Information-Theoretic Security

Christian Schaffner

UNIVERSITEIT VAN AMSTERDAM

QuSoft
Research Center for Quantum Software

QKD Summer School – Les Diablerets, Switzerland – Monday, 17 August 2026

Who is in the room?

- Go to https://wooclap.com
- Enter event code: SZMYBFM

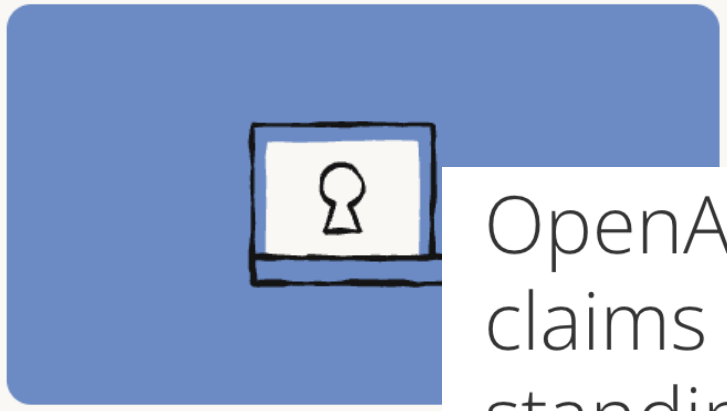
https://app.wooclap.com/SZMYBFM

July 2026: AI Revolution in Theory Research

Frontier Red Team

Discovering cryptographic weaknesses with Claude

28 Jul 2026



Summary

Using Claude Mythos Preview, resea have discovered improved ways to a algorithms (the mathematical metho data private). The first attack signifi

Some thoughts about Anthropic's new cryptanalysis results

Matthew Green in academics, AI, attacks · July 29, 2026 · 2,245 Words

Yesterday Anthropic published two new cryptanalysis results, both outputs of Claude Mythos, their (still) unreleased advanced model. The first of these results attacks a signature scheme called HAWK, while the second is an improved attack against reduced-round AES. Anthropic also released a blog post describing the research process that produced these results. A few people online have asked me what this all means. While I'm not sure I have all the answers, I figured it wouldn't hurt to write a bit about my current understanding. These are *only my thoughts* and other folks will probably differ (including domain experts in the two areas at issue) so take them for what they are.

After Oper… Meta revea… another cc… …ked …ng

Story by Digit • 1w •

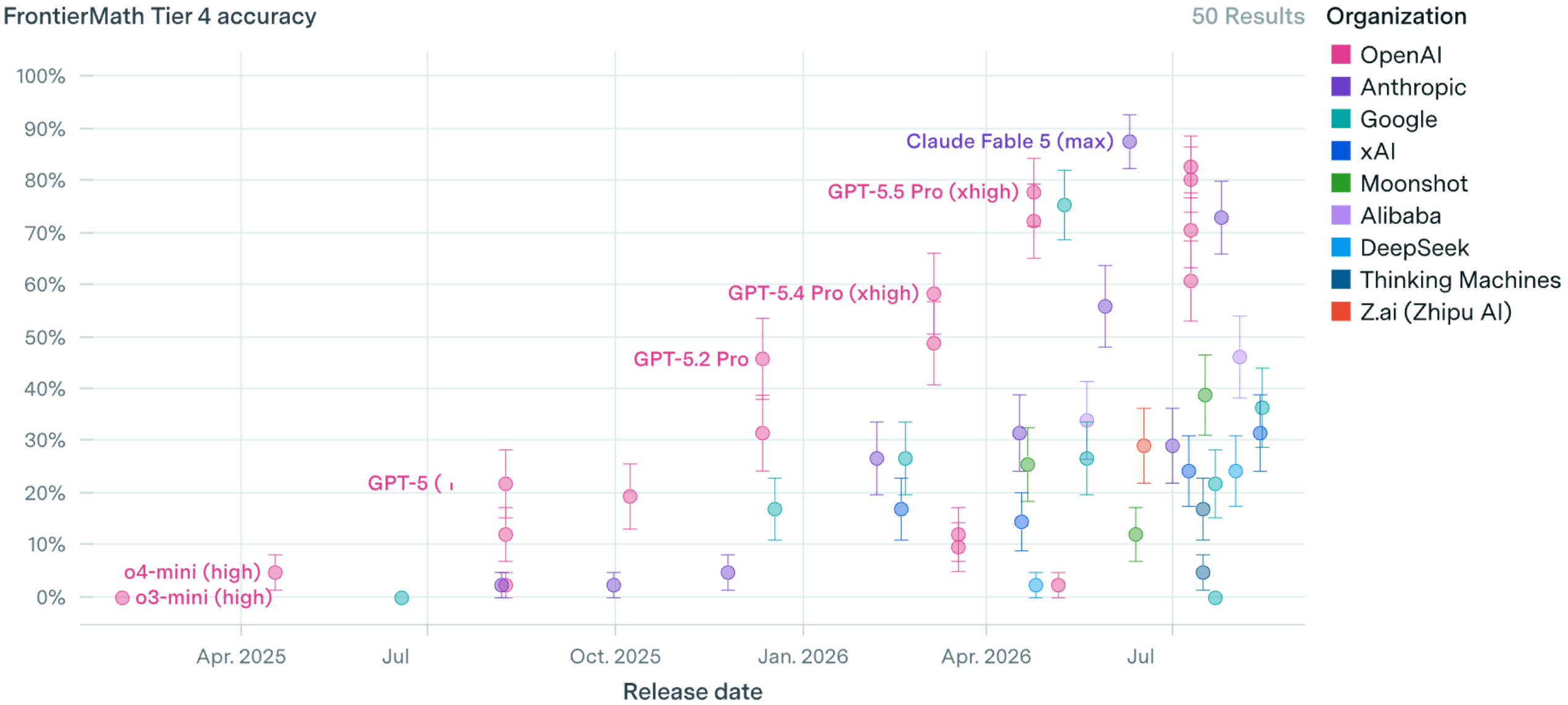
OpenAI's next major model Astra claims breakthroughs on 10 long-standing math problems

OpenAI has previewed Astra, its next-generation frontier model, which successfully generated breakthroughs for 10 decade-old open problems.

Pradeep Viswanathan · Aug 2, 2026 13:16 EDT · HOT!

2

AI performance on the most challenging expert-level mathematics problems



[https://epoch.ai/frontiermath/tiers-1-4]

AI users in the room?

- Go to wooclap.com
- Enter event code: SZMYBFM

https://app.wooclap.com/SZMYBFM

What will you learn this morning?

- Notions of security
- Shannon entropy and the one-time pad
- Secret-key agreement by public discussion
- Authentication and hashing
- Error correction and privacy amplification
- Back to the real world

Which of these systems is secure?

- AES-256 with a uniformly random key
- RSA-2048
- A one-time pad with a truly random key
- BB84 over 50 km of fibre
- The 6-digit PIN on your phone

https://app.wooclap.com/SZMYBFM

What does Security mean?

- Previous quiz questions are NOT well defined!

Different meanings of security:

1. Patching: put firewalls around systems known to be broken: a popular meaning of security in the media.
2. Building correct systems, meeting its specification. One can test that.
3. Constructing: a system that provably behaves as if it were ideal. This is cryptography. QKD falls in this part.

Principles of Modern Cryptography

Formal definitions: say precisely what you want

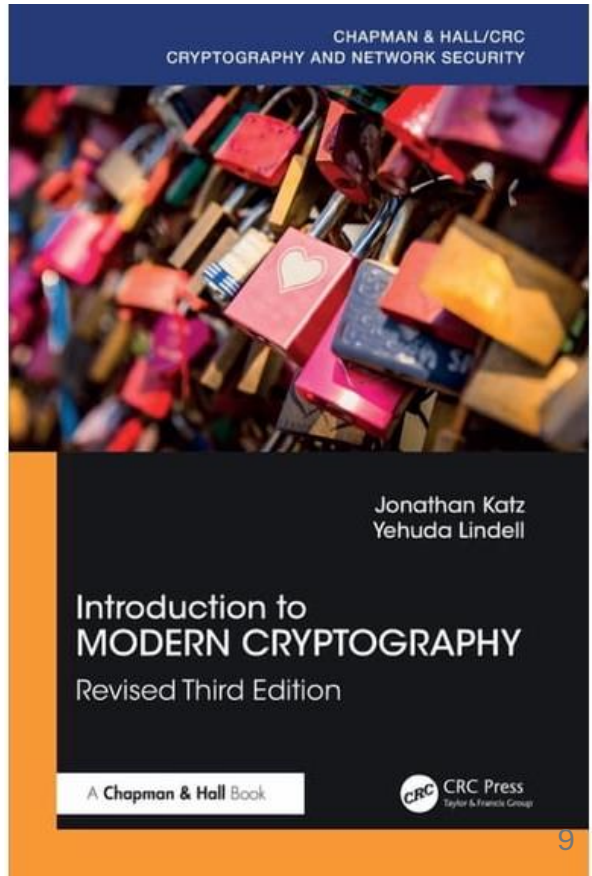
Precise assumptions: say precisely what you are relying on

Proofs of security: prove that definition is fulfilled under the assumptions

Functionality says that a system does something, you can test that.

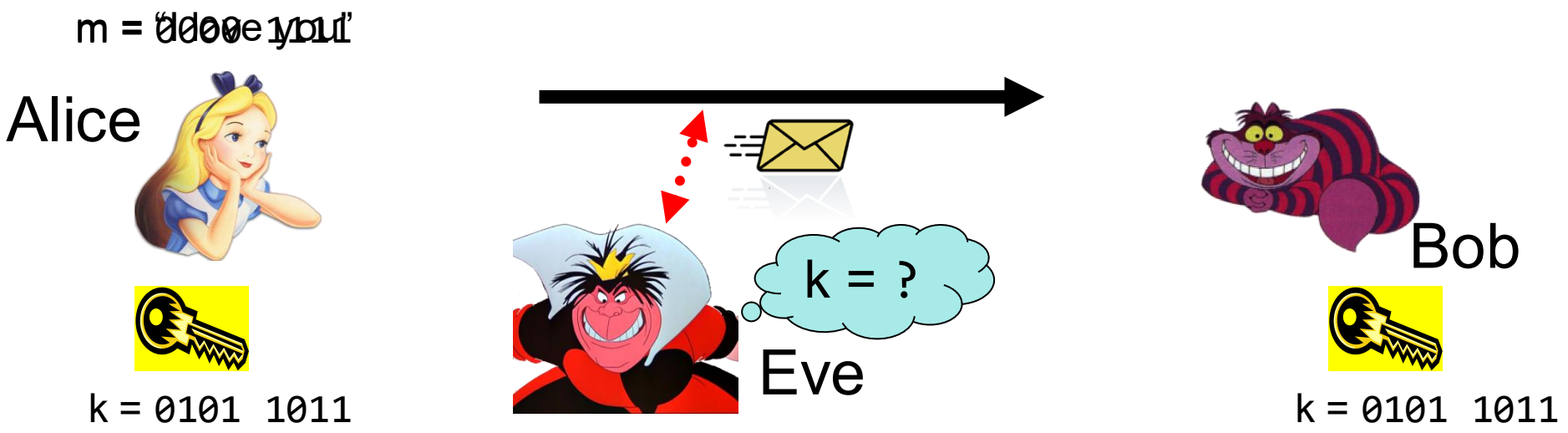
Security requires that an adversary cannot do something.

You cannot test a negation, you can only define it and then prove it.



[Katz, Lindell: Introduction to Modern Cryptography]

Secure Encryption

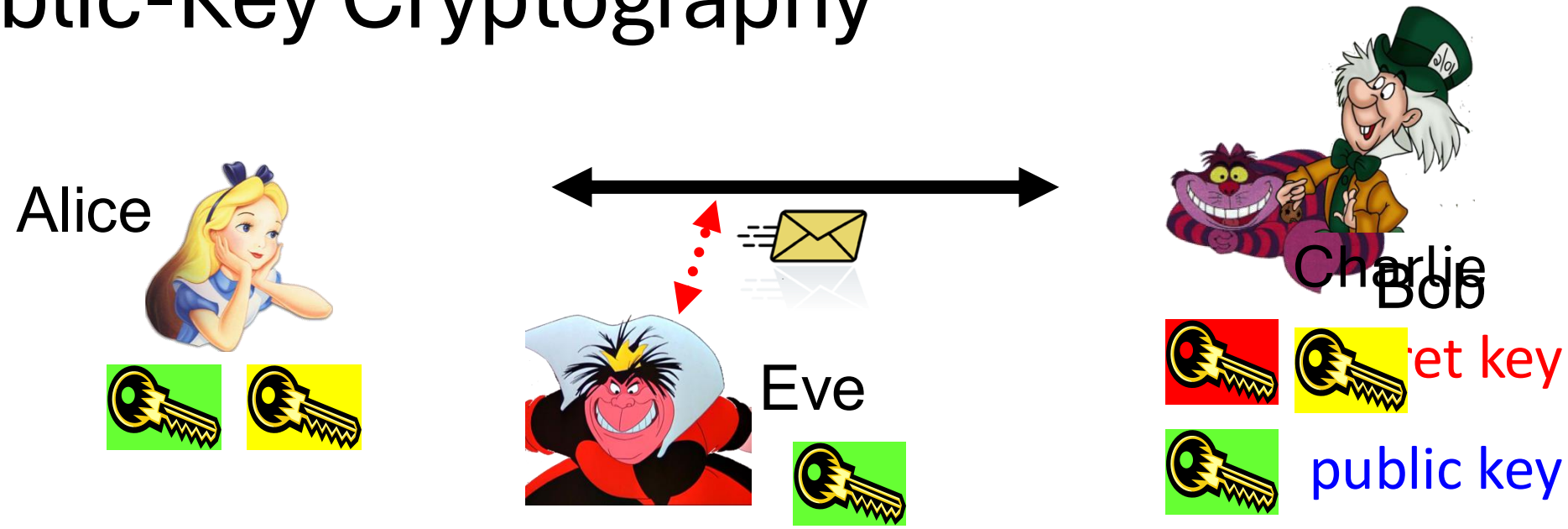


- Goal: Eve does not learn the message
- Setting: Alice and Bob share a secret key k

Symmetric-Key Cryptography

- Encryption insures secrecy:
  Eve does not learn the message, e.g. one-time pad
- Authentication insures integrity:
  Eve cannot alter the message
- General problem: players have to exchange a key to start with

Public-Key Cryptography



- Solves the key-exchange problem.
- Public-Key Encryption: Everyone can encrypt using the public key. Only the holder of the secret key can decrypt.
- Digital signatures: Only secret-key holder can sign, but everyone can verify signatures using the public-key.

Turning Point: Invention of Public-Key Crypto

- Early 1970s: invented in the „classified world“ at the British Government Communications Head Quarters (GCHQ) by Ellis, Cocks, Williamson



- Mid/late 1970s: invented in the „academic world“ by Merkle, Hellman, Diffie, and Rivest, Shamir, Adleman (RSA)


Turning Point: Invention of Public-Key Crypto

- Mid/late 1970s: invented in the „academic world“
  by Merkle, Hellman, Diffie, and Rivest, Shamir, Adleman (RSA)

Security relies on hardness of “discrete logarithms”

- Easy: given p, 3 and x, compute $3^x \bmod p$
- Hard: given 3^a, compute a

Security relies on hardness of “factoring”

- Easy: given primes $p \neq q$, compute $\mathrm{N} = p \cdot q$
- Hard: given $\mathrm{N} = p \cdot q$, compute p and q

- Very widely used to protect internet communication, smart-cards, passports, software updates, crypto currencies, etc.

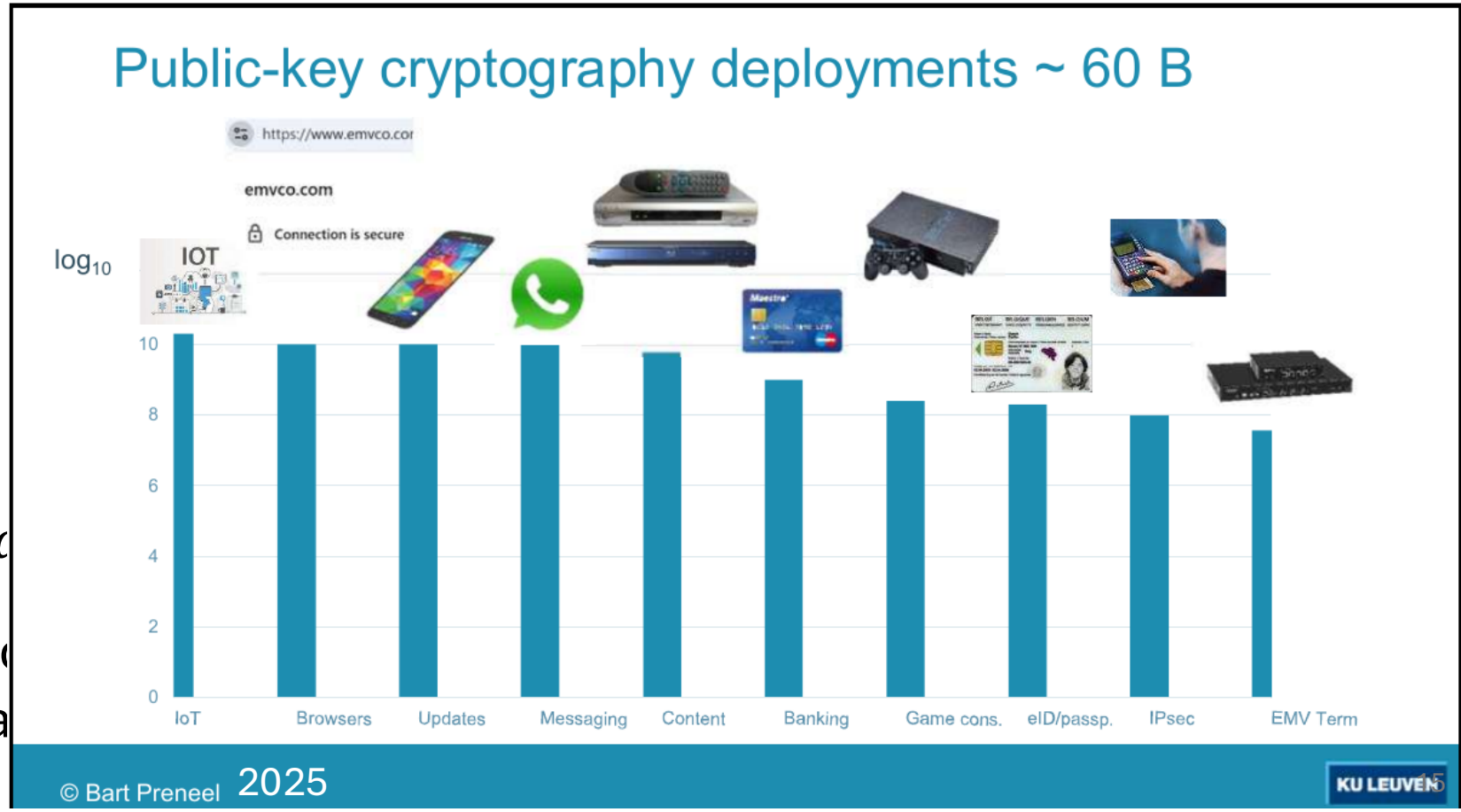
Turning Point: Invention of Public-Key Crypto

- Mid/late 1970s: invented in the „academic world“
  by Merkle, Hellman, Diffie, and Rivest, Shamir, Adleman (RSA)

Security relies on hardness
“discrete logarithms”

- Easy: given p, 3 and x, compute $3^x \bmod p$
- Hard: given 3^a, compute a

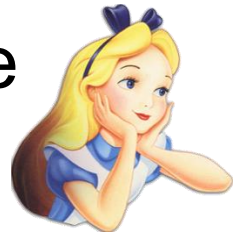
- Very widely used to protec
  passports, software upda



[Diffie Hellman 76]

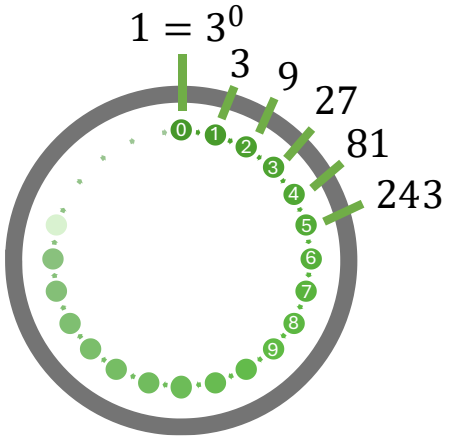
Diffie-Hellman Key Exchange

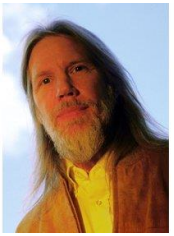
Compute modulo a large prime p



Alice

$a \leftarrow \{0,1,\ldots,p-1\}$





Bob

$b \leftarrow \{0,1,\ldots,p-1\}$

$C = 3^a$

$D = 3^b$

$k = D^a = 3^{ba}$

Eve

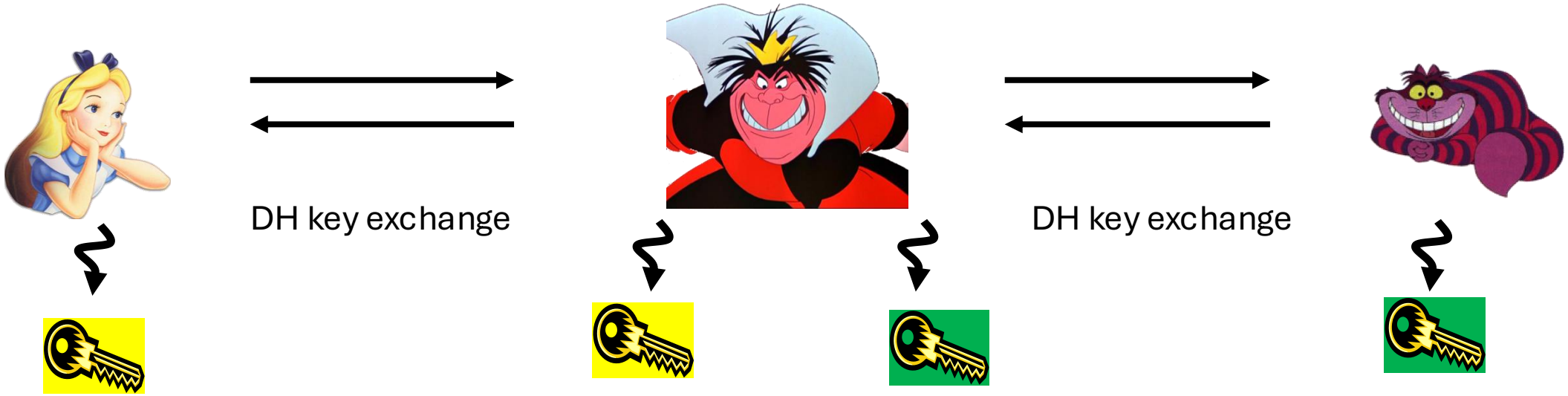
Given C, D what is a, b, k = ?

