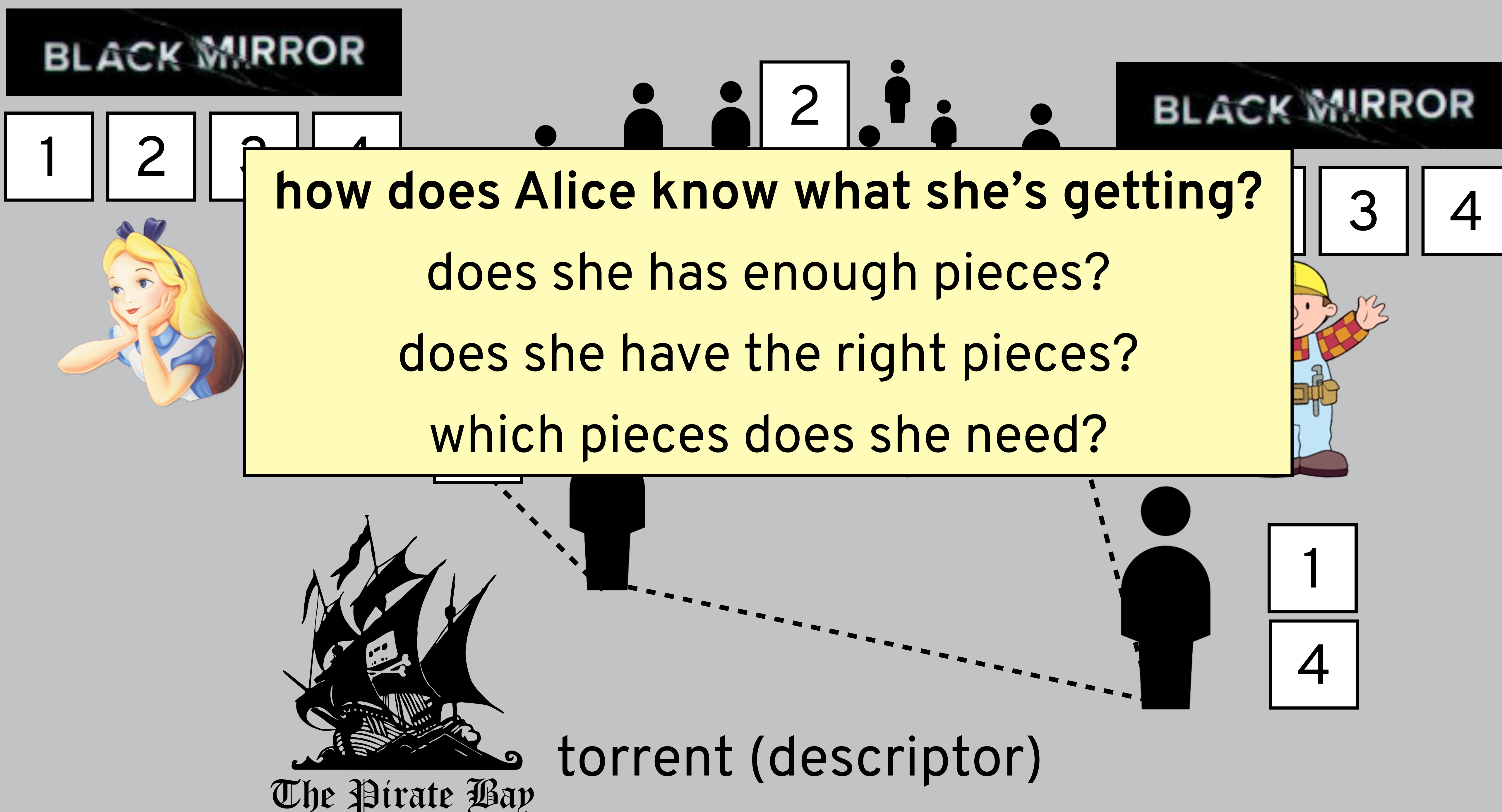
how to:

- know who you’re talking to?
- **know what data you’re getting?**

BITTORRENT


BLACK MIRROR
1
2
3
4
1
2
3
2
1
3
3
BLACK MIRROR
1
2
3
4
1
4
The Pirate Bay
torrent (descriptor)

BITTORRENT



how does Alice know what she's getting?

