Lecture 2: Unpredictability

- Pick up a copy of the syllabus if you don't have one already
- Lab 1 posted on piazza.com,
  due on gradescope.com on Monday 1/28 at 11pm

Bruce Schneier's 4 truths of computer security

1. *Attackers have the advantage*	"Complexity is the worst enemy of security, [and] the Internet is the most complex machine man has ever built by a lot."
2. *Interconnections → New vulnerabilities*	"The more we connect things to each other, the more vulnerabilities in one thing affect other things."
3. *Attacks scale*	"The Internet is a massive tool for making things more efficient. That's also true for attacking."
4. *Defense requires smart people who know what to do*	"Our computers are secure [only because] the engineers at Google, Apple, Microsoft spent a lot of time on this."

Source: www.schneier.com/essays/archives/2016/11/testimony_at_the_us_.html *kryptos = secret, hidden*

Cryptology

Cryptography

the art of making codes

Cryptanalysis

the art of breaking codes

Schneier's law: Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself can't break.

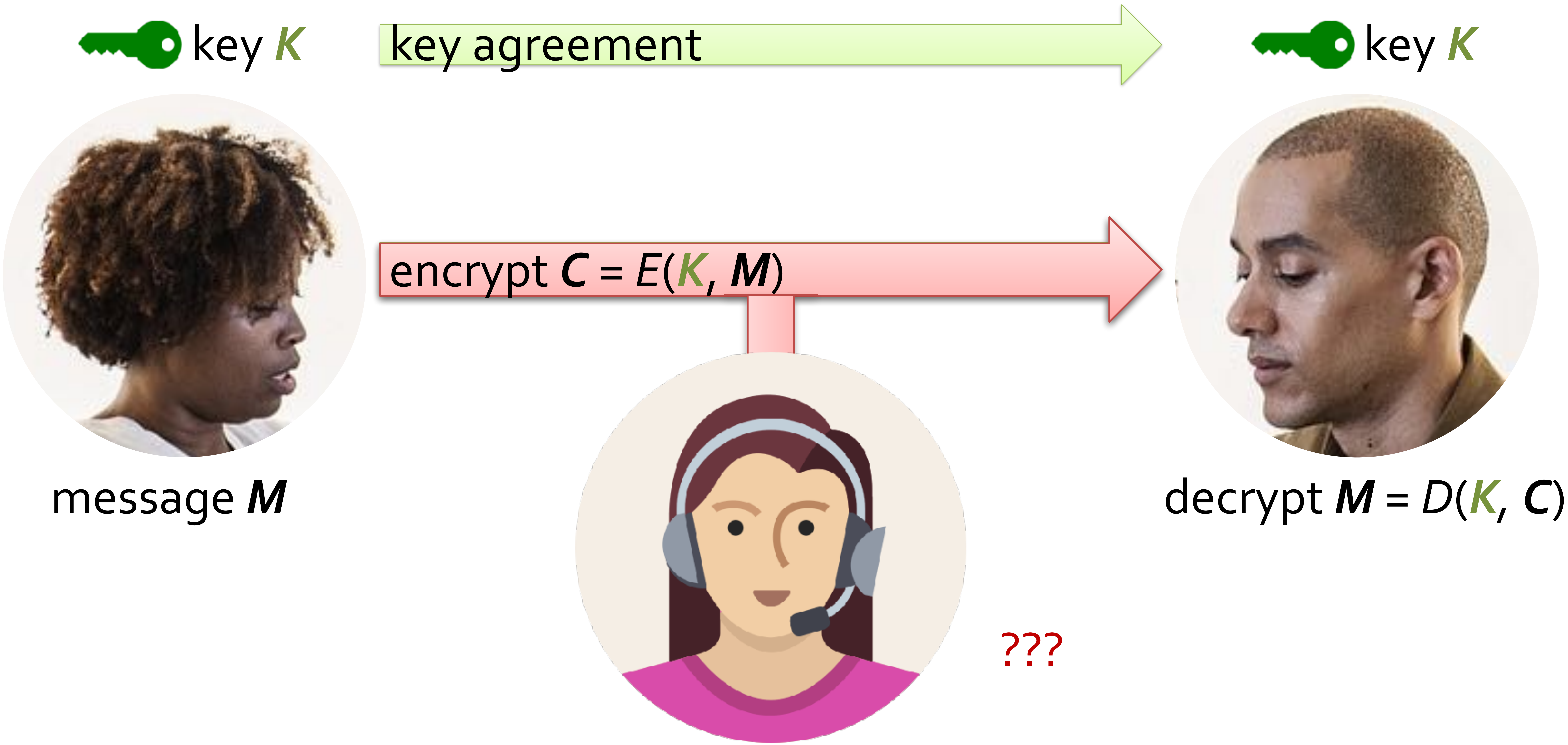
Course outline

1. Protecting data at rest
2. Attacking data at rest
3. Protecting data in transit
4. Crypto law and policy
5. Protecting data during use
6. Design + cryptanalysis of crypto building blocks

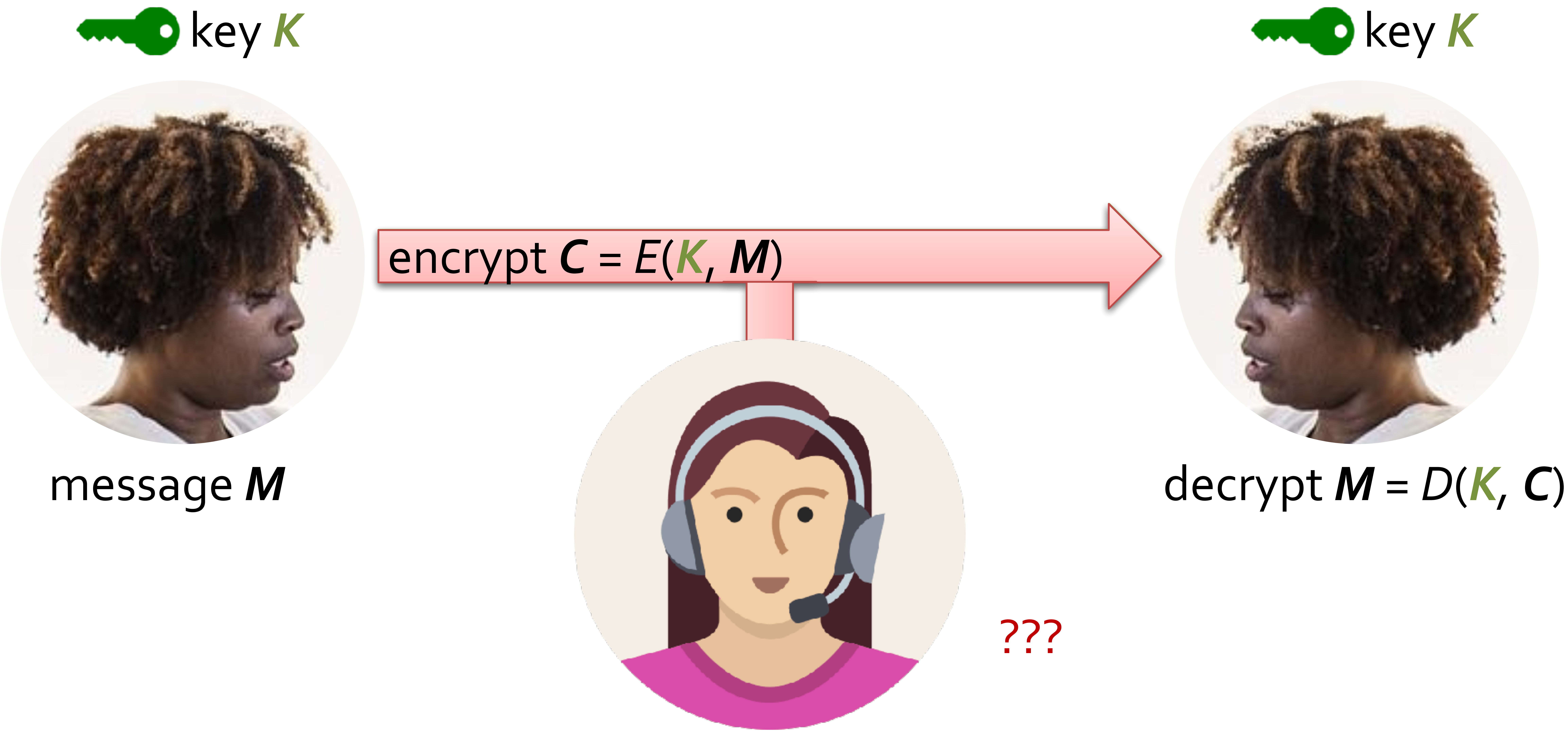
“Cryptography is about communication in the presence of an adversary.”