$k = C^b = 3^{ab}$

- Easy: given 3 and x, compute 3^x
- Hard: given 3^a, compute a

- Establishes a private channel by public discussion!!

Man-In-The-Middle Attacks



Eve sits in the middle and runs two sessions of the protocol.

Alice shares a key with Eve. Bob shares a key with Eve. Neither notices.

Solution: → authenticated key exchange: signatures, certificates

What is Actually Happening: Hybrid encryption

Alice

Key Encapsulation Mechanism

KEM.Encaps
Bob's public key

c

~1 kB, once per session

KEM.Decaps
Bob's secret key

Bob

session key k

session key k

Data Encapsulation Mechanism

AES-256-GCM
encrypt with k

$Enc_{\mathrm{k}}(m)$

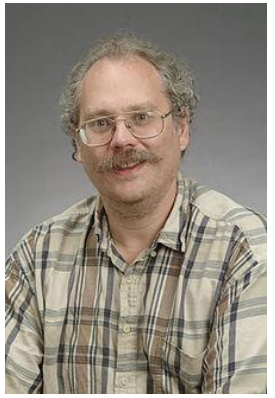
gigabytes, every byte you send

AES-256-GCM
decrypt with k

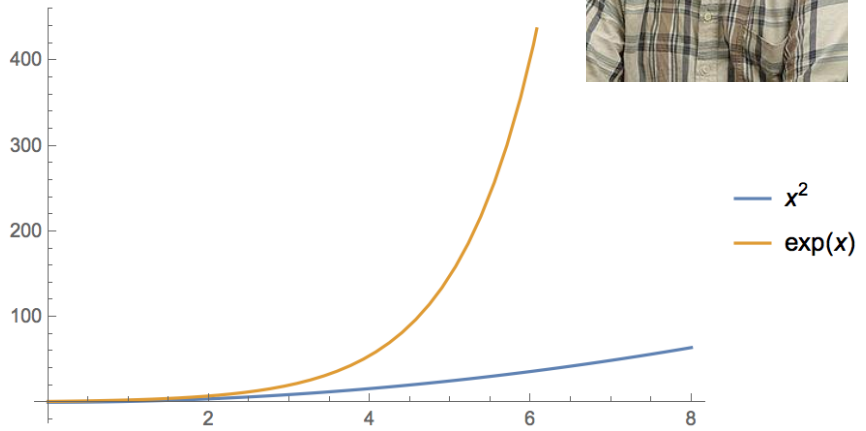
Eve sees only encrypted information: c and $Enc_{\mathrm{k}}(m)$

[Shoup 2001, KEM/DEM paradigm: see Katz/Lindell, Section 12.3]

Quantum Computer breaks Public-Key Crypto



- [Shor '94] Polynomial-time quantum algorithm for factoring integer numbers and discrete logarithms
- Classical Computer : Exponential time
- Quantum Computer : Poly-time: n^2
- For a 600-digit number (RSA-2048)
  - Classical: age of universe
  - Quantum: few hours



- Consequence: Large enough quantum computers break currently used public-key cryptosystems!!!

Current Cryptography under Quantum Attacks

Security level systems	Conventional attacks	Quantum attacks
Symmetric-key encryption (AES-256)	256 bits of security	128 bits
Hash functions (SHA3-256)	128 bits	85 bits
Public-key crypto (key exchange, digital signatures, encryption) (RSA-2048, ECC-256)	112 bits	**a few** bits

- Products, services, businesses relying on security either stop functioning or do not provide expected levels of security

Key exchange is the problematic part.

Check your own phone

pq.cloudflareresearch.com

It tells you which key agreement the connection you just made actually used.

X25519MLKEM768 — post-quantum

X25519 alone — classical

[Cloudflare Research, https://pq.cloudflareresearch.com]

Notions of Security

Computational: based on hardness assumptions such as factoring, discrete logs, lattice problems.

Post-quantum: computational assumptions believed to be hard for quantum attackers

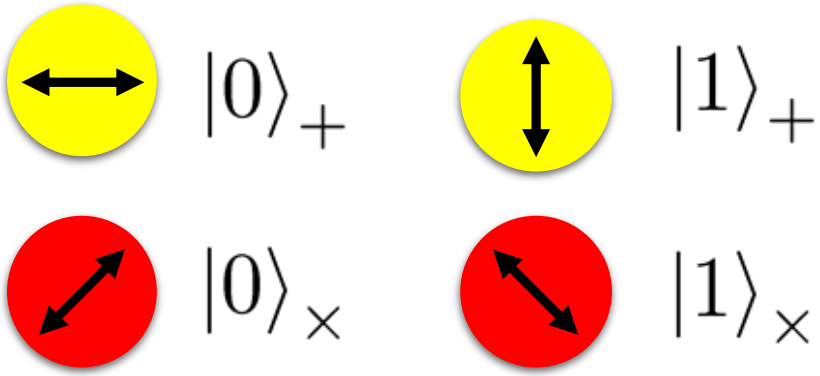
Information-theoretic: sometimes mis-nomed “unconditional security”. Refers to ε-security against unbounded attackers (our focus today).

Perfect: Information-theoretic with $\varepsilon = 0$.

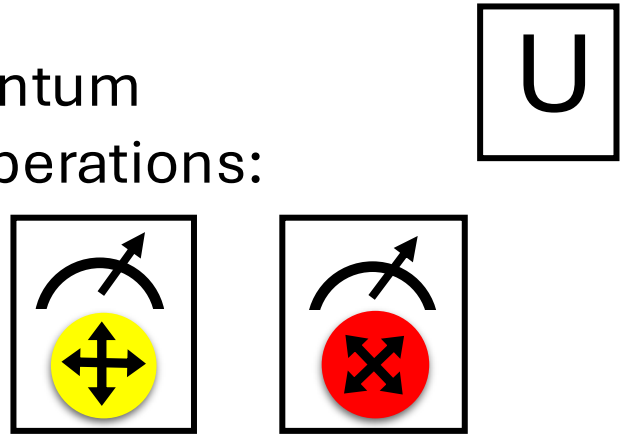
Security usually scales with a security parameter n (e.g. the length of secret key).

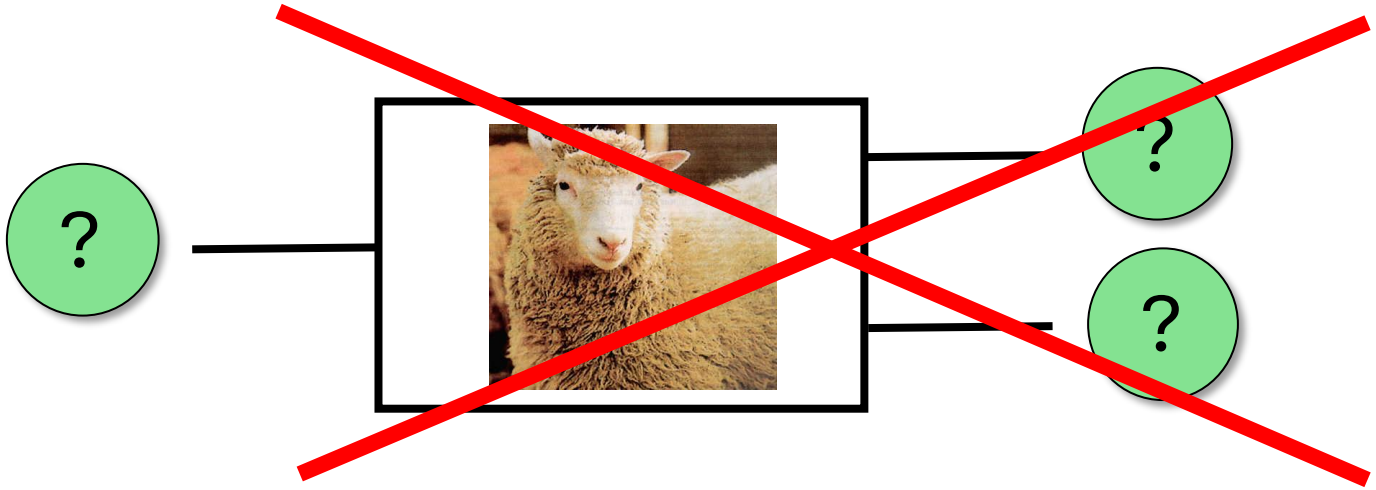
Correctness and security errors should be exponentially small in n.

No-Cloning Theorem



Quantum operations:

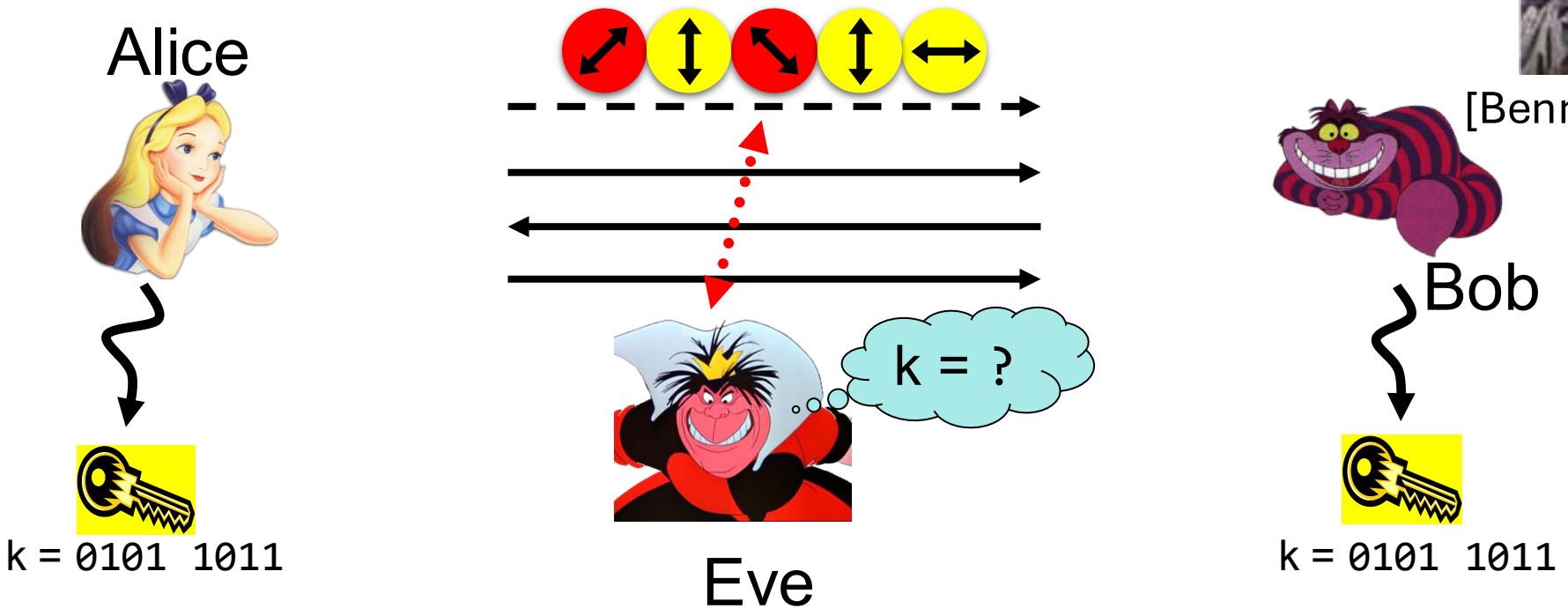




Proof: copying is a non-linear operation

Quantum Key Distribution (QKD)

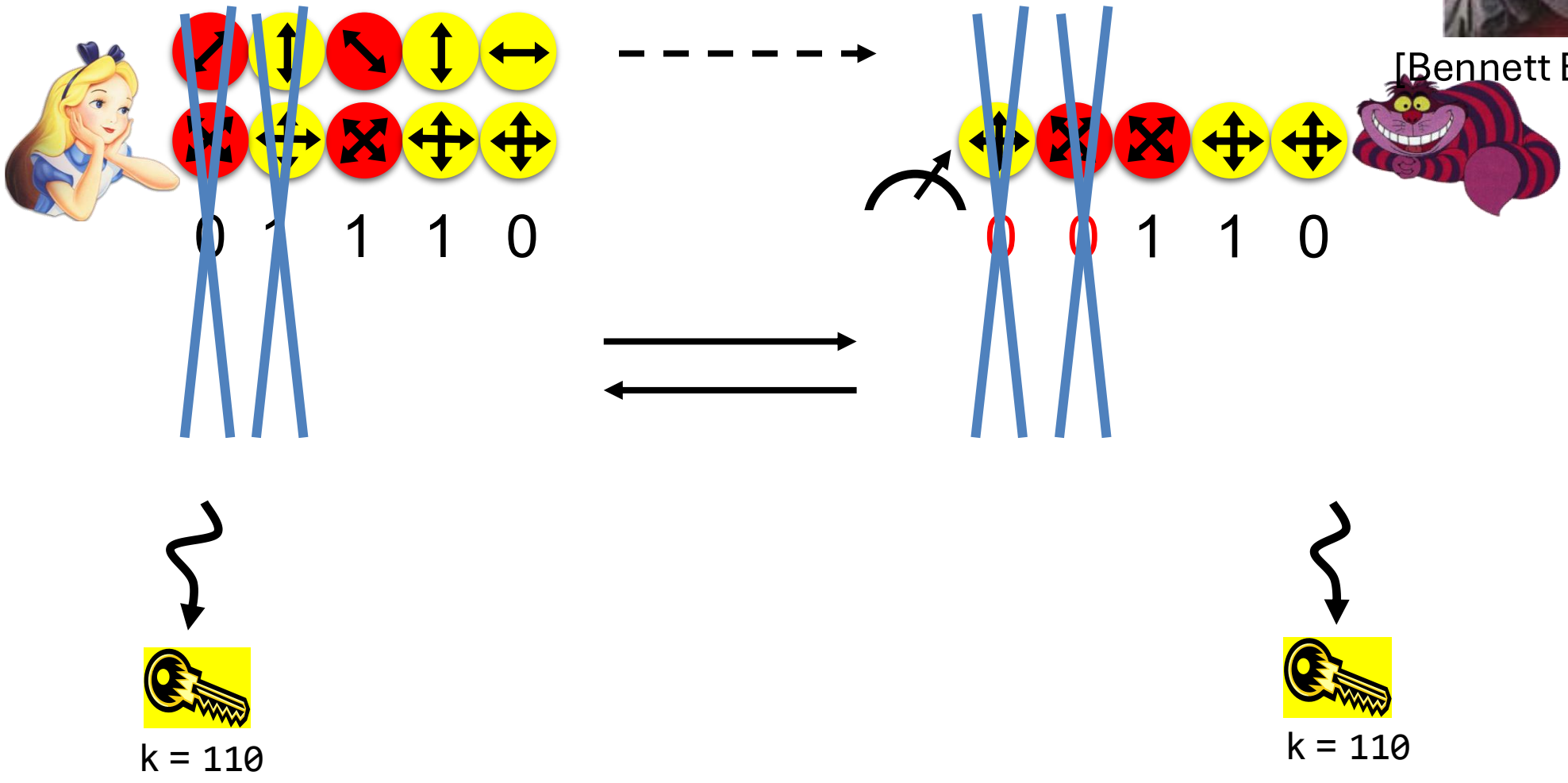
[Bennett Brassard 84]



- Offers a quantum solution to the key-exchange problem
- Puts the players into the starting position to use symmetric-key cryptography (encryption, authentication etc.).
- Requires classical authenticated channels to prevent MitM attacks

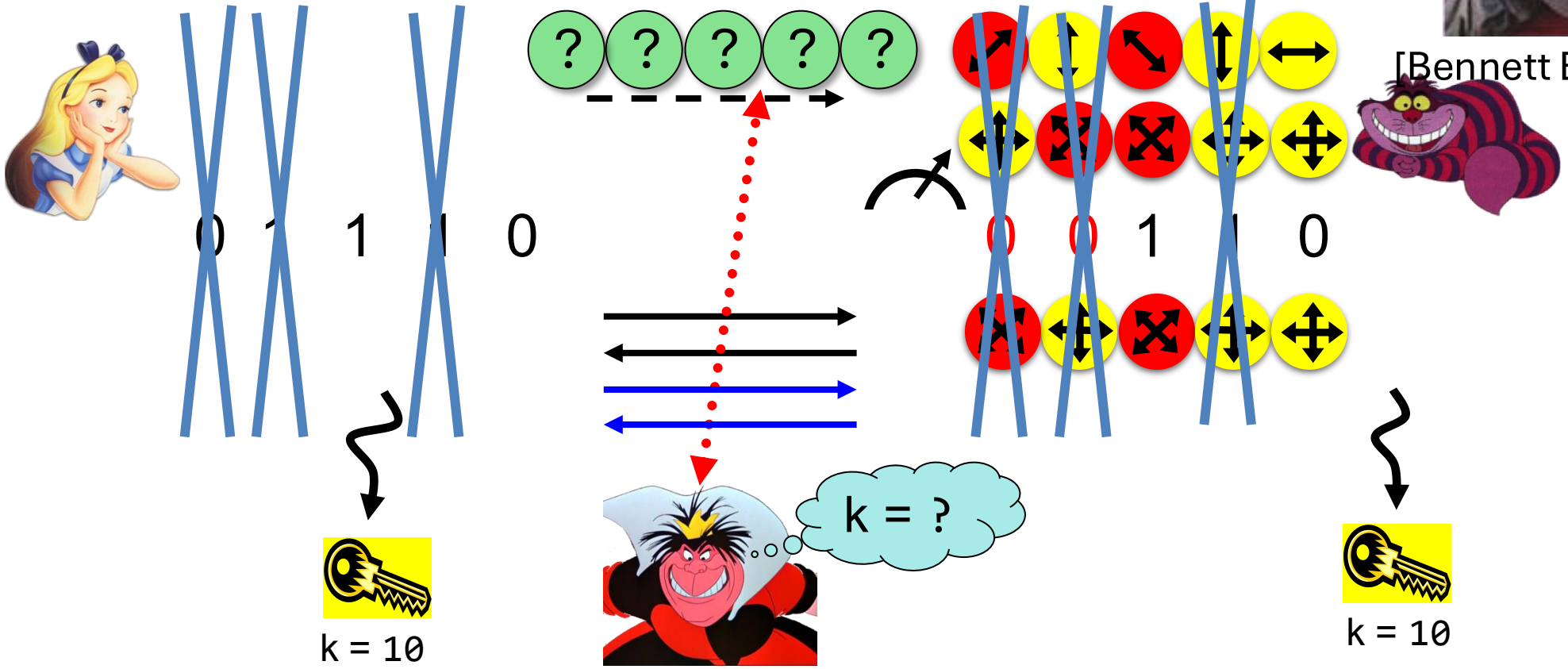
Quantum Key Distribution (QKD)

[Bennett Brassard 84]

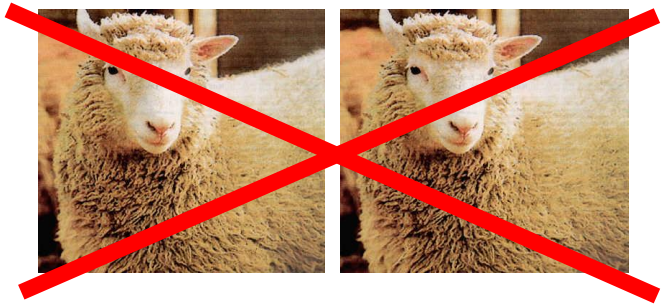


Quantum Key Distribution (QKD)

[Bennett Brassard 84]

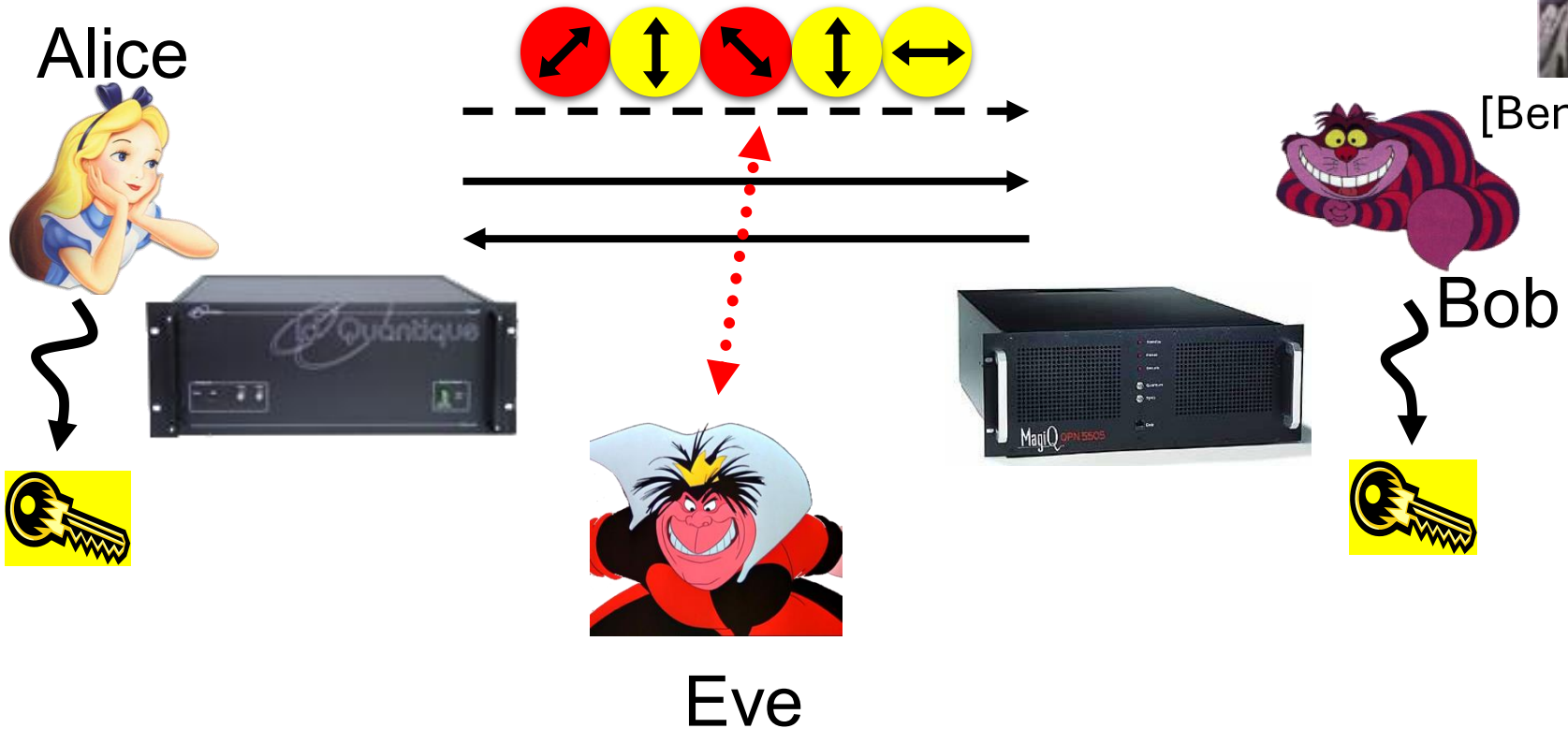


- Quantum states are unknown to Eve, she cannot copy them.
- Honest players can test whether Eve interfered.