does she has enough pieces?

does she have the right pieces?

which pieces does she need?

torrent (descriptor)

BITTORRENT

this might get corrupted

1

these are big

BLACK MIRROR

1 2 3 4

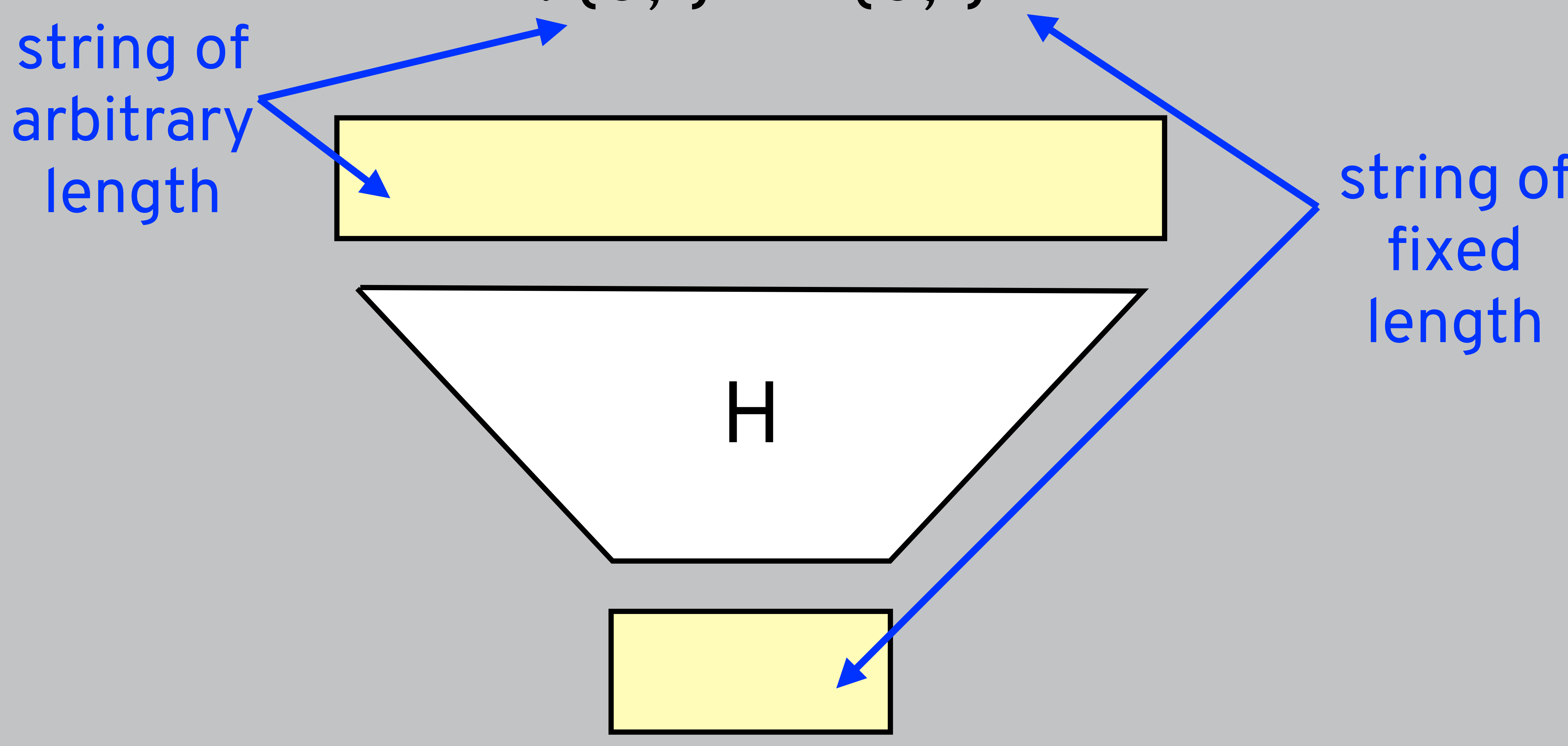
this needs to be small
but unique to the file