–Ron Rivest

Protecting data in transit



Part 1: Protecting data at rest



How can Alice encode messages so Eve can’t read them?

1. Substitute each character with another one
2. Write in a foreign language
3. Make it hard for Eve to determine where Alice wrote her message

Caesar cipher

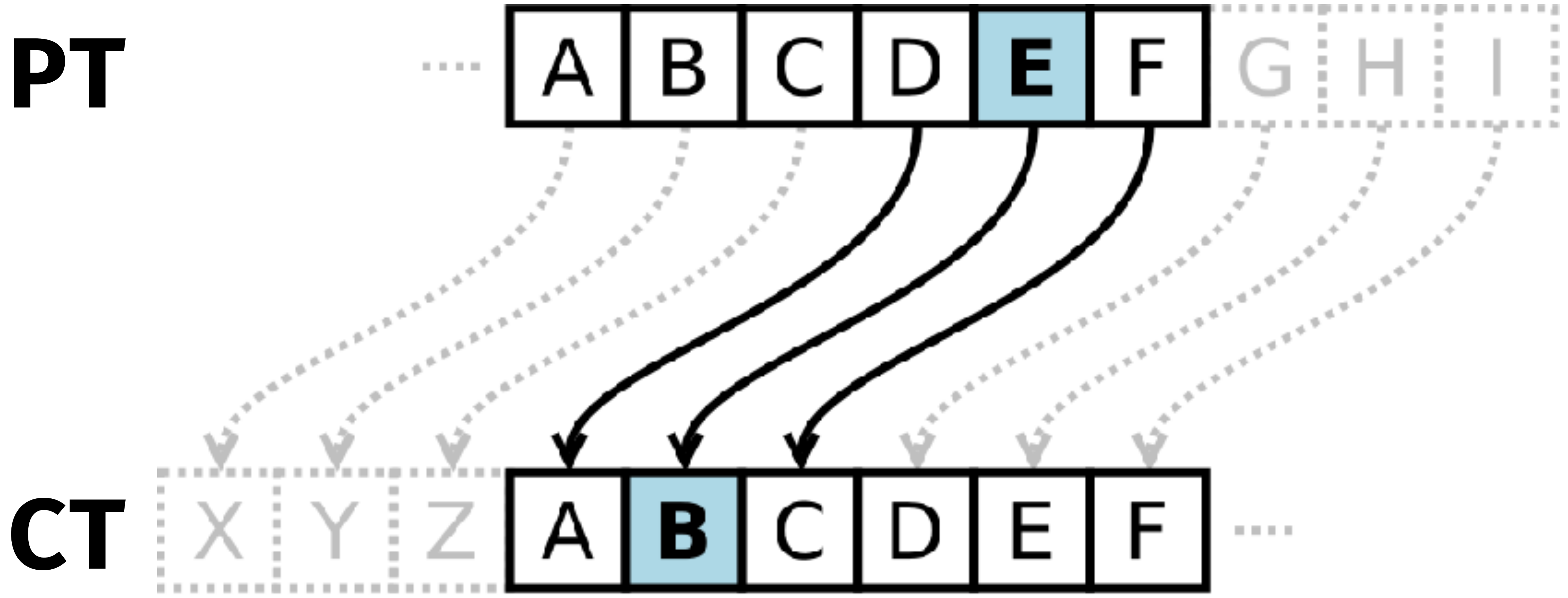


Image source: Wikipedia

- Encipher one character at a time
- Figure shows cipher with key $K = 3$
  - `one` ↦ `lkb`
  - `two` ↦ `qtl`
- Problem?
  - `three` ↦ `qeobb`
- How to resolve?

Binary representation of data

Quantities

- bit $\in \{0,1\}$
- byte $\in \{0,1\}^8$

Formats

- Raw bits (some of which are ASCII printable)
- Hex characters

Dec	Hx	Oct	Html	Chr
64	40	100	@	@
65	41	101	A	A
66	42	102	B	B
67	43	103	C	C
68	44	104	D	D
69	45	105	E	E
70	46	106	F	F

Reminder: Keep track of the format of a string during the labs!

If you compute an output whose length is double what you expected,
then you almost surely operated over a hex encoding rather than the raw string.

One time pad

⊕	0	1
0	0	1
1	1	0

- XOR function measures whether 2 inputs are identical
- OTP "masks" the message by applying a Caesar cipher independently to each bit

```
    message   0110 0110 1001
XOR key       0011 1100 1110