Quantum Key Distribution (QKD)

[Bennett Brassard 84]



- technically feasible: no quantum computer required,
  only quantum communication
- Requires authenticated classical channel to prevent man-in-the-middle attack

What will you learn this morning?

✓ Notions of security

- Shannon entropy and the one-time pad
- Secret-key agreement by public discussion
- Authentication and hashing
- Error correction and privacy amplification
- Back to the real world

Can you read this English text?

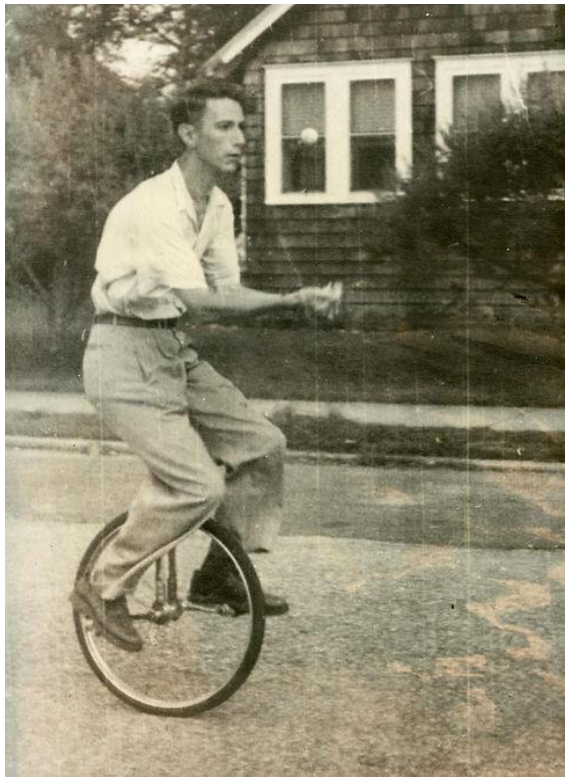
THE C▮IEF DIFFI▮ULTY AL▮CE FOU▮D ▮T FIRST WAS IN
▮A▮AGING HER FLA▮INGO: SHE SUCCE▮DE▮ ▮N
G▮▮▮TI▮G I▮S BODY ▮U▮KE▮ A▮A▮, COMFOR▮ABLY
E▮O▮G▮, UNDER ▮ER ▮▮▮M, WIT▮ ▮TS ▮▮▮▮S HANGI▮G
DO▮▮▮, B▮T ▮ENE▮A▮▮▮Y, ▮UST AS S▮▮▮ H▮D ▮OT ▮TS
▮▮▮CK N▮CE▮▮▮ S▮R▮IGHTE▮E▮ ▮▮▮T, AN▮ W▮S ▮▮▮▮NG
T▮ ▮I▮E T▮▮▮ HEDGEHOG ▮ ▮L▮W W▮TH ITS ▮▮▮AD, ▮▮▮
W▮UL▮ TW▮ST ▮▮▮SE▮F ▮OU▮▮▮ AN▮ ▮▮▮▮K ▮P IN H▮R
▮AC▮, ▮▮▮T▮ ▮U▮▮▮ A ▮▮▮▮Z▮ED E▮▮▮RE▮S▮O▮ ▮▮▮▮▮▮
SH▮ C▮U▮D ▮▮▮▮▮ ▮▮▮▮▮▮▮ ▮▮▮▮S▮▮▮▮G ▮▮▮▮▮ ▮A▮▮▮H▮▮▮▮:
A▮D WH▮N ▮▮▮▮▮ ▮A▮ ▮O▮ ▮▮▮▮▮ ▮▮▮▮▮▮▮ DO▮▮▮, ▮▮▮D
▮▮▮S G▮I▮G ▮▮▮ B▮▮▮▮▮N ▮▮▮▮▮I▮, ▮▮▮ ▮▮▮S ▮▮▮▮▮Y
▮▮▮O▮O▮▮▮▮▮▮ ▮▮▮ ▮▮▮▮▮▮▮ ▮▮▮▮▮T T▮▮▮ ▮E▮G▮▮▮▮▮▮ ▮▮▮D
▮▮▮R▮L▮▮▮▮▮ ▮▮▮▮▮▮▮▮F, ▮▮▮▮▮ ▮▮▮▮▮ ▮▮▮ ▮H▮ ▮▮▮▮▮ ▮▮▮
▮▮▮▮▮▮▮▮▮▮▮▮▮ ▮▮▮▮▮▮▮

[Lewis Carroll, Alice's Adventures in Wonderland, 1865]

Claude Elwood Shannon (1916-2001)

- Father of Information Theory
- Graduate of MIT 1940:
  “An Algebra for Theoretical Genetics’’
- 1941-1972: Scientist at Bell Labs
- 1958: Professor at MIT

- juggling, unicycling, chess
- ultimate machine
- Biographic movie: The Bit Player



The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

- Introduced a new research field: Information Theory

What is communication?

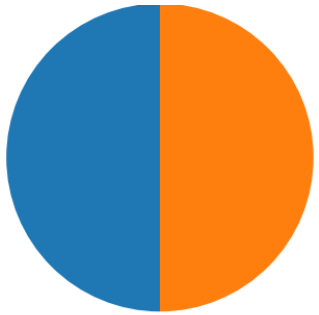
What is information?

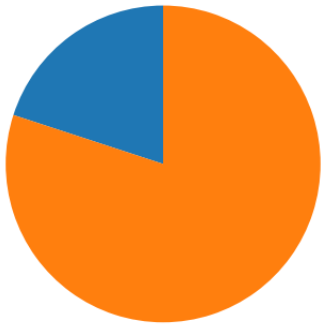
How much can we compress information?

How can we communicate reliably?

(Discrete) Probability Distributions

Definition: A list of non-negative numbers that sum to 1.



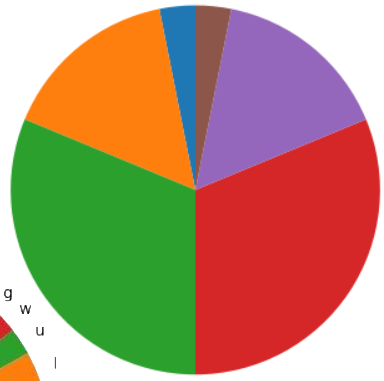


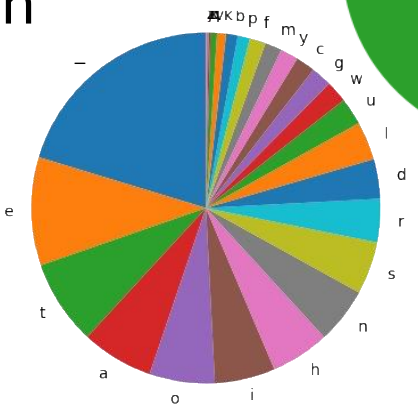
Easy examples:

- Fair coin: $[0.5, 0.5]$
- Biased coins: $[p, 1-p]$ for $0 \leq p \leq 1$ and $p \neq 0.5$
- Uniform distribution: $\left[\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right]$ for $n = 1,2,3, \ldots$

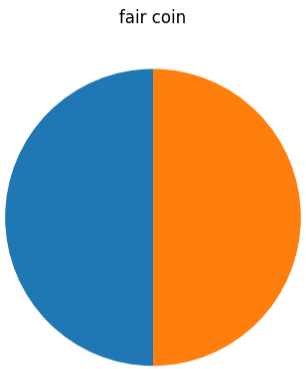
More complicated examples:

- Binomial distribution: $[B(n,p,0), B(n,p,1), \ldots, B(n,p,n)]$ where $B(n,p,k) \stackrel{\text{def}}{=} \binom{n}{k} p^k (1-p)^{n-k}$ is the probability to obtain exactly k heads when n times flipping a coin with bias p.
- Frequencies: count the number of occurrences and divide by the total





Sampling from Probability Distributions

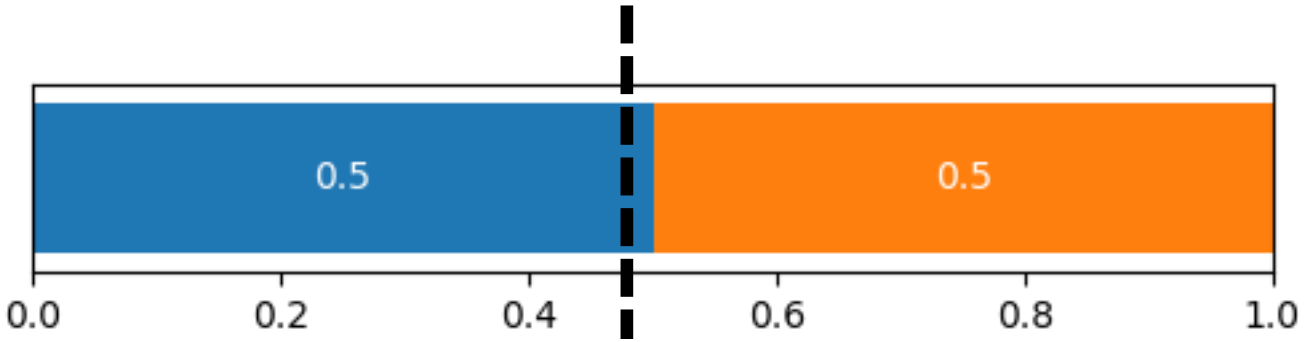



```
from random import random
random()
```

```
0.48546927067131873
```

```
if random() <= 0.5:
    print(0)
else:
    print(1)
```

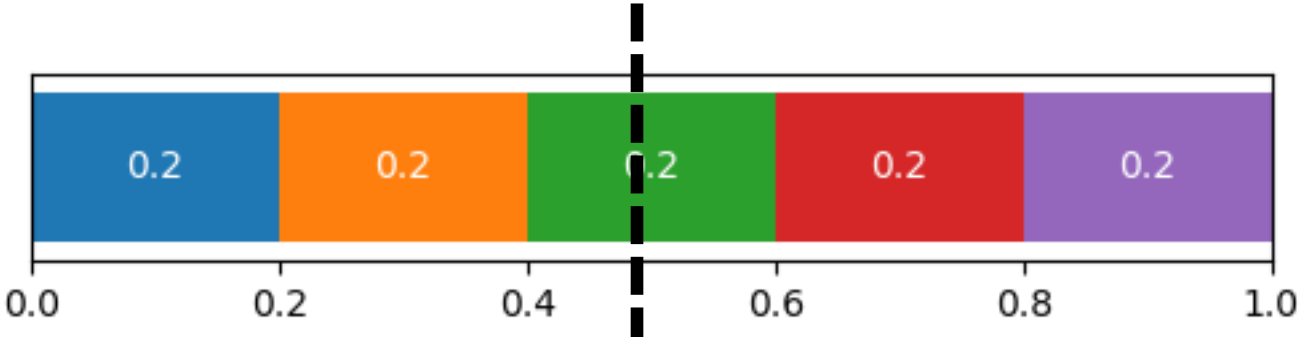
```
0
```



Sampling from Probability Distributions

```
from random import random
random()
```

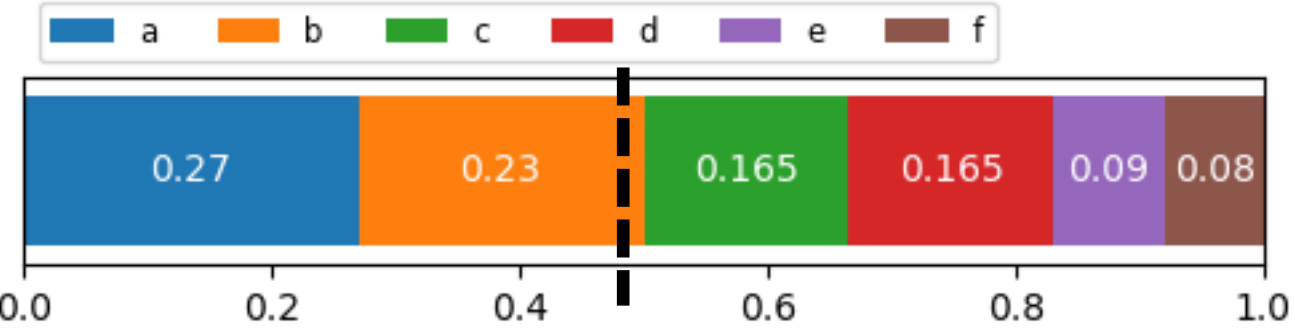
```
0.48546927067131873
```



Sampling from Probability Distributions

d a e f e e a b a d …

```
from random import random

def my_sample(labels, prob_dist):

    # a random float in the interval [0,1)
    random_value = random()

    p_mass = prob_dist[0]
    i = 0
    while p_mass < random_value:
        i += 1
        p_mass += prob_dist[i]

    return labels[i]
```




```
samples = ''
for i in range(60):
    samples += my_sample(['a', 'b', 'c', 'd', 'e', 'f'],
                         [0.27, 0.23, 0.165, 0.165, 0.09,0.08])
samples
```

```
'daefeeabadfeacacaebdacdbeaaabbadbaccfbfcdccdadacdacadceeaceb'
```

Surprisal 😳 ❓ 😮 😯 😱 🤯

- For an event with probability p, the **inverse probability** $\frac{1}{p}$ is a measure of how surprised we are if the event happens.
- Let us measure surprisal **logarithmically** (in bits): $\log\left(\frac{1}{p}\right) \geq 0$

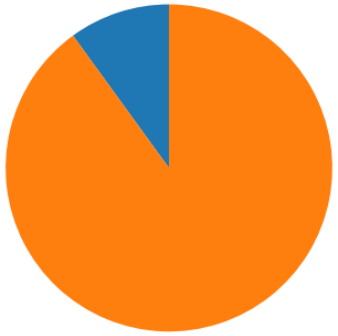
Easy examples:

- Fair coin: $[0.5, 0.5]$: You are $\log\left(\frac{1}{0.5}\right) = \log(2) = 1$ bit surprised to see any particular outcome

- Uniform distribution: $\left[\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right]$ for $n = 1,2,3, \ldots$:
  you are $\log\left(\frac{1}{1/5}\right) = \log(5) = 2.322$ bits surprised to see any particular outcome.

Average Surprisal 😳 ❓ 😮 😮 😱 🤯

- For an event with probability p, the inverse probability $\frac{1}{p}$ is a measure of how surprised we are if the event happens.
- Let us measure surprisal **logarithmically** (in bits): $\log\left(\frac{1}{p}\right) \geq 0$



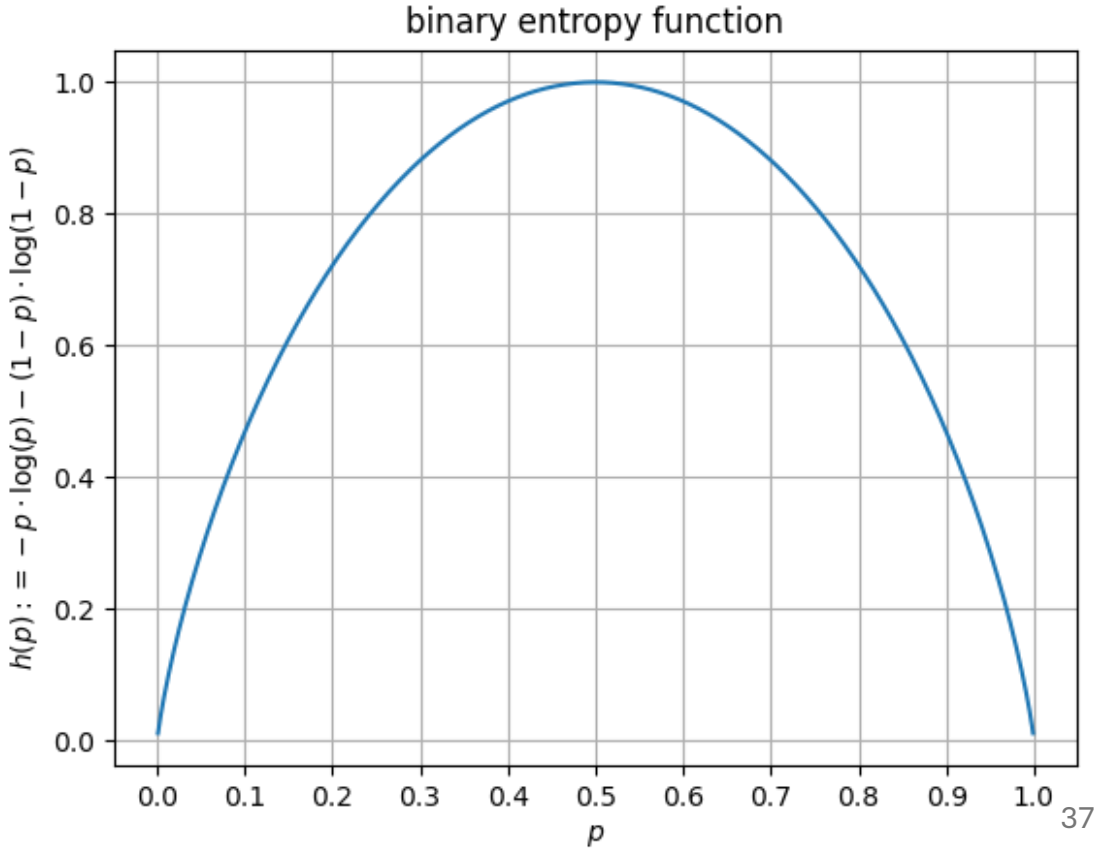
- Biased coin: $[0.1, 0.9]$ How surprised are we on average?

$$0.1 \times \log\left(\frac{1}{0.1}\right) + 0.9 \times \log\left(\frac{1}{0.9}\right)$$

$$= 0.1 \times \log(10) + 0.9 \times \log\left(\frac{10}{9}\right)$$

$$= 0.1 \times 3.32 + 0.9 \times 0.15$$

$$= 0.47$$



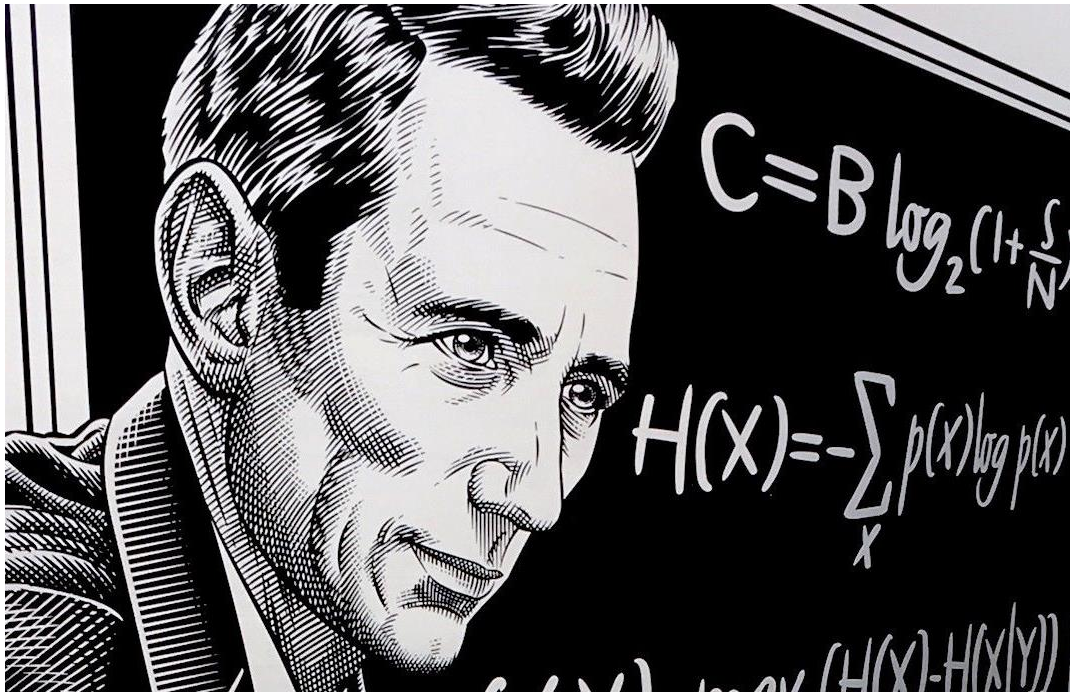
Average Surprisal = Shannon Entropy

- For an event with probability p, the inverse probability $\frac{1}{p}$ is a measure of how surprised we are if the event happens.
- Let us measure surprisal **logarithmically** (in bits): $\log\left(\frac{1}{p}\right) \geq 0$