The Pirate Bay

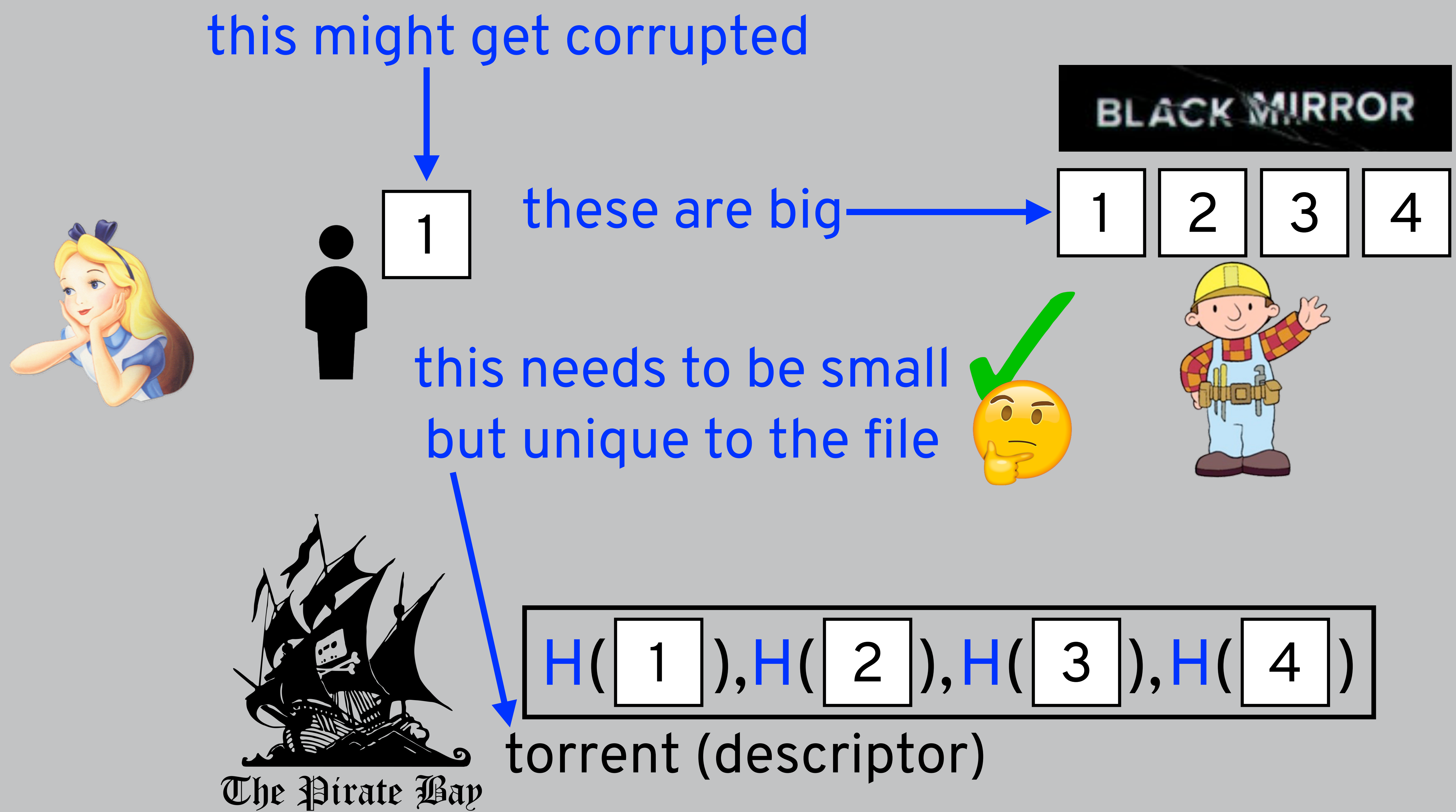
torrent (descriptor)

HASH FUNCTION

$$H: \{0,1\}^* \rightarrow \{0,1\}^k$$



BITTORRENT


this might get corrupted
1
these are big
BLACK MIRROR
1 2 3 4
this needs to be small
but unique to the file
H(1),H(2),H(3),H(4)
torrent (descriptor)
The Pirate Bay