    cipher    0101 1010 0111
```

- XOR is a "lossless" function, so it is invertible (XOR is its own inverse)

```
    cipher    0101 1010 0111
XOR key       0011 1100 1110
    message   0110 0110 1001
```

- Drawbacks?
  - Key length == plaintext message length
  - No integrity: easy to manipulate ciphertext

How can Alice encode messages so Eve can’t read them?

1. Substitute each character with another one
2. Write in a foreign language
3. Make it hard for Eve to determine where Alice wrote her message


Slowenisch
Rumänisch
Collins Gem
Russian
COLLINS
SPANISH
Französisch - Deutsch
Deutsch - Französisch
ROMANIAN-ENGLISH
ENGLISH-ROMANIAN
spanisch deutsch
deutsch spanisch
PONS
Praxis
Klett

Goal 1: Unintelligible to Eve

Plain word	Encoded word
aba	nrq
abs	mbk
ace	ybd
act	wxv
add	jen
ado	hhg
aft	uxv
age	zmx
ago	dgs
aha	ase
aid	ktf
⋮	⋮
zip	cyu
zoo	dux

3-letter words 3 characters from random.org

Goal 2: Simple for Alice

- Fast + easy to compute ✗ Slow
- Secret key is small ✗ Big
  and easy to change
- Infinitely reusable ✗ Frequency analysis

Note that Alice chose her encoded words to be distinct. Why?

Is Alice's custom codebook secure?

X	Y
aba	nrq
abs	mbk
ace	ybd
act	wxv
add	jen
ado	hhg
aft	uxv
age	zmx
ago	dgs
aha	ase
aid	ktf
⋮	⋮
zip	cyu
zoo	dux