Shannon entropy

$$\mathrm{H}([p_0, p_1, \ldots, p_n]) \stackrel{\text{def}}{=} \sum_i p_i \times \log\left(\frac{1}{p_i}\right)$$

is the average surprisal measured in bits



Properties of Shannon Entropy

Shannon entropy

$$\mathrm{H}([p_0, p_1, \ldots, p_n]) \stackrel{\text{def}}{=} \sum_i p_i \times \log\left(\frac{1}{p_i}\right)$$

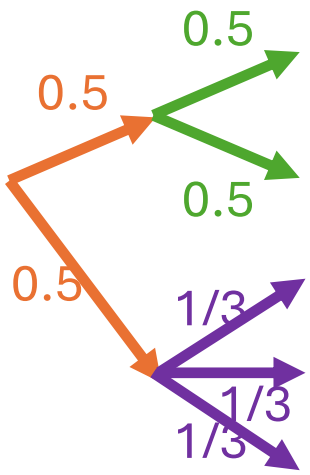
is the average surprisal measured in bits

```python
def entropy(prob_dist):
    '''
    :param prob_dist: a list of non-negati
    :return: (base 2) Shannon entropy of p
    '''
    assert(all(p>=0 for p in prob_dist))
    assert(np.isclose(sum(prob_dist),1))

    # https://en.wikipedia.org/wiki/Entrop
    # Shannon entropy H(p_i) = - sum_i p_i
    ent = 0.0
    for p in prob_dist:
        ent -= p * np.log2(p)

    return ent
```

- $0 \leq H(dist)$ with equality only for distributions with one certain outcome
- $H(dist) \leq \log n$ with equality for uniform distributions

- Computation trick:

$$H = 1 + 0.5 \times 1 + 0.5 \times \log(3)$$



symbol	probability
a	0.25
b	0.25
c	0.166
d	0.166
e	0.166

Order these in terms of Entropy

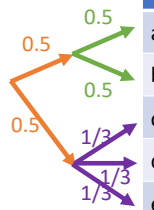
Shannon entropy

$$\mathrm{H}([p_0, p_1, \ldots, p_n]) \stackrel{\text{def}}{=} \sum_i p_i \times \log\left(\frac{1}{p_i}\right)$$

is the average surprisal measured in bits

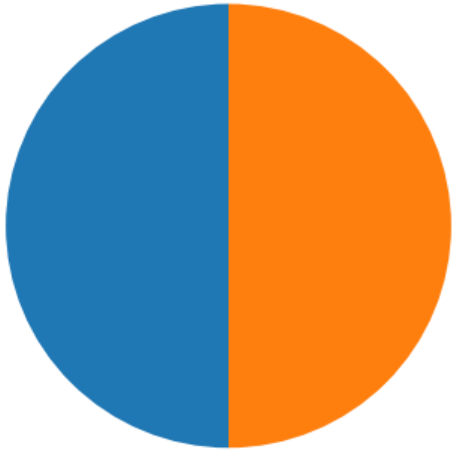
- $0 \leq H(dist)$ with equality only for distributions with one certain outcome
- $H(dist) \leq \log n$ with equality for uniform distributions

$$H = 1 + 0.5 \times 1 + 0.5 \times \log(3)$$



symbol	probability
a	0.25
b	0.25
c	0.166
d	0.166
e	0.166

$\mathrm{H} = \log(5) = 2.322$



$\mathrm{H} = \log(2) = 1$

$\mathrm{H} = \log(8) = 3$

Order these in terms of Entropy

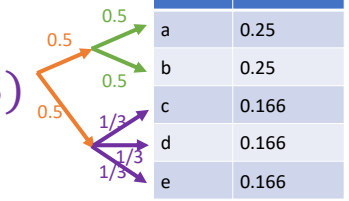
Shannon entropy

$$\mathrm{H}([p_0, p_1, \ldots, p_n]) \stackrel{\text{def}}{=} \sum_i p_i \times \log\left(\frac{1}{p_i}\right)$$

is the average surprisal measured in bits

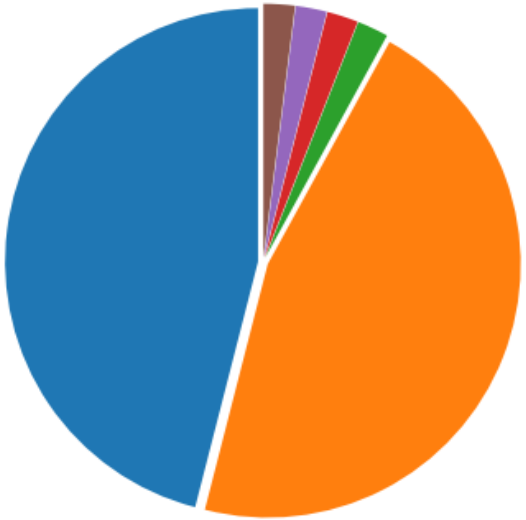
- $0 \leq H(dist)$ with equality only for distributions with one certain outcome
- $H(dist) \leq \log n$ with equality for uniform distributions

$$H = 1 + 0.5 \times 1 + 0.5 \times \log(3)$$



symbol	probability
a	0.25
b	0.25
c	0.166
d	0.166
e	0.166

$\mathrm{H} = \log(6) = 2.585$



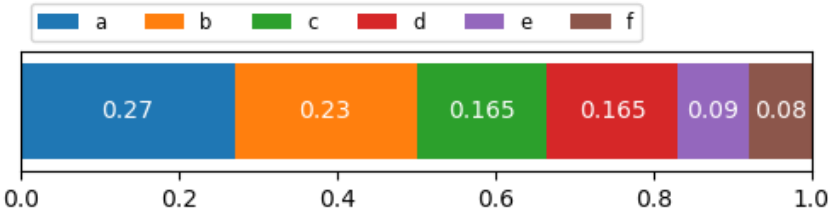
$\mathrm{H} = 1.482$

$\mathrm{H} = 2.457$

How to measure information content?

0 1 0 1 0 0 0 1 ...

d a e f e e a b a d ...



Measuring Information Content

How much “information” is contained in the data?

- compress it into minimal number of L bits (per source symbol)
- so that data can still be recovered perfectly

⇒ **average information content** is L bits per symbol

Information content L ≈ H(data distribution)

Part 1: lower bound on L

Information content L ≧ H(data distribution)

We want to prove that at least H(dist) bits per source symbol are required no matter what compression scheme is used.

$$
\begin{aligned}
H(X) - \ell_C(P_X) &= -\sum_{x\in\mathcal{X}} P_X(x)\log(P_X(x)) - \sum_{x\in X} P_X(x)\ell_x \\
&= \sum_{x\in\mathcal{X}} P_X(x)\left(-\log(P_X(x)) - \log\left(2^{\ell_x}\right)\right) \\
&= \sum_{x\in\mathcal{X}} P_X(x)\log\left(\frac{1}{P_X(x)\cdot 2^{\ell_x}}\right) \\
&\le \log\left(\sum_{x\in\mathcal{X}} \frac{1}{2^{\ell_x}}\right) && \text{(by Jensen's inequality)} \\
&\le \log(1) = 0 && \text{(by Kraft's inequality)} \qquad (2.9)
\end{aligned}
$$

■ qed

Part 2: upper bound on L

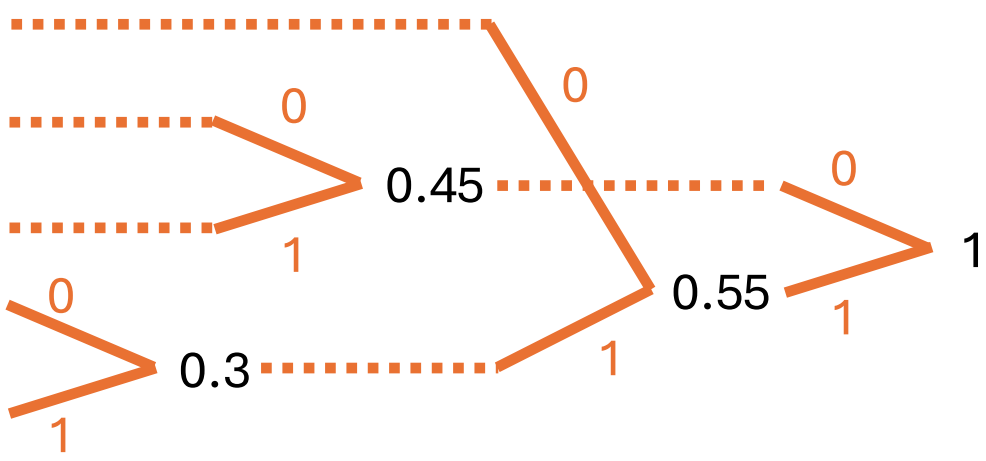
Information content L ≦ H(data distribution)

We give a compression method that uses H(dist) bits per symbol.

Huffman coding (by example):

symbol	probability
a	0.25
b	0.25
c	0.2
d	0.15
e	0.15

1. Combine the two nodes with *smallest probability* into new one with combined weight
2. Codeword is given by route from root to leaf.



Huffman code for letter frequency

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
a	0.0575	4.1	4	0000
b	0.0128	6.3	6	001000
c	0.0263	5.2	5	00101
d	0.0285	5.1	5	10000
e	0.0913	3.5	4	1100
f	0.0173	5.9	6	111000
g	0.0133	6.2	6	001001
h	0.0313	5.0	5	10001
i	0.0599	4.1	4	1001
j	0.0006	10.7	10	1101000000
k	0.0084	6.9	7	1010000
l	0.0335	4.9	5	11101
m	0.0235	5.4	6	110101
n	0.0596	4.1	4	0001
o	0.0689	3.9	4	1011
p	0.0192	5.7	6	111001
q	0.0008	10.3	9	110100001
r	0.0508	4.3	5	11011
s	0.0567	4.1	4	0011
t	0.0706	3.8	4	1111
u	0.0334	4.9	5	10101
v	0.0069	7.2	8	11010001
w	0.0119	6.4	7	1101001
x	0.0073	7.1	7	1010001
y	0.0164	5.9	6	101001
z	0.0007	10.4	10	1101000001
–	0.1928	2.4	2	01

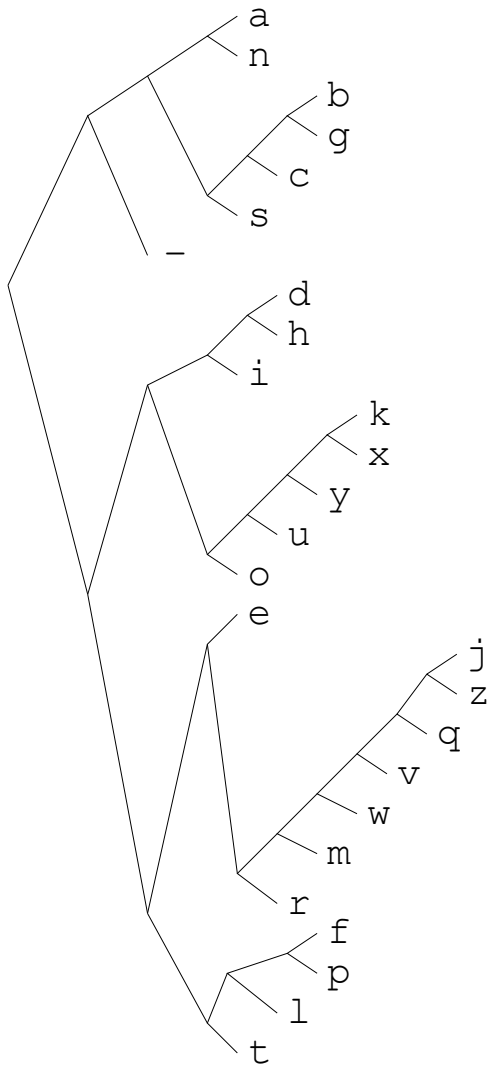
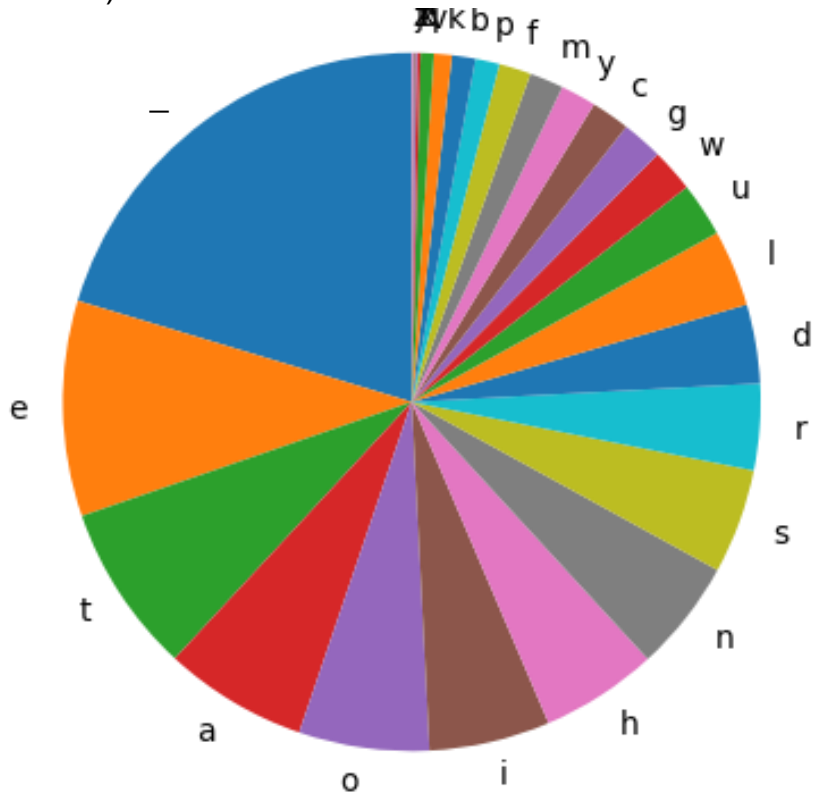


Figure 5.6. Huffman code for the English language ensemble (monogram statistics).

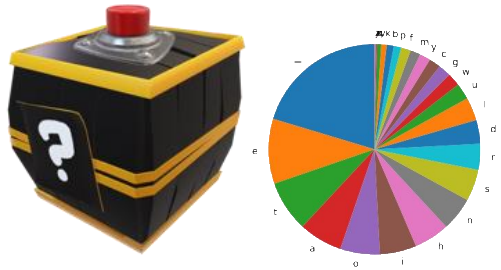


$$H = 4.042$$

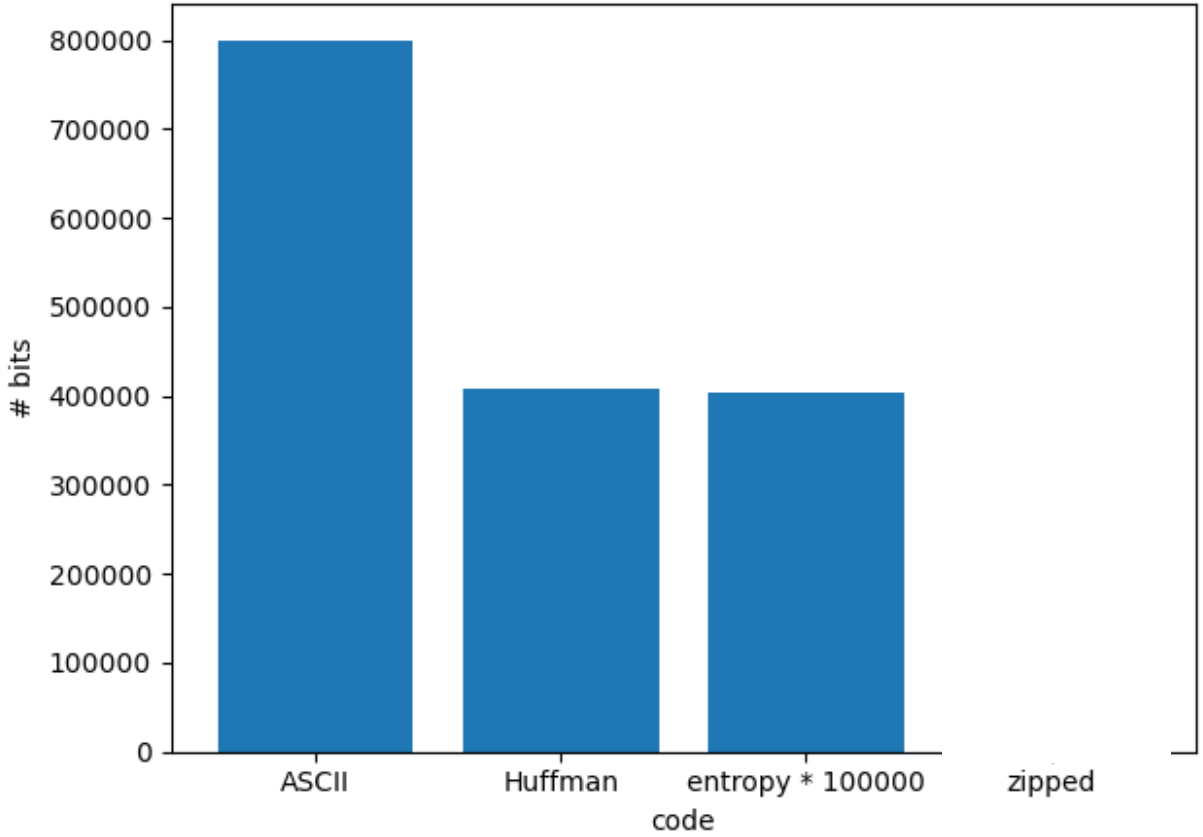
Book by David MacKay

Experiments: Alice vs letter samples

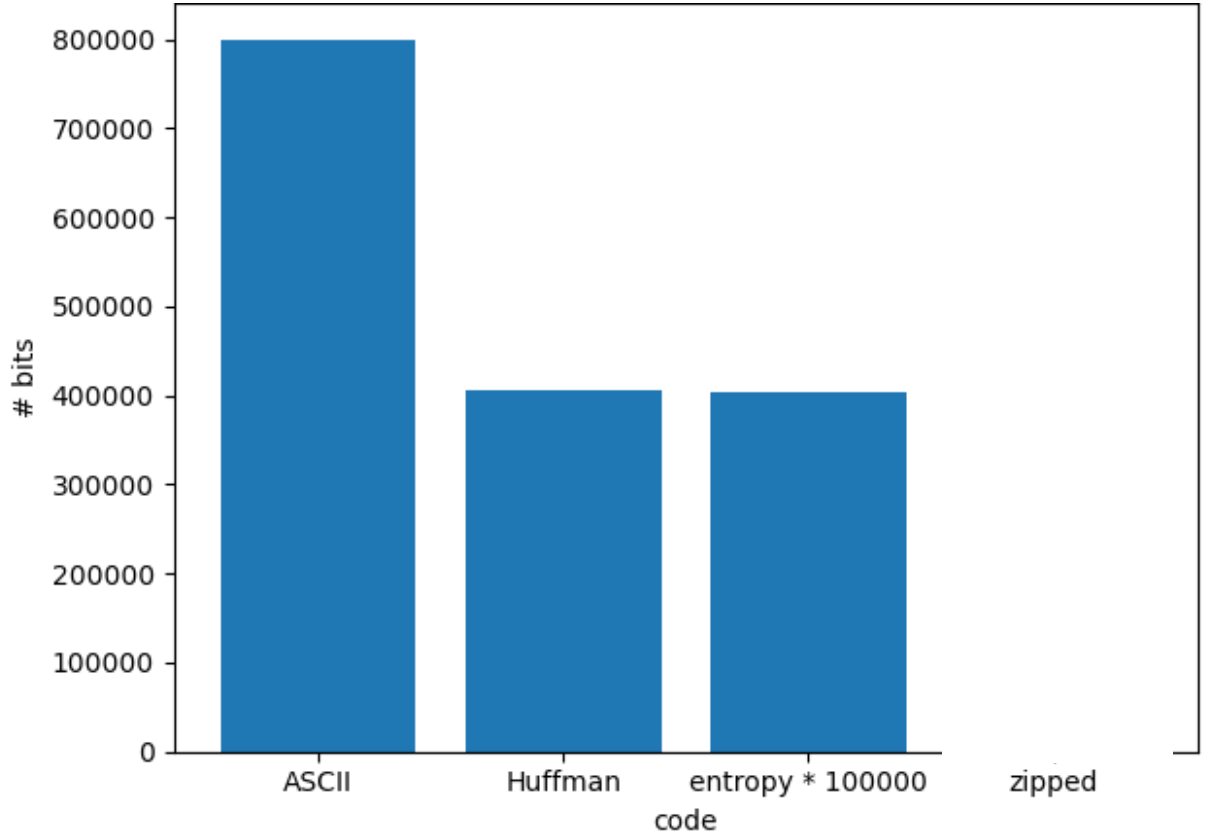
Information content L ≈ H(data distribution)