UNIFORMITY

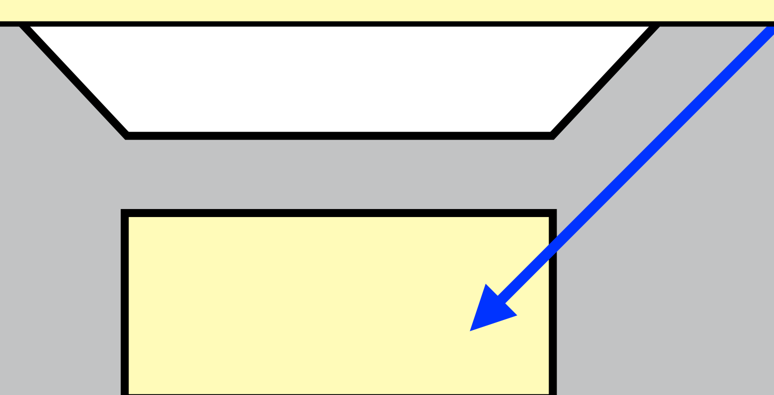
$$H: \{0,1\}^* \rightarrow \{0,1\}^k$$

stri
arbi
len

uniformity: even small changes in input yield big changes in output

uniqueness: given $h = H(m)$, should be very low chance of collision (m_2 s.t. $H(m_2) = h$)

string of fixed length



CRYPTOGRAPHIC HASHES

SHA256 hashes of…

sarah

28d628a681884cbfe83875d74ae6d9e9b4f2f211b73427ab3e83c3937d0fd028

sarah1

a2b2a43003a3e63e4c50ffb2b68d2d4d55a6cd1b8627e3e3601e984e2251ee7f

sarah12

f3bd2f4bf7e713611c5e6854a74e83c681ec9e6754ab65e63a3ce760e7c22770

sarah123

7b2935a21b68f3a6361118b2024f5547bfe9fdcc80445a4afbf62ea231a6496b

BITTORRENT

this might get corrupted

1

these are big

BLACK MIRROR

1 2 3 4

this needs to be small

but unique to the file

The Pirate Bay

H(1),H(2),H(3),H(4)

torrent (descriptor)

CRYPTOGRAPHIC HASH FUNCTION

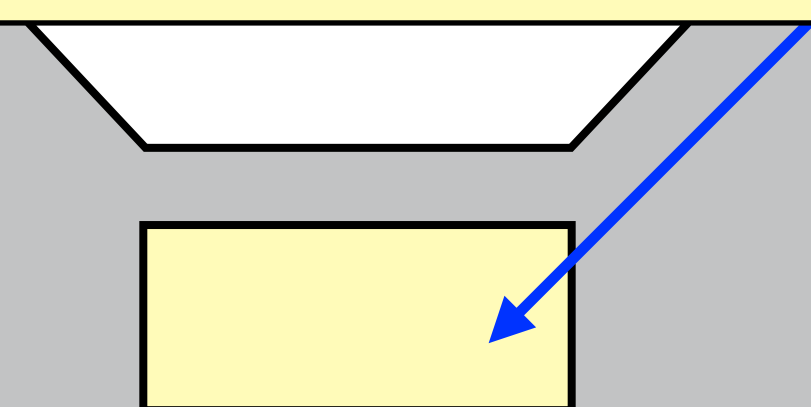
$$H: \{0,1\}^* \to \{0,1\}^k$$

strin
arbit
len

pre-image resistance: given h, hard to find m such that H(m) = h

collision resistance: hard to find x and y such that x≠y but H(x) = H(y)

tring of
fixed
length



BITTORRENT

this might get corrupted

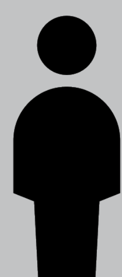
(by uniqueness)

(by pre-image resistance)

(by collision resistance)

BLACK MIRROR

1	2	3	4



1

$h_1 = H(\boxed{1})$? h_3 h_4

The Pirate Bay

torrent (descriptor)

HASH FUNCTIONS

Two main security properties:

- **Pre-image resistance:** given H(x) it's hard to find x
- **Collision resistance:** it's hard to find x and y so that x ≠ y but H(x) = H(y)

COLLISION ATTACK

How quickly can we find a collision $x_1 \neq x_2$ such that $H(x_1) = H(x_2)$?