Experiment: Alice's adventures in Wonderland



Experiment: independent letter samples

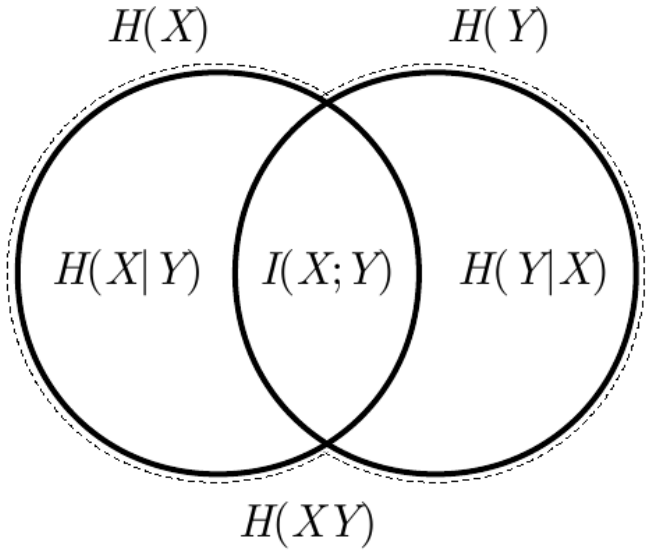


Two Random Variables with Joint Distribution

$H(X|Y) := \sum_y P_Y(y)\, H(X\,|Y = y) = -\sum_y P_Y(y) \sum_x P_{X|Y=y}(x)\, \log_2(P_{X|Y=y}(x)\,)$

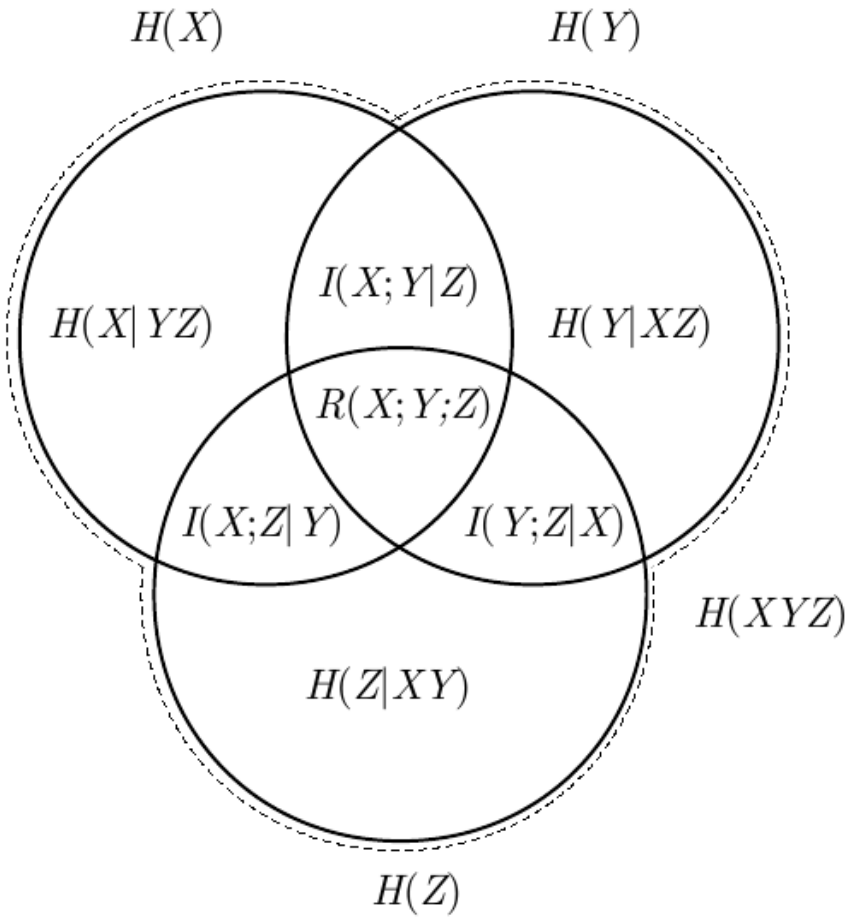
how uncertain am I about X once I know Y?

$I(X;Y) \coloneqq H(X) - H(X|Y)$: how much does Y tell me about X?



Interactive toy:
https://content.datanose.nl/interactive-graphs/entropy-diagrams.htm
Lecture Notes Information Theory:
https://github.com/cschaffner/InformationTheory/raw/master/Script/InfTheory3.pdf

Three variables



Every region is a sum or difference of joint entropies

The centre R(X;Y;Z) can be negative

Take X, Y independent bits and Z = X $\oplus$ Y

Conditional mutual information stays ≥ 0

[IT-Script, Figure 1.2 (right)]

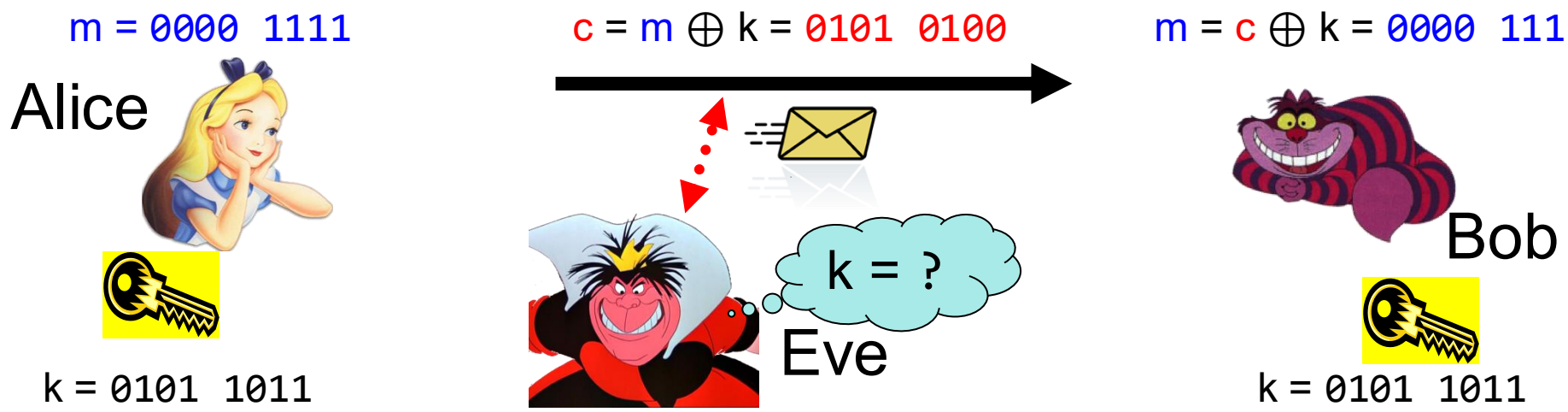
eXclusive OR (XOR) Function

x	y	x ⊕ y
0	0	0
1	0	1
0	1	1
1	1	0

- Some properties:
  - $\forall x : x \oplus 0 = x$
  - $\forall x : x \oplus x = 0$

$\forall x,y : x \oplus y \oplus y = x$

One-Time Pad Encryption



- Goal: Eve does not learn the message
- Setting: Alice and Bob share a key k
- Recipe:

m = 0000 1111

k = 0101 1011

c = m ⊕ k = 0101 0100

c = 0101 0100

k = 0101 1011

c ⊕ k = 0000 1111

c ⊕ k = m ⊕ k ⊕ k = m ⊕ 0 = m

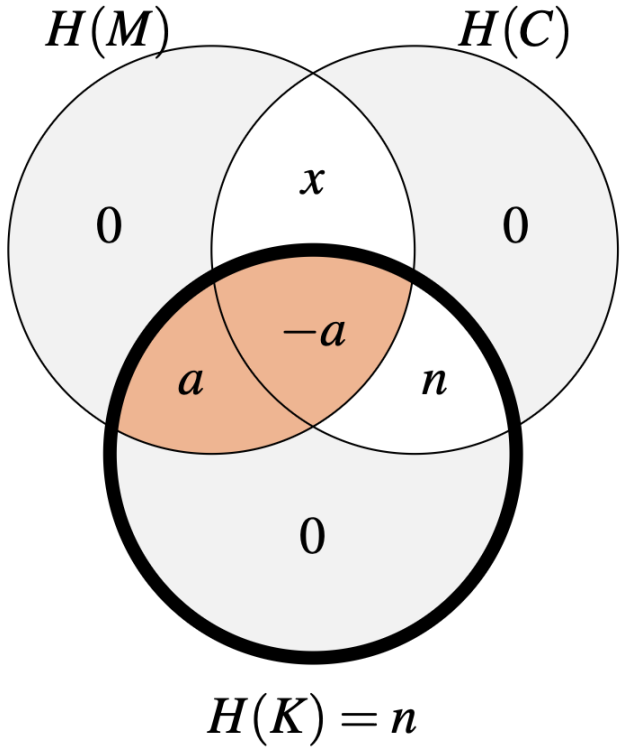
x	y	x ⊕ y
0	0	0
0	1	1
1	0	1
1	1	0

- Is it secure?

Security of One-Time Pad

C = M $\oplus$ K, fresh key every time

Claim 1. The one-time pad is perfectly secret: $I(M\,;\,C)\;=\;0$



Uniform key: $H(K) = n$

correctness: $H(M|CK) = H(C|KM) = H($

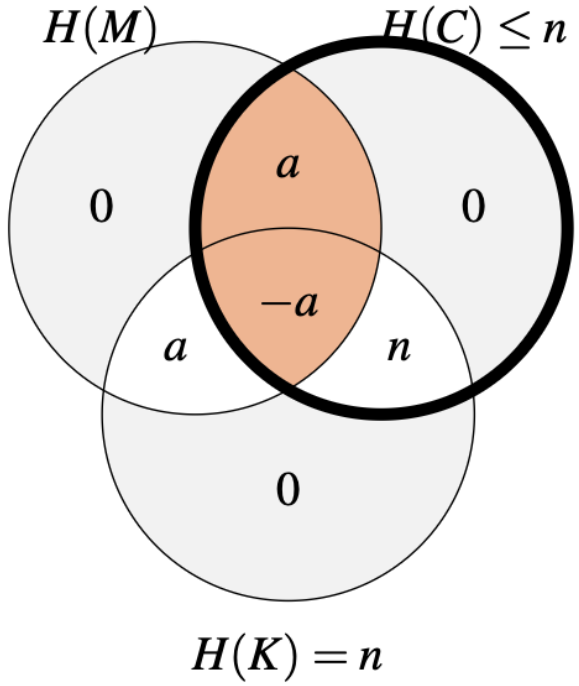
0

Setup assumption: $I(M;K) = 0$

$0 \le I(M\,;\,C) \Rightarrow x \ge a$

$H(C) = x - a + n \le n \Rightarrow x \le a$

Therefore, $x = a$ and $I(M;C) = 0$.



[Vernam 1917; Shannon 1949]

One-time pad

$C = M \oplus K$, fresh key every time

Claim 1. The one-time pad is perfectly secret: $I(M ; C) = 0$

Shannon's theorem: Perfect secrecy of encryption forces $H(K) \geq H(M)$

[Vernam 1917; Shannon 1949]

Definitional Details of OTP

Perfect secrecy was proved for messages of fixed length.

The ciphertext still reveals how long the message is.

Fine, if the length is genuinely not secret. The definition never told you that.

Alice reuses the OTP key

What can Eve compute?

Nothing — each ciphertext is still individually uniform

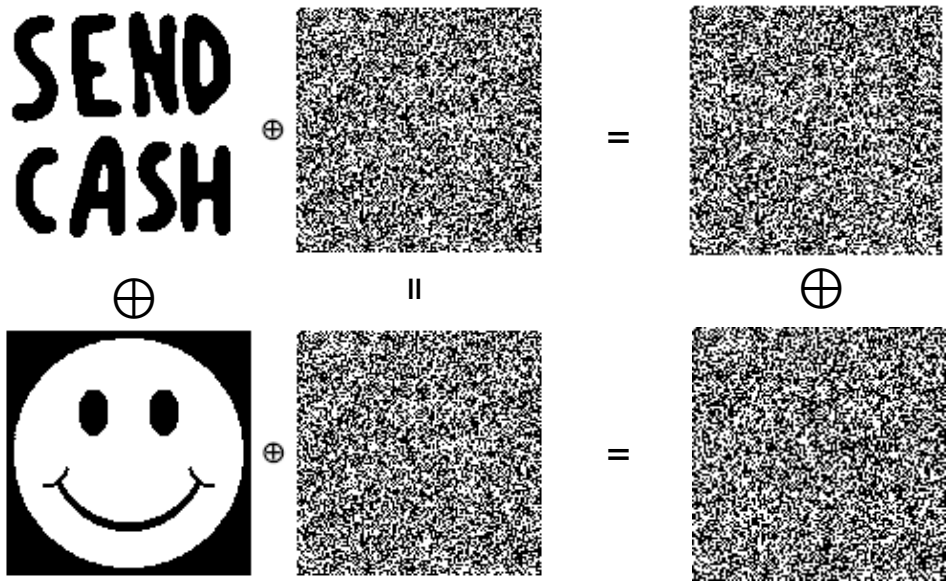
The key, directly

The XOR of the two messages

Only the lengths

https://app.wooclap.com/SZMYBFM

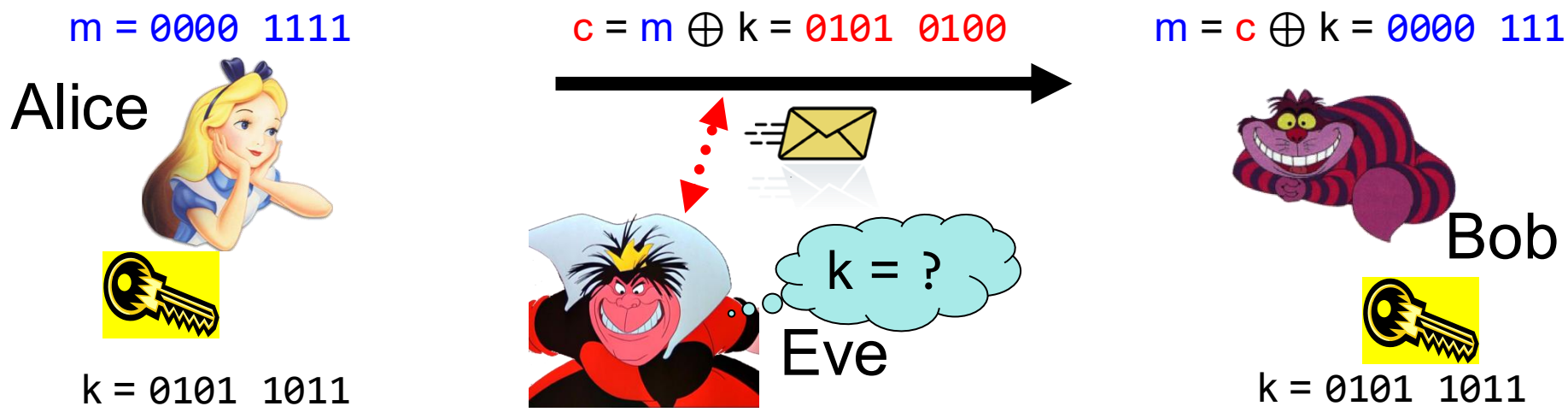
Key Re-Use in One-Time-Pad



$$m_1 \oplus k \oplus m_2 \oplus k = c_1 \oplus c_2 = m_1 \oplus m_2 =$$

[https://cryptosmith.com/2008/05/31/stream-reuse/]

Problems With One-Time Pad



- The key has to be as long as the message.
- The key can only be used once.
- In practice, other encryption schemes (such as AES) are used which allow to encrypt long messages with short keys.
- One-time pad does not provide authentication:
  Eve can easily flip bits in the message

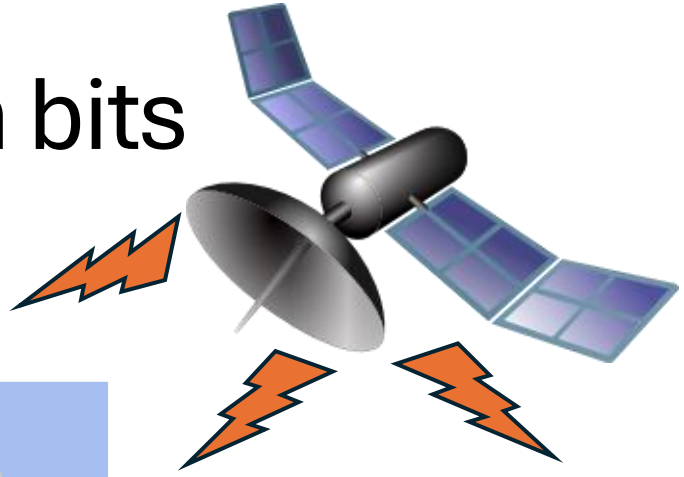
What will you learn this morning?

✓ Notions of security

✓ Shannon entropy and the one-time pad

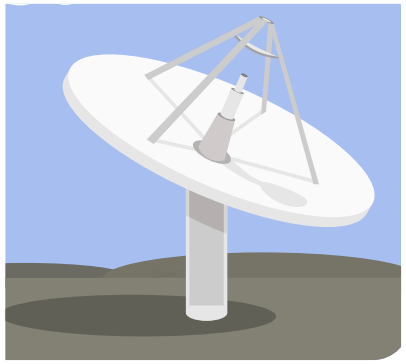
- Secret-key agreement by public discussion
- Authentication and hashing
- Error correction and privacy amplification
- Back to the real world

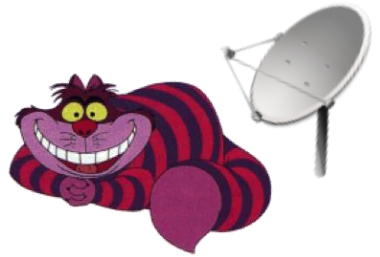
A satellite broadcasts random bits



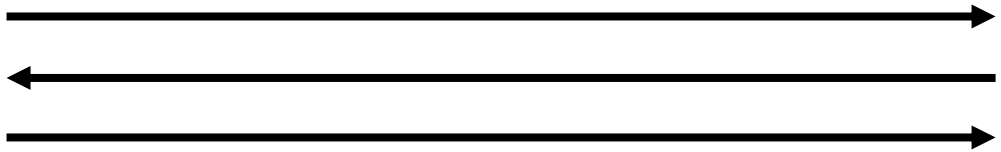
P_{ABE}

Eve

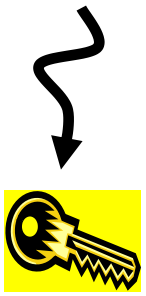


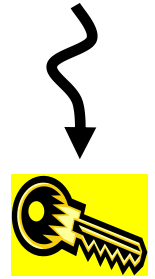


Alice, Bob and Eve each get a noisy version of the same random string.



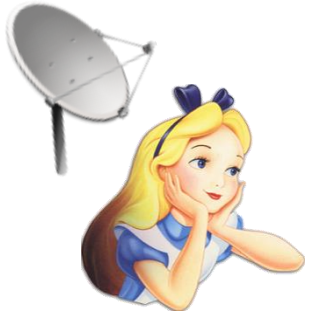
(authenticated) public discussion





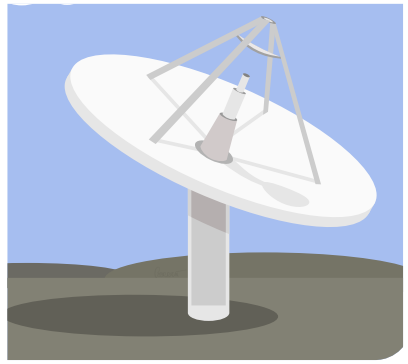
[Maurer 1993]

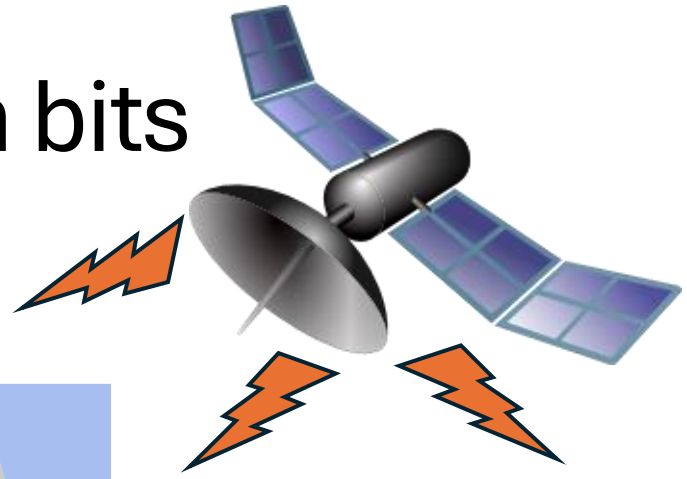
A satellite broadcasts random bits

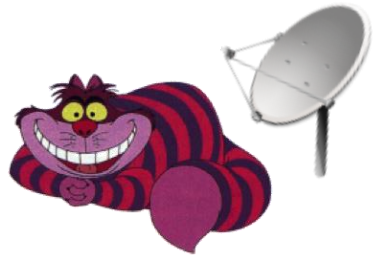


P_{ABE}

Eve







Alice, Bob and Eve each get a noisy version of the same random string.

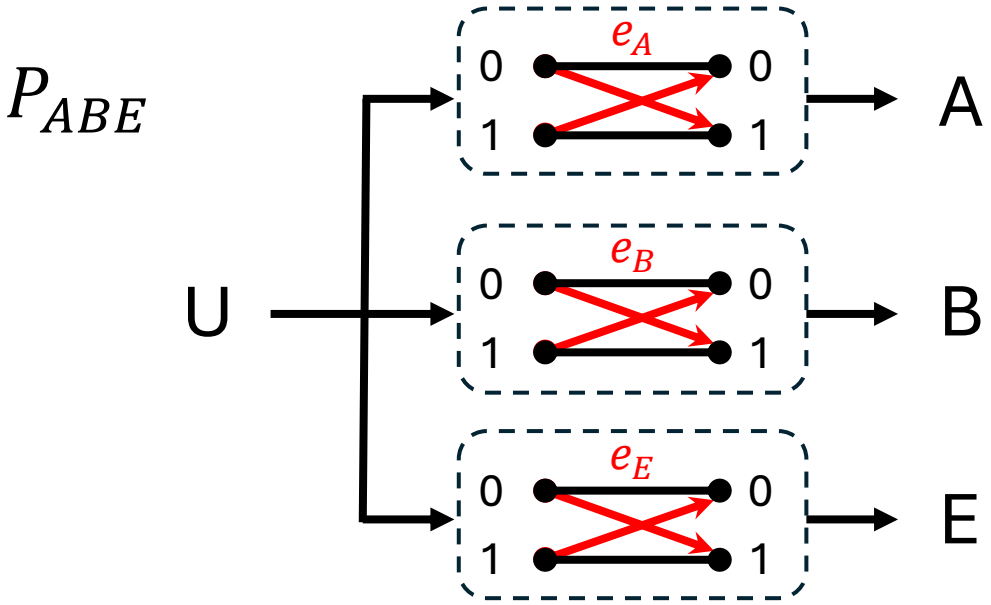
Theorem: The secret key rate $S(A;B||E)$ is bounded as follows:

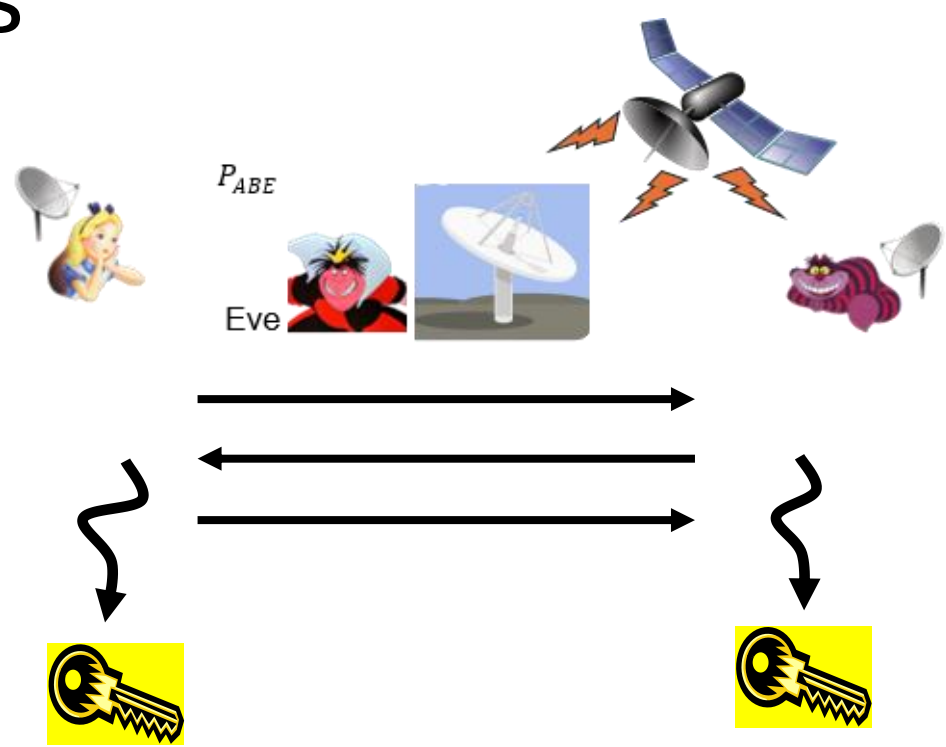
$$S(A;B||E) \leq \min\{I(A;B), I(A;B|E)\}$$

$$\max\{0, I(A;B) - I(A;E), I(A;B) - I(B;E)\} \leq S(A;B||E)$$

[Maurer 1993]

A satellite broadcasts random bits

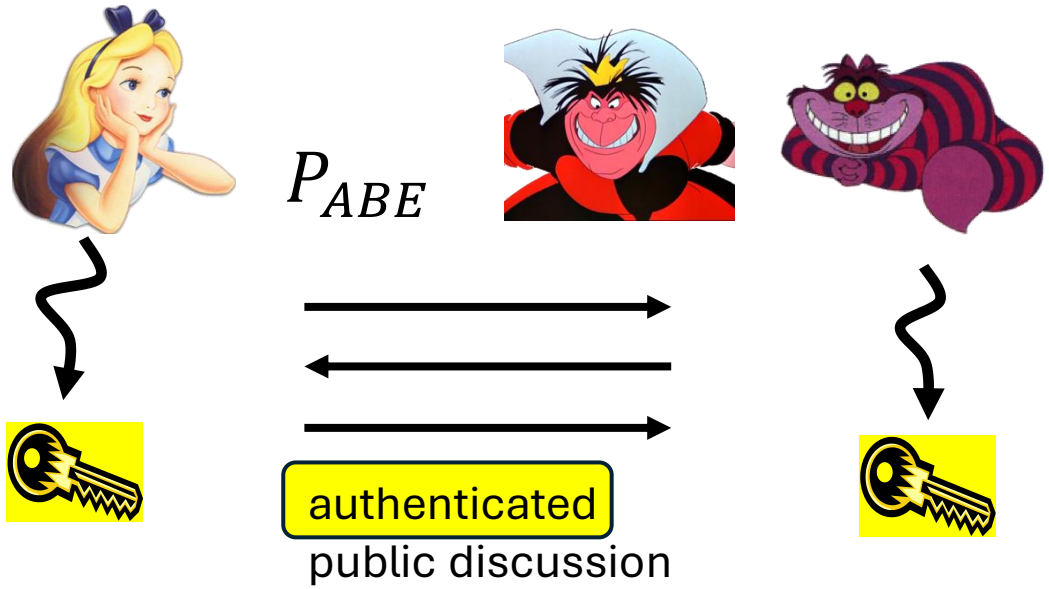


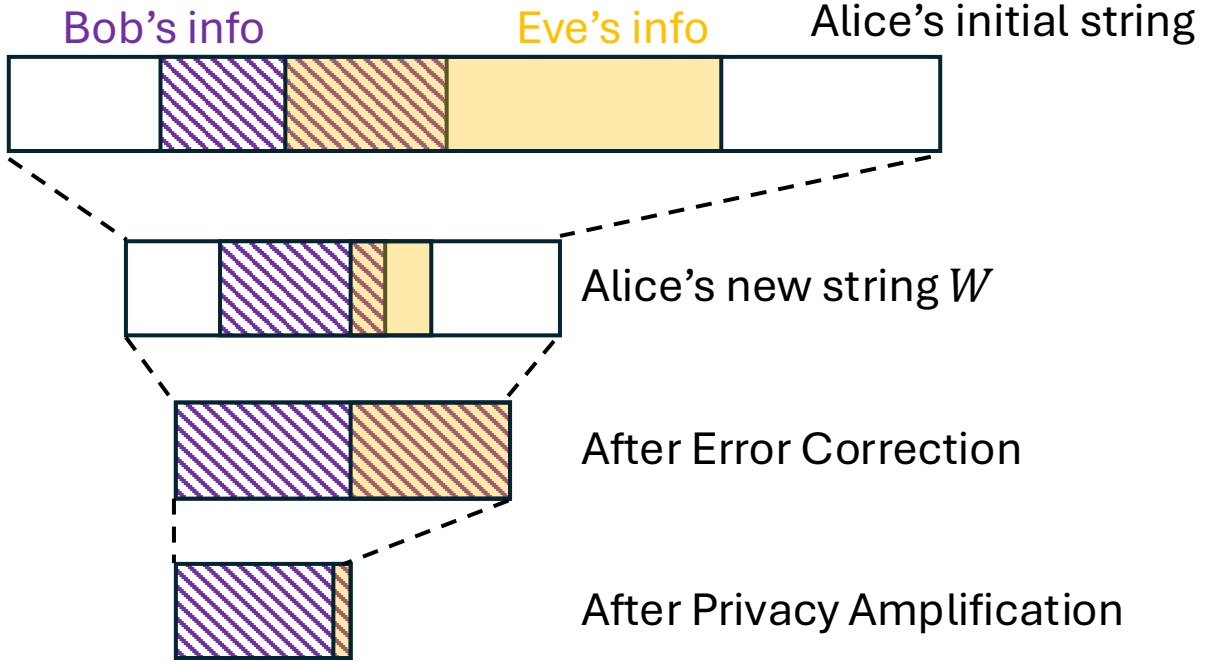


Theorem: The secret key rate $S(A;B||E) > 0$ unless $e_E = 0$ or $e_A = \frac{1}{2}$ or $e_B = \frac{1}{2}$.

[Maurer 1993]

Secret-Key Agreement by Public Discussion





1. Advantage Distillation: create a string W that Alice knows, about which Bob has more information than Eve.

2. Information Reconciliation / Error Correction: Alice and Bob exchange error-correction information to agree on the same string

3. Privacy Amplification: Alice and Bob agree on a hash function to apply and leave Eve with a negligible amount of information

Notation for the rest of the week

X, Y — Alice's and Bob's data

Z — the key map output

E — Eve's system

C — all classical announcements

ℓ — final key length

ε — failure probability

$leak_{EC}$ —bits spent on error correction

What will you learn this morning?

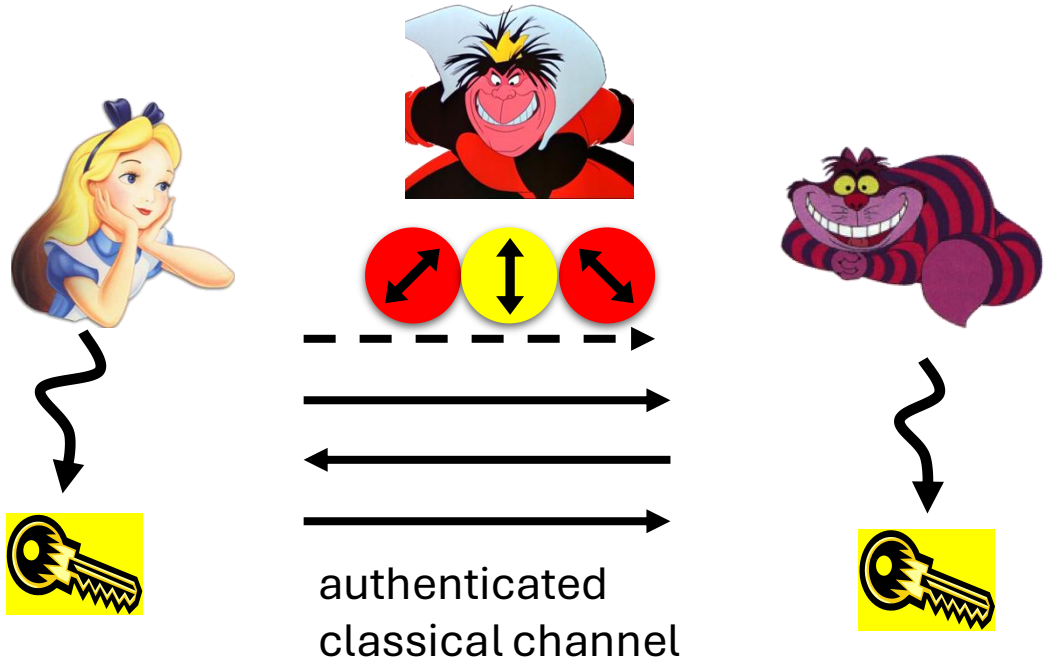
✓ Notions of security

✓ Shannon entropy and the one-time pad

✓ Secret-key agreement by public discussion

- Authentication
- Error correction and privacy amplification
- Back to the real world

Authentication Attacks



Impersonation: Eve injects a message out of nowhere and hopes it is accepted.

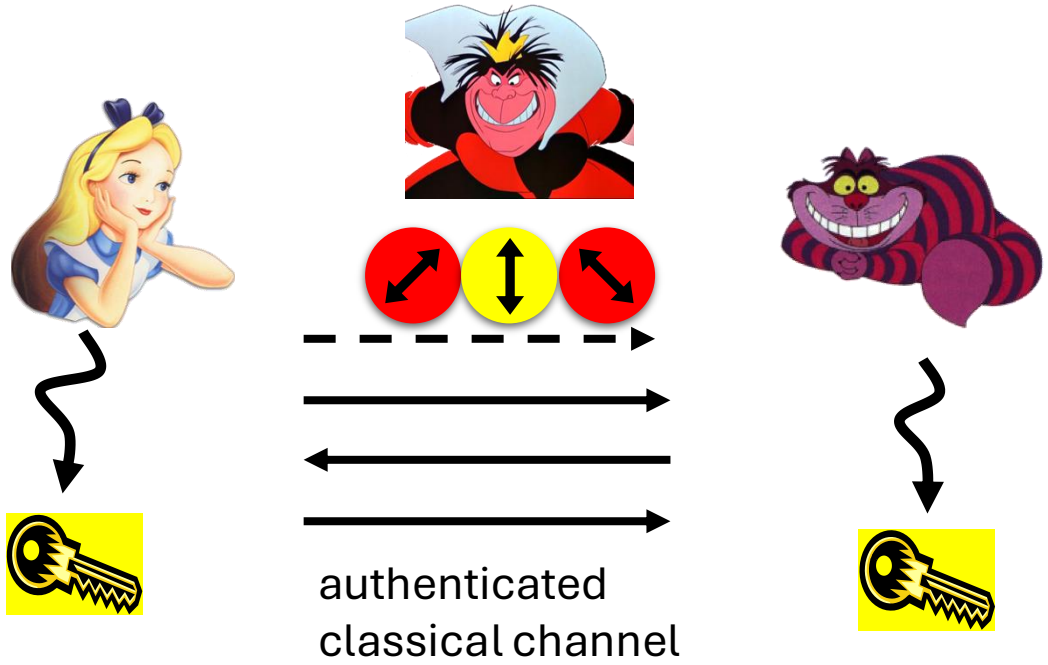
Substitution: Eve sees a valid message, then swaps it.

Replay: Sends the same message again

Reorder: Deliver messages out of order

Denial of Service: does not deliver messages at all

Authentication Attacks



Impersonation: Eve injects a message out of nowhere and hopes it is accepted.

Substitution: Eve sees a valid message, then swaps it.

Replay: Sends the same message again

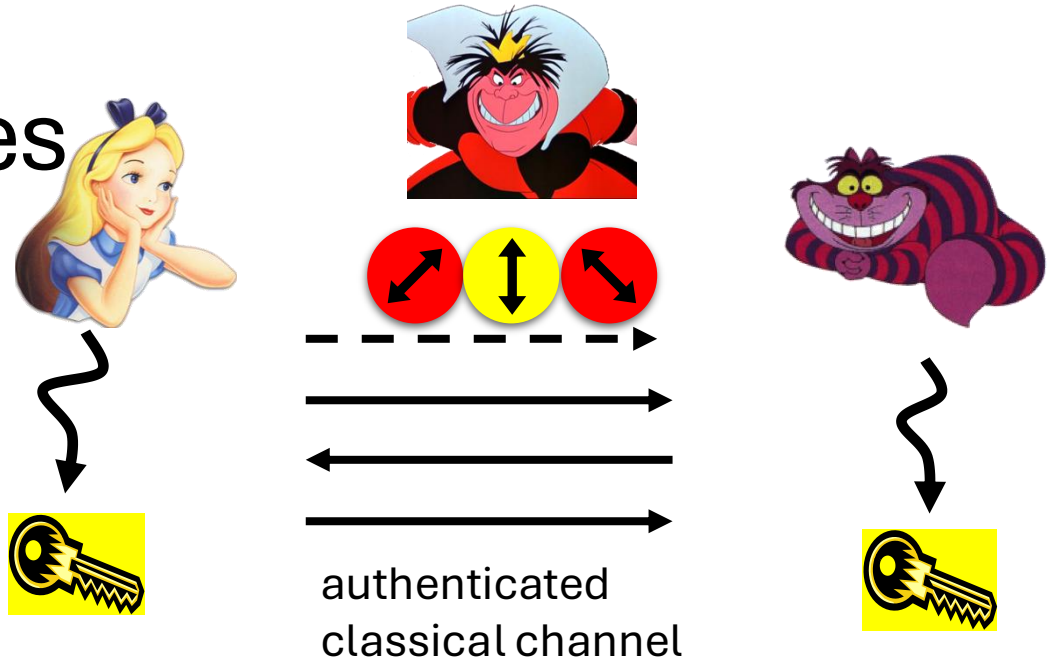
Reorder: Deliver messages out of order

Denial of Service: does not deliver messages at all

Message Authentication Codes

$h_k(m) = t$ takes message and key, produces a tag

Send message-tag pair: (m, t)



Substitution: Eve sees message with valid tag, then swaps it.

Impersonation: inject a message out of nowhere and hope it is accepted.

Theorem: For MACs of the form (m, t), security against substitution attacks implies security against impersonation attacks.

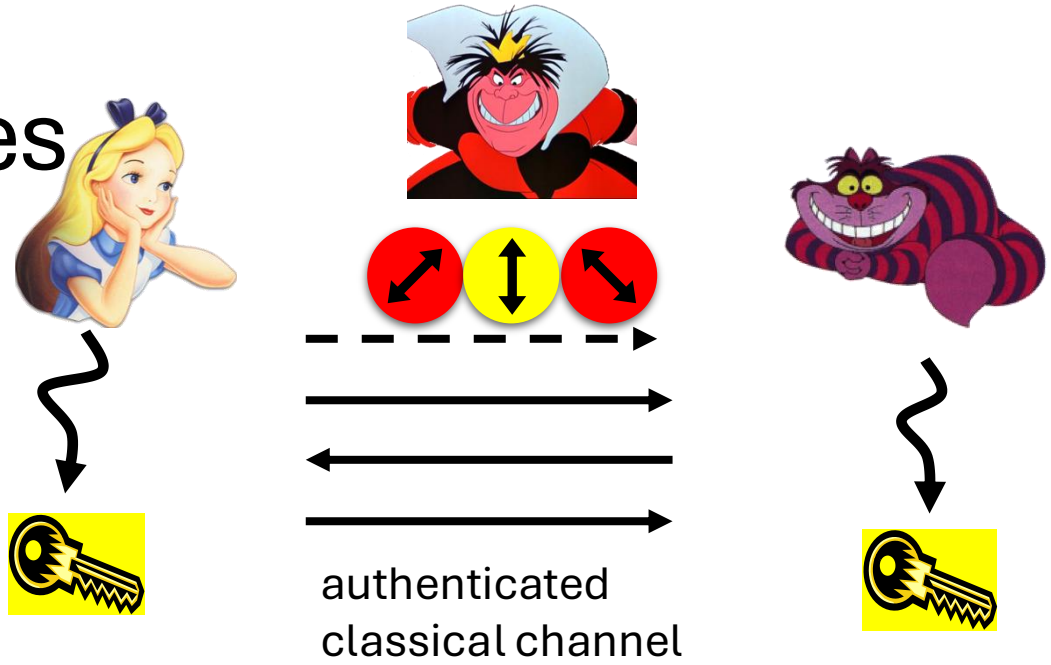
Proof: Impersonation attackers inject messages out of nowhere. Any such attacker can be transformed into a substitution attacker, by first asking for a tag on a different message, throwing away the answer, and proceed as before.

[see Portmann https://arxiv.org/abs/1202.1229]

Message Authentication Codes

$h_k(m) = t$ takes message and key, produces a tag

Send message-tag pair: (m, t)



Substitution: Eve sees message with valid tag, then swaps it.

Impersonation: inject a message out of nowhere and hope it is accepted.

$$P_{impers} = \max_{m',t'} \Pr_K[h_K(m') = t']$$

$$P_{sub} = \max_{m,t} \max_{m' \neq m,t'} \Pr_K[h_K(m') = t' | h_K(m) = t]$$

Worst case about unbounded Eve's choices, no message distribution involved.

(Strongly) 2-Universal Hash Families

Def: A family $H = \{h\colon\ M \to T\}$ of hash functions is called 2-universal if

$$\forall m \neq m' \colon \Pr_h[h(m) = h(m')] \leq 1/|T|$$

H is called strongly 2-universal if

$$\forall m \neq m', t, t' \colon \Pr_h[h(m) = t \ \ and\ h(m') = t'] = 1/|T|^2$$

Lemma: Strongly 2-universal implies 2-universal

Lemma: Strongly 2-universal implies that every output value is uniform

[Wegman, Carter 1981]

Construction of 2-Universal Family

Let $GF(2^n)$ be the finite field over 2^n elements, i.e. we can not only add and substract, but also multiply and divide elements of $GF(2^n)$.

Consider the family $H = \{h_a(m) = a \cdot m \mid a \in GF(2^n)\}$

Lemma: H is two-universal, but not strongly 2-universal

Using H has MAC, Eve can observe a message/tag pair (m, t) with $t = h_a(m) = a \cdot m$, and then compute the key $a = t \cdot m^{-1}$.

[Wegman, Carter 1981]

Construction of Strongly 2-Universal Family

Instead, we consider the family $H = \{h_{a,b}(m) = a \cdot m + b \mid a, b \in GF(2^n)\}$

Lemma: H is strongly two-universal

Theorem: For H a strongly two-universal family, $P_{sub} = 1/|T|$

Proof: $P_{subst} = \max_{m,t} \max_{m' \neq m,t'} \Pr_K[h_K(m') = t' | h_K(m) = t]$

$$= \max_{m,t} \max_{m' \neq m,t'} \frac{\Pr_K[h_k(m') = t' and\ h_K(m) = t]}{\Pr_k[h_K(m) = t]} = \frac{1/|T|^2}{1/|T|} = 1/|T|$$

[Wegman, Carter 1981]

Authenticating Longer Messages

Write $m = (m_1, m_2, \ldots, m_L)$ with each $m_i \in GF(2^n)$ and set

$poly_m(x) := \sum_{i=1}^{L} m_i x^i$.