BIRTHDAY PARADOX

Consider a class of N students with random birthdays (meaning birthdays follow a **uniform distribution** over the days of the year)

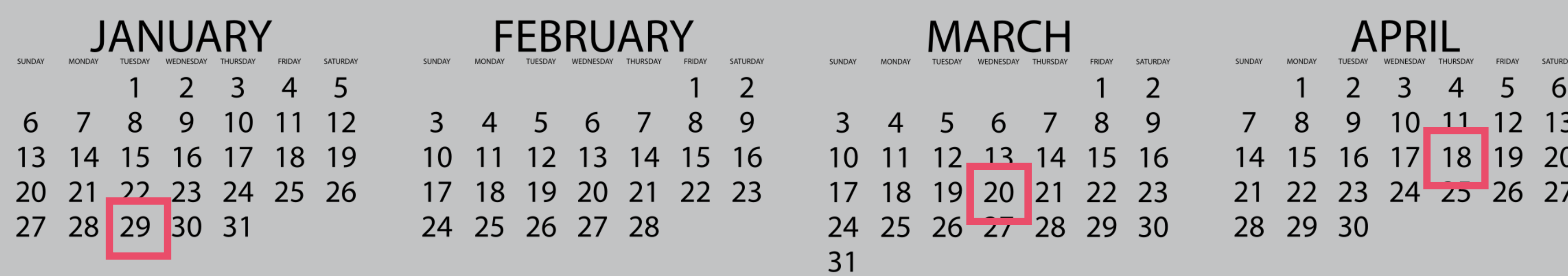
How large does N need to be before there is more than a 50% chance of having two students with the same birthday?

BIRTHDAY PARADOX

$P[A]$ = probability that two people have the same birthday

$P[\bar{A}]$ = probability that no two people have the same birthday

$P[A] = 1 - P[\bar{A}]$



Event 1 (E1) = student 1 has a birthday ($P[E1] = 1$)

Event 2 (E2) = student 2 has a birthday different from student 1 ($P[E2] = (365 - 1) / 365) = 364/365$)

...

Event N (EN) = student N has a birthday different from all previous students ($P[EN] = (365 - N + 1) / 365$)

$P[\bar{A}] = P[E1]...P[EN] = (1 / 365)^N * 365 * 364 * ... * (365 - N + 1)$

BIRTHDAY PARADOX

Consider a class of N students with random birthdays (meaning birthdays follow a **uniform distribution** over the days of the year)

How large does N need to be before there is more than 50% chance of having two students with the same birthday?

Answer: $\sqrt{365} \approx 23$

COLLISION ATTACK

How quickly can we find a collision $x_1 \neq x_2$ such that $H(x_1) = H(x_2)$?

Pick different $x_1, \ldots, x_{\sqrt{N}}$ (where $N = 2^n$ for $H: \{0,1\}^* \rightarrow \{0,1\}^n$)

Compute $y_1 = H(x_1), \ldots, y_{\sqrt{N}} = H(x_{\sqrt{N}})$ and look for a collision

This has almost a 40% chance of finding a collision!

Memory cost: $3n*2^{n/2}$ bits

Computational cost: $2^{n/2}$ hash evaluations

COLLISION ATTACKS IN PRACTICE

	n	birthday	shortcut
MD4	128	64	2
MD5	160	80	21
RIPEMD	128	64	18
RIPEMD160	160	80	
SHA-0	160	80	34
SHA-1	160	80	(51)
SHA-256	256	128	
SHA-3	256	128	

HASH FUNCTIONS

Two main security properties:

- **Pre-image resistance:** given H(x) it's hard to find x
- **Collision resistance:** it's hard to find x and y so that x ≠ y but H(x) = H(y)

Applications:

- File checksum
- MACs
- Digital signatures
- Commitments
- Blockchains
- Virus scanning (next week)
- Password storage (Week 7)
- ...and many more!

CRYPTOGRAPHIC PRIMITIVES

	setup?	confidentiality/ integrity?	fast?
SE	yes	confidentiality	yes
PKE	no*	confidentiality	no
digital signature	no*	integrity	no
MAC	yes	integrity	yes
OWF	no	confidentiality*	no
hash function	no	integrity	yes
AE	yes	both	yes

STRIDE

S T R I D E

Spoofing

integrity

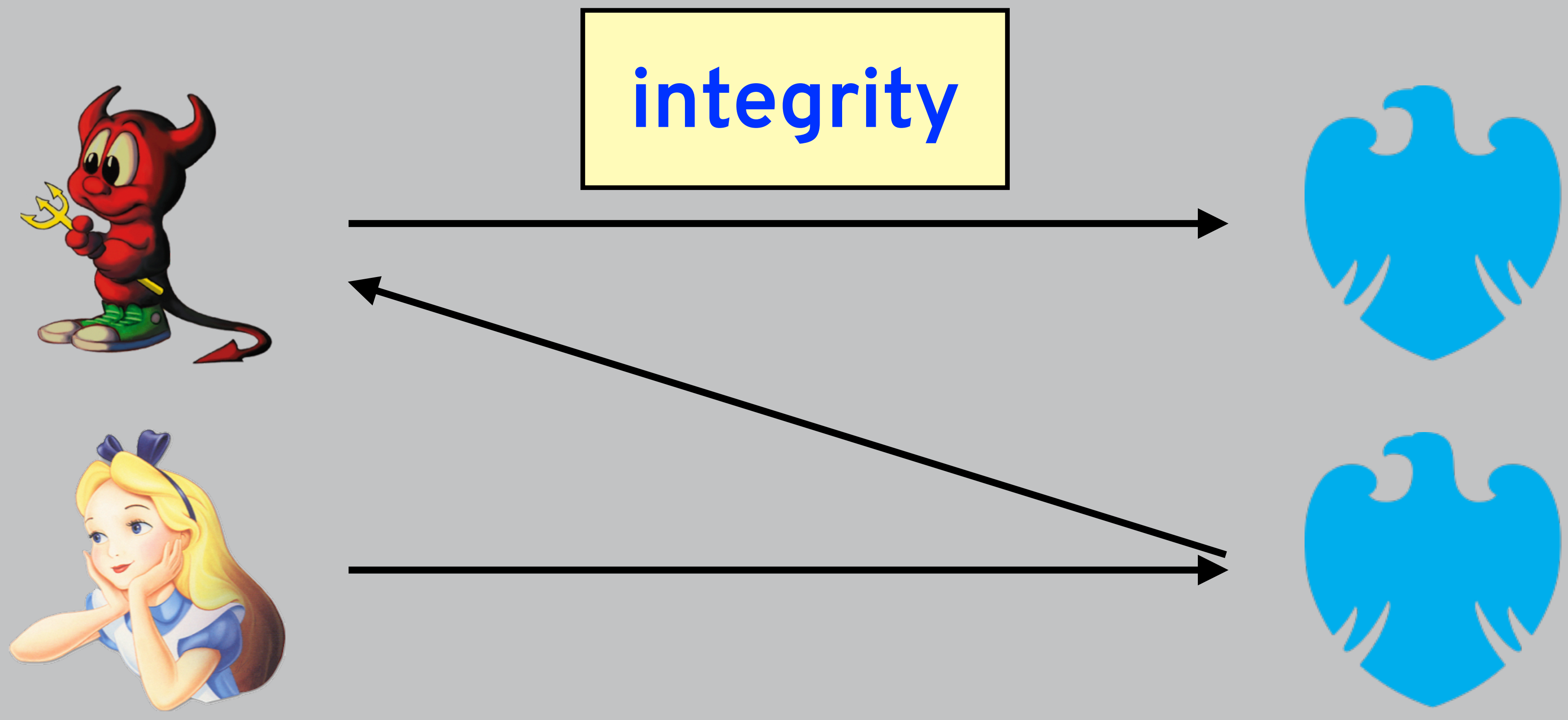
“It’s me, Alice!”

STRIDE

S T R I D E

Spoofing

Tampering



STRIDE

S T R I D E

Spoofing

Tampering

Repudiation

“It’ e!”

integrity

“It wasn’t me!”