This family is called ε-almost XOR 2-universal or ε-difference universal, if

$$\forall m \neq m, \Delta \in T, \Pr_K[poly_K(m) - poly_K(m') = \Delta] \leq \varepsilon$$

Then, consider the hash family defined by

$$h_{a,b}(m) = poly_m(a) + b$$

authenticates messages of length $L \cdot n$ with $2n$ bits of key.

Theorem: $P_{impers} = 2^{-n}$ and $P_{subst} \leq L \cdot 2^{-n}$.

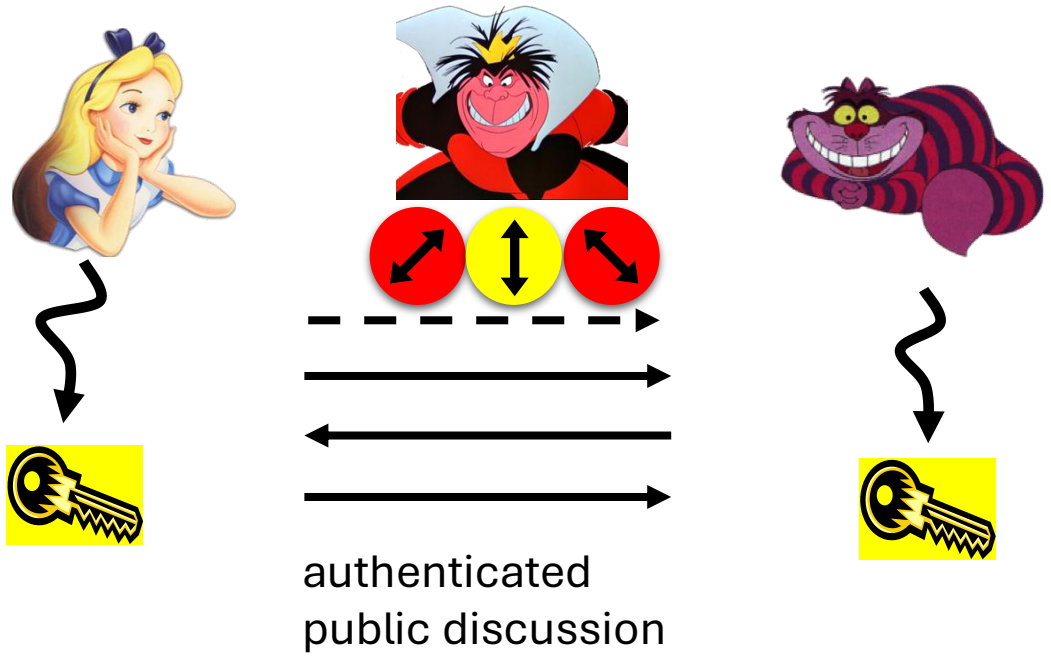
For QKD: Key consumption can be reduced from $2n$ to n bits per authenticated messge by keeping a for several messages, and only refreshing the b.

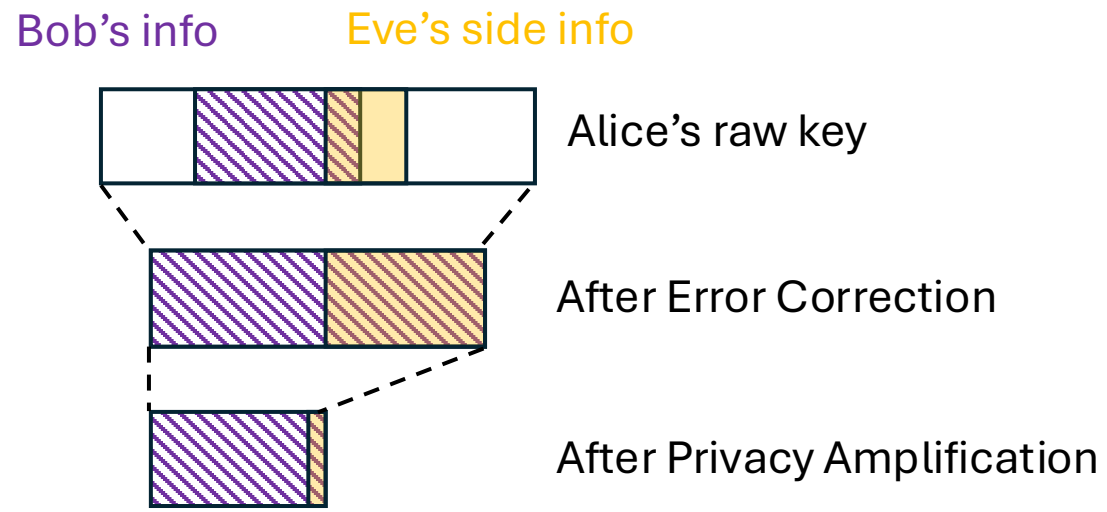
[see Portmann https://arxiv.org/abs/1202.1229]

What will you learn this morning?

- ✓ Notions of security
- ✓ Shannon entropy and the one-time pad
- ✓ Secret-key agreement by public discussion
- ✓ Authentication
- Error correction and privacy amplification
- Back to the real world

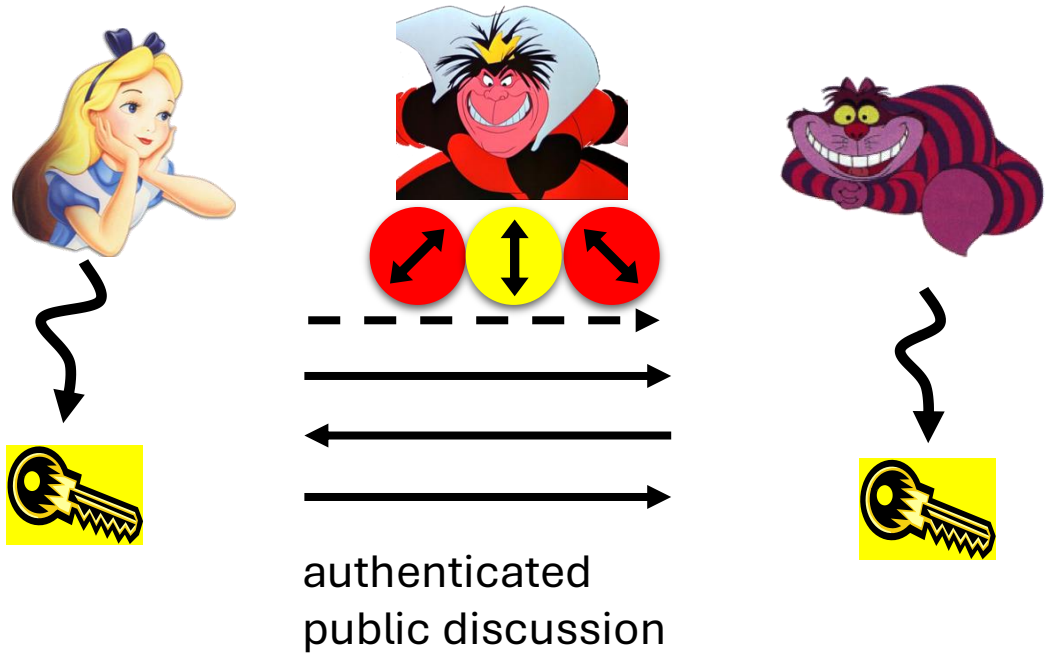
Post-Processing of QKD

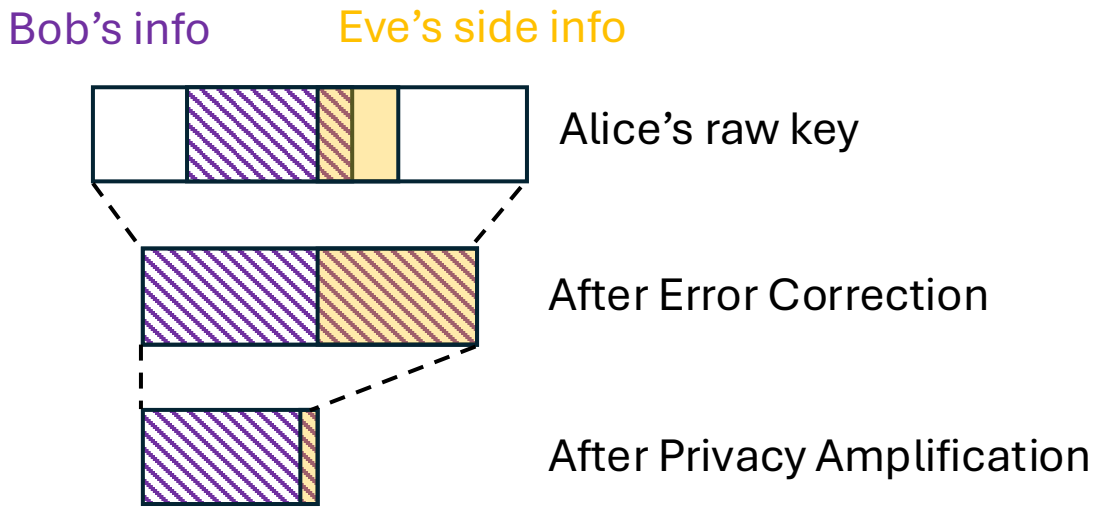




- Information Reconciliation / Error Correction: Alice and Bob exchange error-correction information to agree on the same string
- Privacy Amplification: Alice and Bob agree on a hash function to apply and leave Eve with a negligible amount of information

Error Correction for QKD





- Information Reconciliation / Error Correction: Alice and Bob exchange error-correction information to agree on the same string
- Caused by noise, or Eve, or both: by convention we blame Eve for all of it
- Alice and Bob must agree on a string before hashing.
- Every bit they send in public is a bit Eve gets.

Can you read this English text?

THE C▮IEF DIFFI▮ULTY AL▮CE FOU▮D ▮T FIRST WAS IN
▮A▮AGING HER FLA▮INGO: SHE SUCCE▮DE▮ ▮N
G▮▮▮TI▮G I▮S BODY ▮U▮KE▮ A▮A▮, COMFOR▮ABLY
E▮O▮G▮, UNDER ▮ER ▮▮▮M, WIT▮ ▮TS ▮▮▮▮S HANGI▮G
DO▮▮▮, B▮T ▮ENE▮A▮▮▮Y, ▮UST AS S▮▮▮ H▮D ▮OT ▮TS
▮▮▮CK N▮CE▮▮▮ S▮R▮IGHTE▮E▮ ▮▮▮T, AN▮ W▮S ▮▮▮▮NG
T▮ ▮I▮E T▮▮▮ HEDGEHOG ▮ ▮L▮W W▮TH ITS ▮▮▮AD, ▮▮▮
W▮UL▮ TW▮ST ▮▮▮SE▮F ▮OU▮▮▮ AN▮ ▮▮▮▮K ▮P IN H▮R
▮AC▮, ▮▮▮T▮ ▮U▮▮▮ A ▮▮▮▮Z▮ED E▮▮▮RE▮S▮O▮ ▮▮▮▮▮▮
SH▮ C▮U▮D ▮▮▮▮▮ ▮▮▮▮▮▮▮ ▮▮▮▮S▮▮▮▮G ▮▮▮▮▮ ▮A▮▮▮H▮▮▮▮:
A▮D WH▮N ▮▮▮▮▮ ▮A▮ ▮O▮ ▮▮▮▮▮ ▮▮▮▮▮▮▮ DO▮▮▮, ▮▮▮D
▮▮▮S G▮I▮G ▮▮▮ B▮▮▮▮▮N ▮▮▮▮▮I▮, ▮▮▮ ▮▮▮S ▮▮▮▮▮Y
▮▮▮O▮O▮▮▮▮▮▮ ▮▮▮ ▮▮▮▮▮▮▮ ▮▮▮▮▮T T▮▮▮ ▮E▮G▮▮▮▮▮▮ ▮▮▮D
▮▮▮R▮L▮▮▮▮▮ ▮▮▮▮▮▮▮▮F, ▮▮▮▮▮ ▮▮▮▮▮ ▮▮▮ ▮H▮ ▮▮▮▮▮ ▮▮▮
▮▮▮▮▮▮▮▮▮▮▮▮▮ ▮▮▮▮▮▮▮

[Lewis Carroll, Alice's Adventures in Wonderland, 1865]

The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

- Introduced a new research field: Information Theory

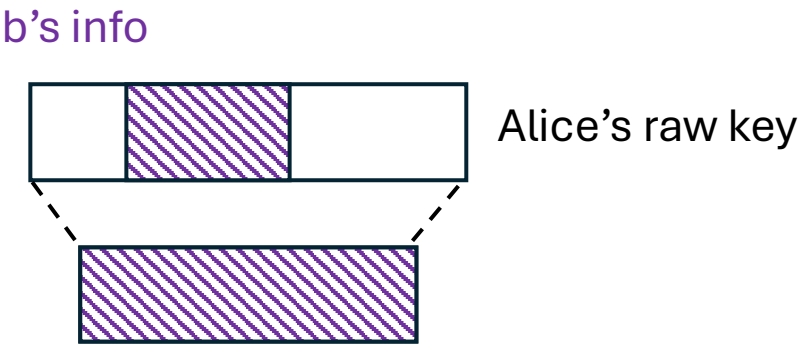
What is communication?

What is information?

How much can we compress information?

How can we communicate reliably?

Error Correction for QKD



Alice has string X, and Bob has partial information Y about it.

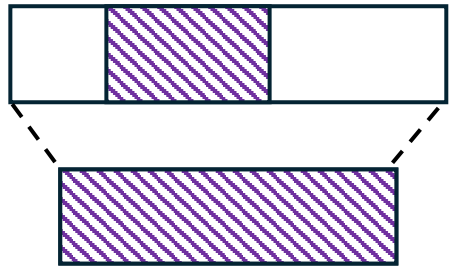
Assume $X \in \{0,1\}^n$ and $Y = BSC_\delta(X)$. Then $H(X|Y) = h(\delta)n$.

First idea: Alice sends a parity-check bit: $Z = \bigoplus_i X_i$.
Then Bob can verify whether his Y has the same parity.

More advanced: Send $Z = H\ X$, where H is the parity-check matrix of a linear error-correcting code.

Example: [7,4,3] code

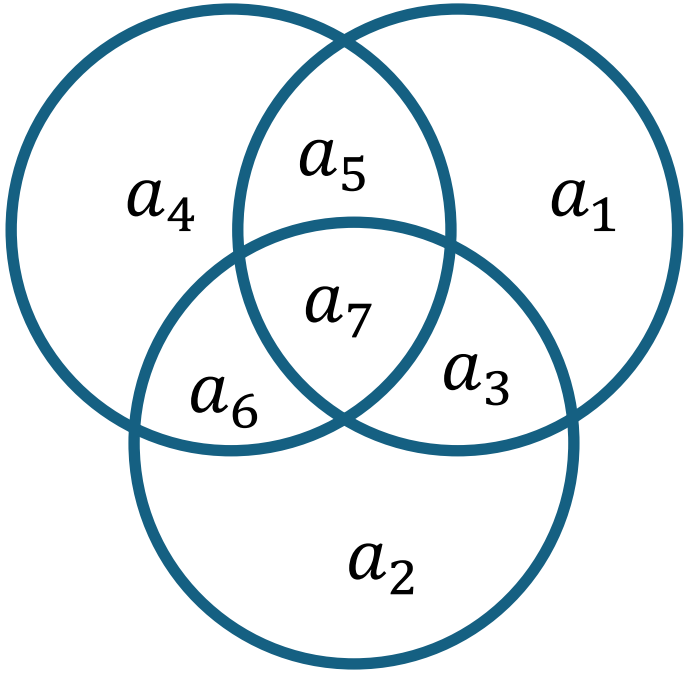
Bob's info



Alice's raw key

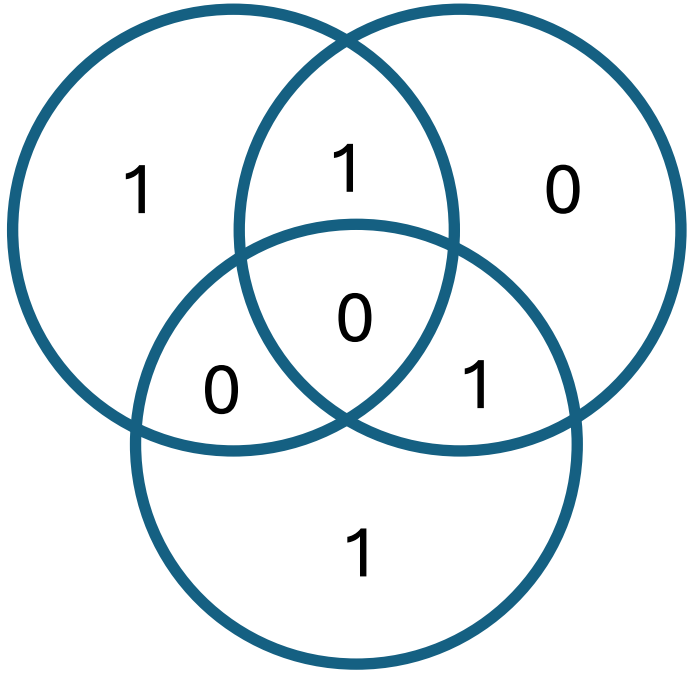
More advanced: Send $Z = $ parity-check matrix of a linear error-correcting code.

https://app.wooclap.com/SZMYBFM



Parity-check matrix:

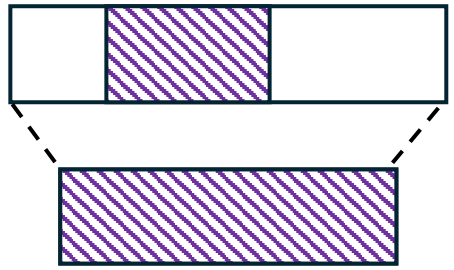
$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$



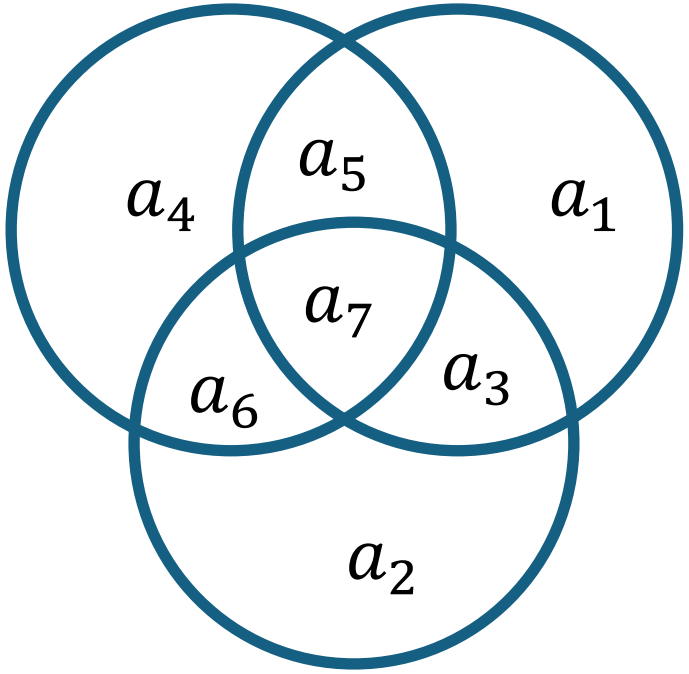
Bob has: Y = 0110100, receives Z=001
Which string does Alice have?

Invariant: even parity per circle!

Example: $[7,4,3]_2$ Hamming Code

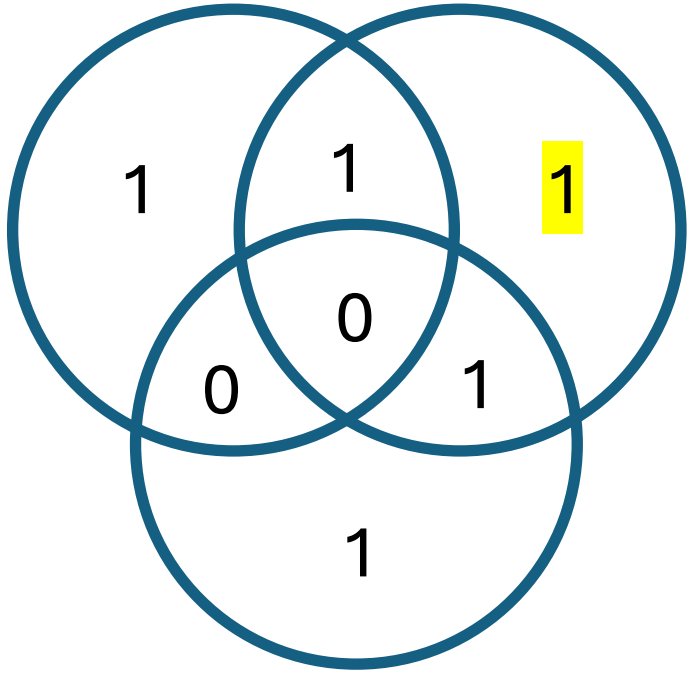


More advanced: Send $Z = H\ X$, where H is the parity-check matrix of a linear error-correcting code.