STRIDE

S T R I D E

Spoofing

Tampering

Repudiation

Information disclosure

confidentiality

account info of users ←

STRIDE

S T R I D E

Spoofing

Tampering

Repudiation

Information disclosure

Denial of service

availability

STRIDE

S T R I D E

Spoofing
Tampering
Repudiation
Information disclosure
Denial of service
Elevation of privilege

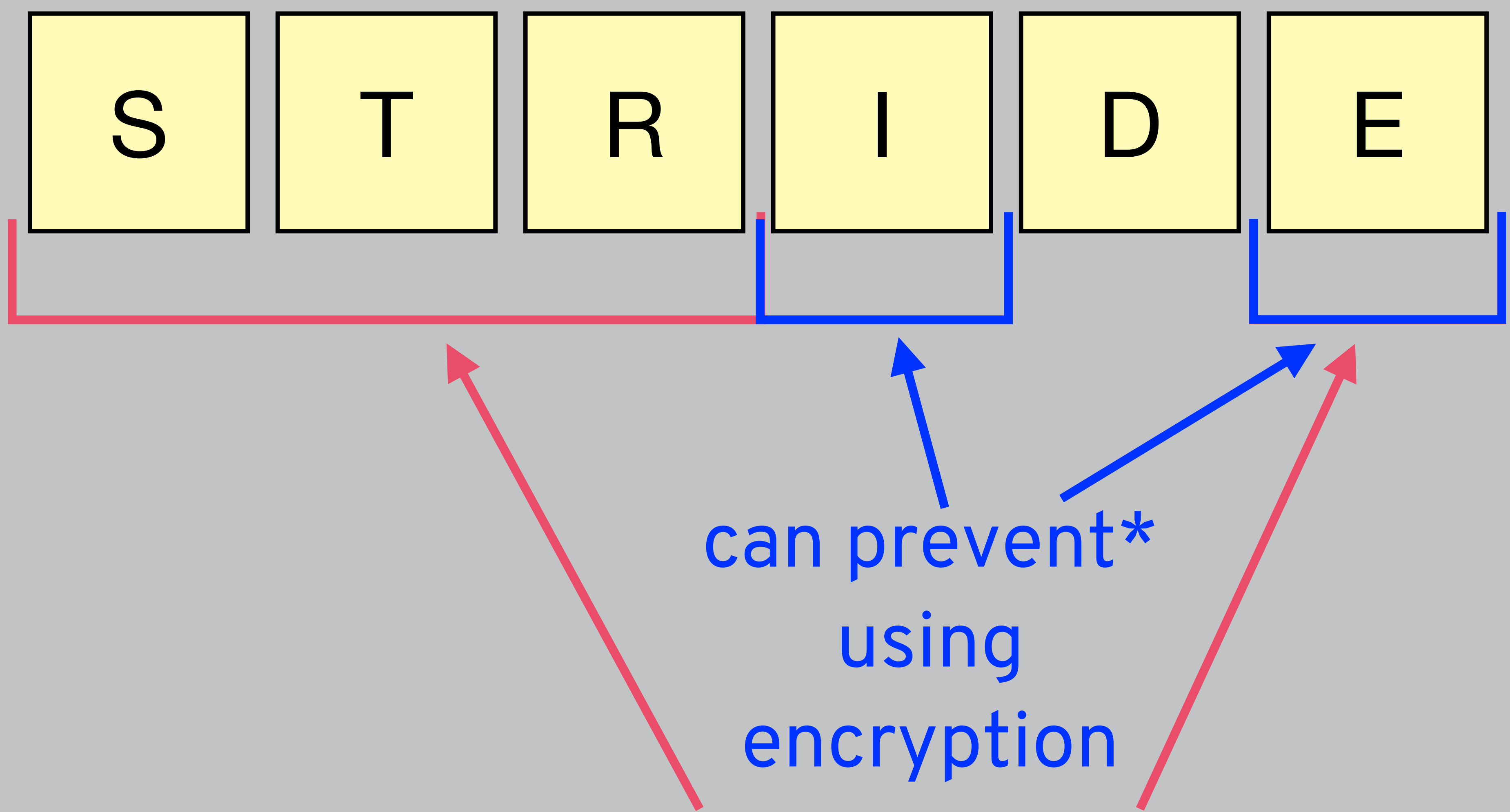
confidentiality

(also integrity)

“It’s me!”

account info of users

STRIDE



can prevent* using encryption

can prevent* using signatures and hash functions

**there is no silver bullet here!

UNANSWERED QUESTIONS

How do I build a block cipher?

How do I build a stream cipher?

How do I build a hash function?

How do I implement any of these?

On the basis of this module: do not!

Use only standardised modes of operation and protocols, and code with only well-established and audited libraries