Parity-check matrix:

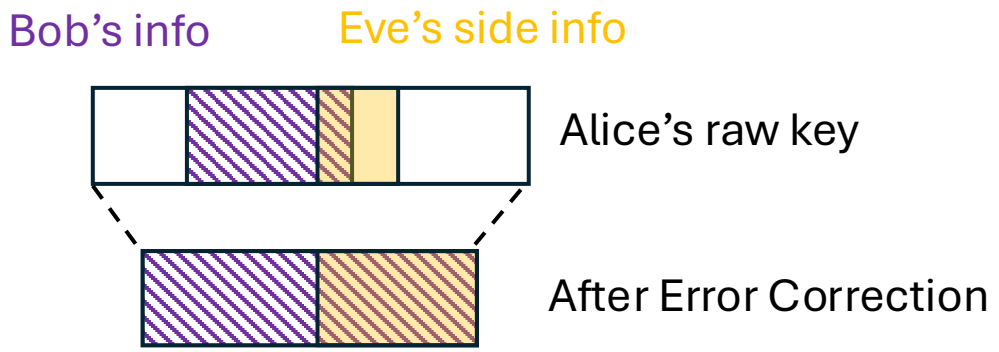
$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$



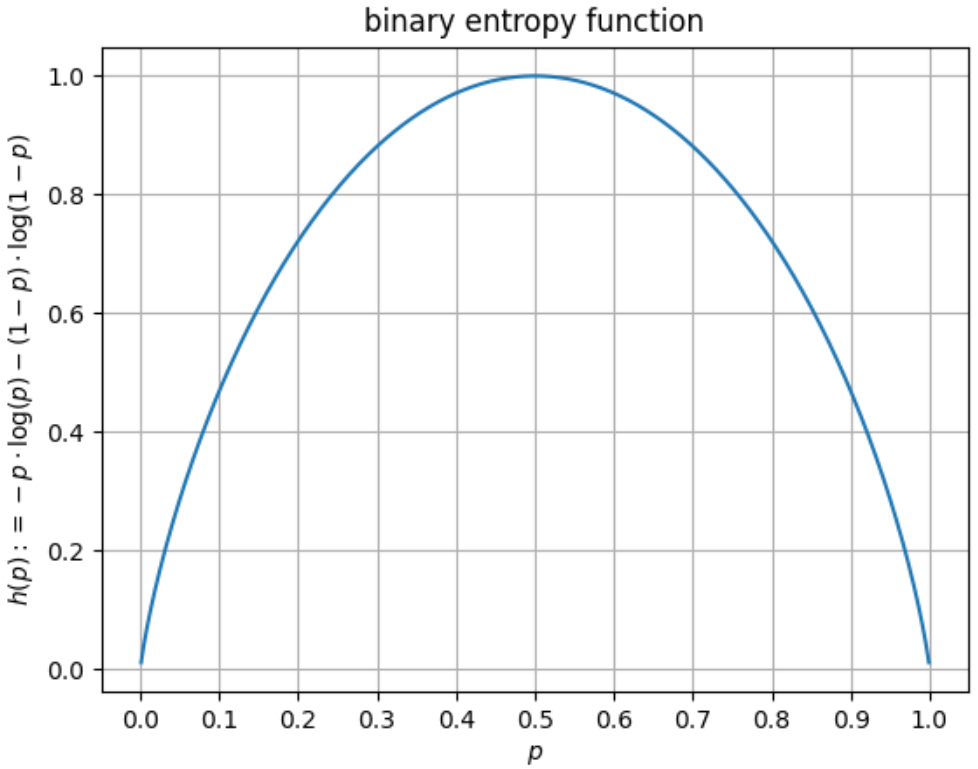
Bob has: Y = 0110100, receives Z=001
Which string does Alice have?

Invariant: even parity per circle!

Beyond the example



- Assume $X \in \{0,1\}^n$ and $Y = BSC_\delta(X)$. Then $H(X|Y) = h(\delta)n$
- Optimal amount of :$|leak_{EC}| \geq n \cdot h(p)$ with $h(p) := -\log p - (1-p)\log(1-p)$
- In practice:$|leak_{EC}| \geq n \cdot f \cdot h(p)$, $f \approx 1.2$
- Any error correction code can be adapted
  - LDPC codes
  - Polar codes
  - Turbo codes
  - Etc.
- Cascade protocol is a two-way error-correction protocol
  - Simple, easy to implement
  - Suboptimal leakage



A key with lots of entropy

K is 1001 bits long, with Shannon entropy H(K) ≈ 500 bits

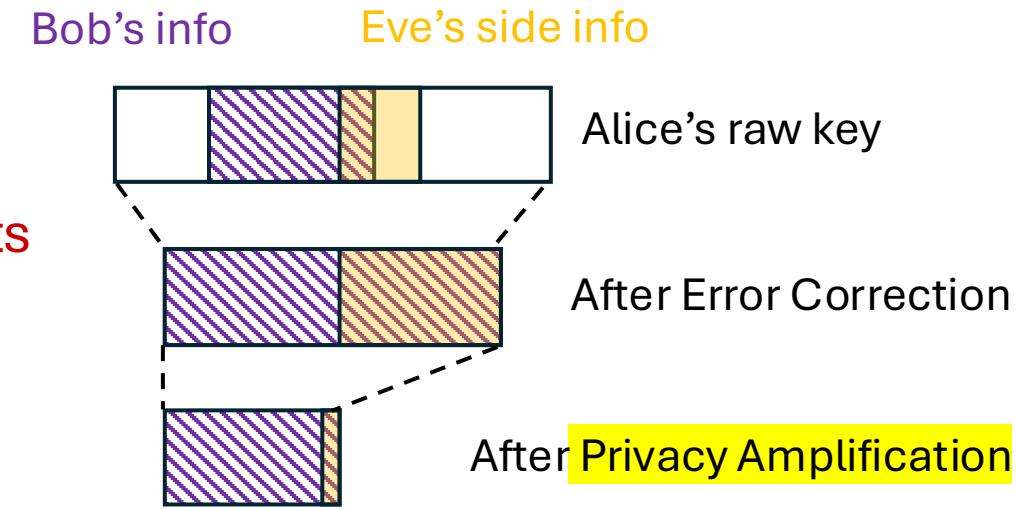
How well can Eve guess it?

with probability about 2^{-500}

with probability about 2^{-250}

somewhere between, depending on the distribution

with probability ½



https://app.wooclap.com/SZMYBFM

Shannon entropy is the wrong tool here

K = 0…0 with probability ½

K = uniform 1000-bit string with probability ½

H(K) ≈ 500 bits. Eve guesses right half the time.

Leftover Hash Lemma (LHL)

Theorem: For a joint distribution P_{XE} with $H_{min}(X|E) \geq k$, and $F: \mathcal{X} \to \{0,1\}^{\ell}$ chosen from a 2-universal family $\mathcal{F}$, it holds that

$$\frac{1}{2}\left\|P_{F(X)F\,E} - P_{U_\ell\,F}P_E\right\|_1 \leq \frac{1}{2}\,2^{-\frac{1}{2}(k-\ell)}$$

Proof for empty $E = \perp$:

$\frac{1}{2}\left\|P_Z - P_{U_m}\right\|_1 \leq \frac{1}{2}\sqrt{2^m Col(Z) - 1}$ (by Cauchy-Schwarz) where $Col(Z) := \sum_z P_Z(z)^2$

Then, $Col(F(X), F) = \Pr[\,(F(X), F) = (F'(X'), F')]$

$= \Pr[F = F']\,(\Pr[X = X'] + \Pr[X \neq X']\,\Pr[F(X) = F(X')|X \neq X']) \leq \frac{1}{|F|}(2^{-k} + 2^{-\ell})$

Then put it together using $m = |F|2^{\ell}$.

Side Information: Classical & Quantum

That proof assumed Eve's side information is classical

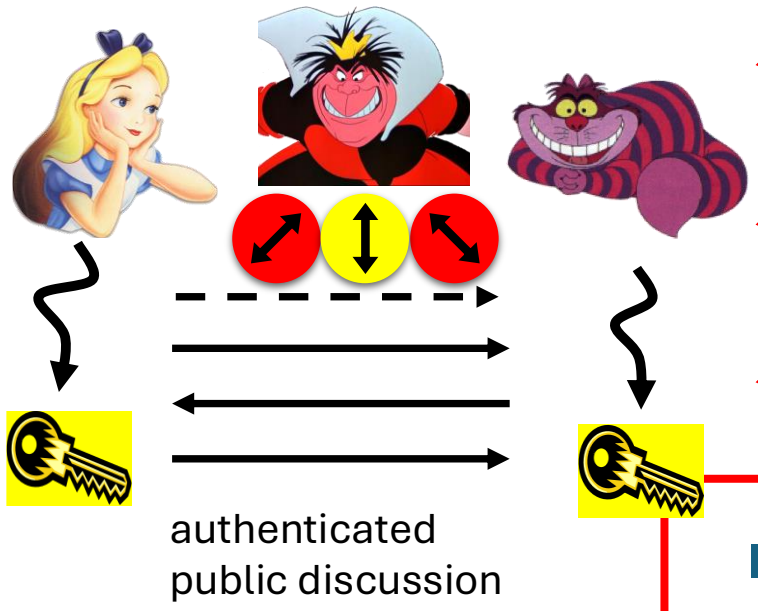
It does not automatically survive quantum side information

A good classical extractor need not be a good quantum one

See Renato’s lectures tomorrow how to deal with quantum side information

[Trevisan's extractor needed its proof redone; Renner, König, Terhal, de Wolf]

What will you learn this morning?



- ✓ Notions of security
- ✓ Shannon entropy and the one-time pad
- ✓ Secret-key agreement by public discussion
- ✓ Authentication
- ✓ Error correction and privacy amplification
- Back to the real world

The Cautionary Tale of Composability

Given all these ingredients, can we compose them?

Theorem A: QKD produces a provably secure key

Theorem B: the one-time pad is perfectly secure

Corollary: encrypt with a QKD key and you have a secure channel.

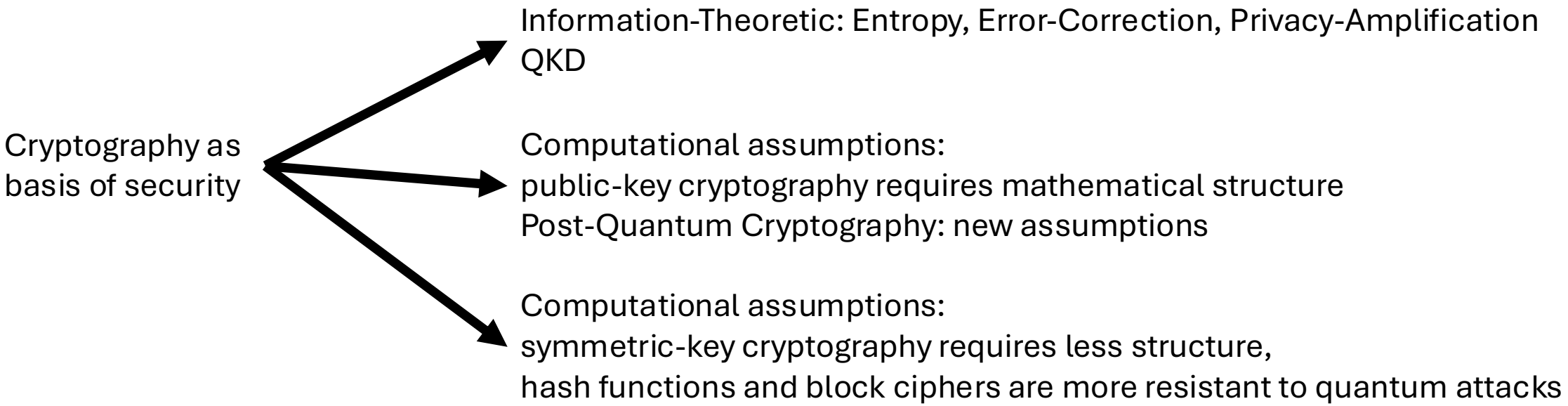
Not necessarily:

As Theorems were proven in different frameworks, they did not compose.

→ motivated Renato's composable security definition.

[König Renner Bariska Maurer 2007: https://doi.org/10.1103/PhysRevLett.98.140502 , Renner 2005]

QKD in the Cryptography Landscape



Remember the fork: new computational assumptions, or none at all

We spent almost three hours on the top branch.
Real-world cryptography mostly takes place in the other two.

How do these branches interact?

Where is QKD actually deployed?


Kathrin Hövelmanns; Sebastian Verschoor

- Their verdict: “in most cases, QKD provides very limited or no advantage”
- Positive about 2 out of 11 use cases,
  with long-lived, highly sensitive data; short link; one-time pad, not AES.
- Four problematic ways:
  1. Geneva 2007 election totals become public at closing; they needed authenticity only, and QKD already assumes an authenticated channel.
  2. QKD→AES: confidentiality now rests on computational and quantum assumptions,
     more than either pure option.
  3. QKD-protecting a Bitcoin private key: a quantum attacker derives it from the public key regardless.
  4. Backups encrypted in flight, stored in the clear.
- QKD can offer Post-Compromise Security (PCS): the ability to recover from a corruption of a party’s state and get back to a point of full confidentiality

[A Critical Analysis of Deployed Use Cases for Quantum Key Distribution and Comparison with Post-Quantum Cryptography, https://arxiv.org/abs/2502.04009]

What do the security agencies say?

NSA, 2020 "does not support the usage of QKD"

UK NCSC, Aug 2025 "will not support … for government or military applications"

FR · DE · NL · SE, 2024 "only in some niche use cases"

BSI, Jan 2026 "the BSI does not recommend QKD protocols at this time"

scope: national security systems · *"unless these limitations are overcome"*

Jul 2025 QKD protects "military and intelligence data"

[NSA, QKD and QC · NCSC, Quantum networking technologies · ANSSI/BSI/NLNCSA/SE, Position Paper on QKD · BSI TR-02102-1 §2.1 · Quantum Europe Strategy, COM(2025) 363 §2.4]

Two Types of Objection

engineering

cost
distance
trusted nodes
denial of service
no certification

contingent — all of it could be false in 2040

the model is not the device

"The actual security provided by a QKD system is not the theoretical unconditional security from the laws of physics … but rather the more limited security that can be achieved by hardware and engineering designs."

true of every proof and every box

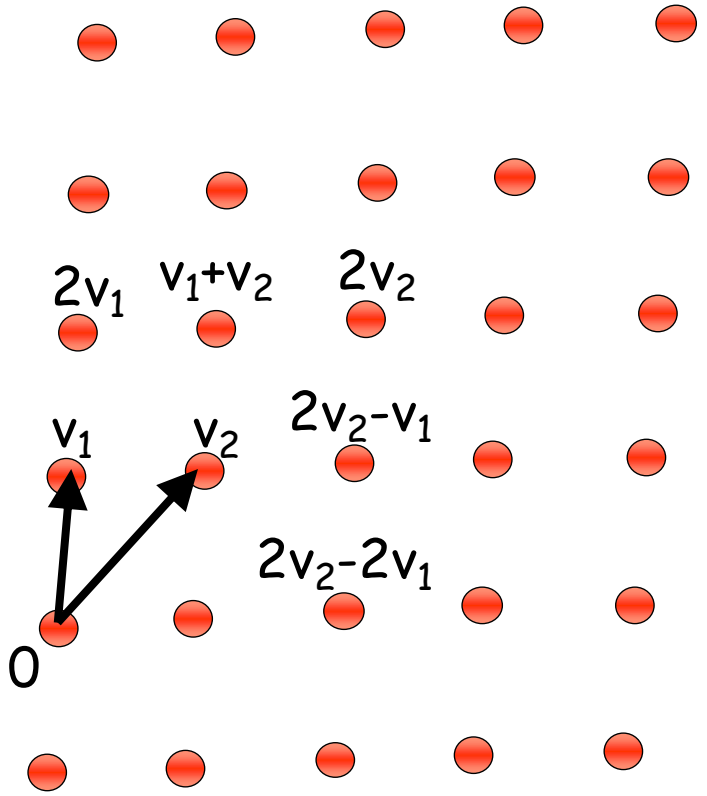
Information-theoretic security removes the computational assumption,
not the implementation.

[NSA, QKD and QC, limitation 4 · cf. ANSSI/BSI/NLNCSA/SE 2024 §2 · Renner & Wolf, arXiv:2307.15116, Fig. 4]

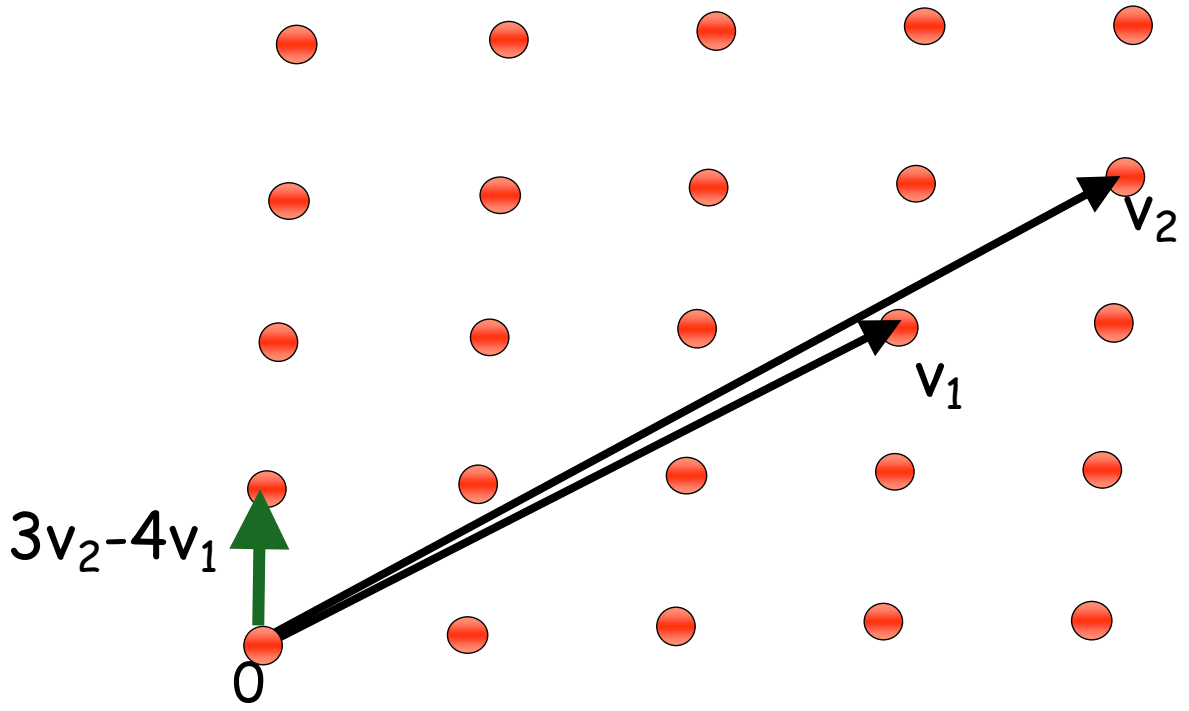
PQC Example: Lattice-Based Cryptography

- For any vectors $v_1, \ldots, v_n$ in $\mathbb{R}^n$, the lattice spanned by $v_1, \ldots, v_n$ is the set of points $L = \{a_1 v_1 + \cdots + a_n v_n | \; a_i \text{ integers}\}$

- Shortest Vector Problem (SVP): given a lattice L, find a shortest (nonzero) vector



PQC Example : Lattice-Based Cryptography

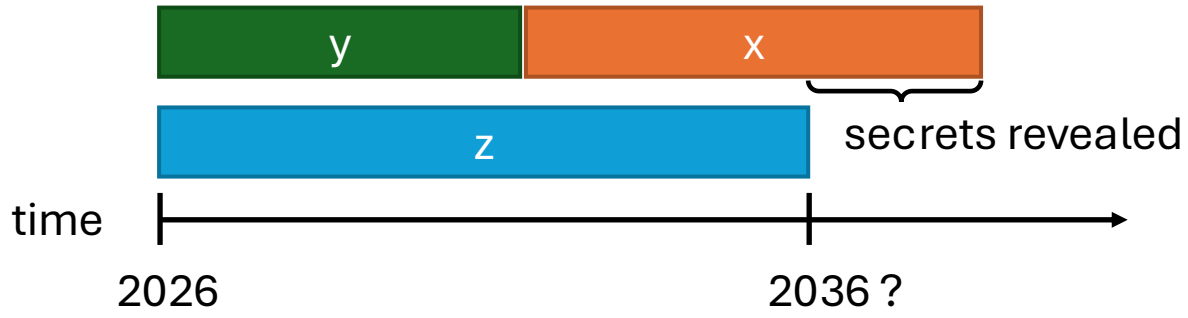


- Shortest Vector Problem (SVP): given a lattice, find a shortest (nonzero) vector
- no efficient (classical or quantum) algorithms known
- public-key encryption schemes can be built on the computational hardness of SVP

Harvest Now, Decrypt Later

Depends on:

- How long do you need to keep your secrets secure? (x years)
- How much time will it take to re-tool the existing infrastructure? (y years)
- How long will it take for a large-scale quantum computer to be built? (z years)
- **Theorem (Mosca):** If x + y > z, then worry.



- **Corollary:** If x > z or y > z, you are in big trouble!

Slide by
Michele Mosca

Transition to Post-Quantum Cryptography

- Large enough quantum computers break currently used public-key cryptosystems!

- 2018: NIST started a competition for quantum-safe public-key cryptography
- **Summer 2024:** new cryptographic standards (ML-KEM, ML-DSA, SLH-DSA, …), which are resistant against quantum attackers
- World-wide adoption is starting, but will take a long time
- Harvest-Now, Decrypt-Later is already a problem

QuSoft CWI

How to Combine QKD with Post-Quantum Crypto?

Post-Quantum Cryptography

- Can be deployed without quantum technologies
- Believed to be secure against quantum attacks of the future

+

Quantum Cryptography

- Requires some quantum technology (but no large-scale quantum computer)
- Typically no computational assumptions



Kathrin Hövelmanns; Daan Planken, CS, Sebastian Verschoor
QKD Oracles for Authenticated Key Exchange
https://arxiv.org/abs/2509.12478 - QCrypt 2026

→ Friday 11.00

Thank you for your attention!

Questions

https://app.wooclap.com/SZMYBFM

QuSoft
Research Center for Quantum Software

Quantum Computer Science

Universiteit van Amsterdam

Resources

Detailed links and references at the bottom of the slides

Lecture notes on information theory: by CS and Yfke Dulek
https://github.com/cschaffner/InformationTheory/raw/master/Script/InfTheory3.pdf

Nuggets of Shannon Information Theory: https://www.youtube.com/watch?v=we7T7NqTrlU
https://github.com/cschaffner/ITNuggets :
Shannon demos, Huffman, entropy of English etc.

https://arxiv.org/abs/2509.12478 : combining QKD and post-quantum cryptography

Email me at c.schaffner@uva.nl if you are missing anything.