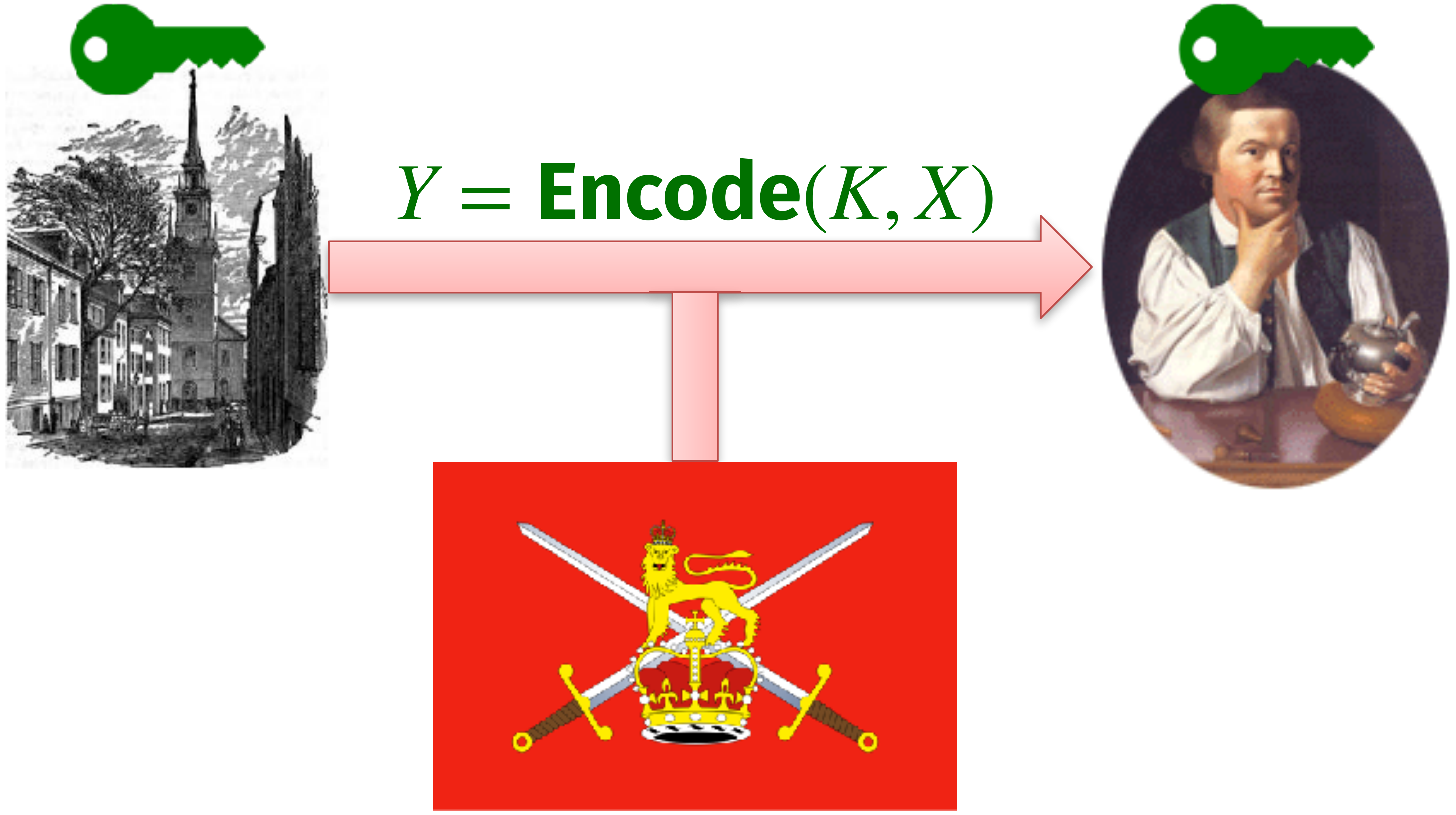
Alice has secret message X^*, sends $Y^* = \text{Encode}(X^*)$

Question: can Eve recover X^* when given:

1. Only Y^*
2. Above, plus many (X_i, Y_i) pairs chosen at random
3. Above, plus many (X_i, Y_i) pairs for X_i of Eve's choice
4. Above, plus Eve can choose the X_i one at a time, and adapt her choices based on the Y_i responses she receives
5. Above, plus Eve can also decipher Y_i of her choice

Upshot: security depends on the adversary's powers

When can we get away with a short key?



Two options:

Plain word	Encoded word
land	1
sea	2

Plain word	Encoded word
land	2
sea	1

Key == secret + unpredictable

Randomness ⇒ Unpredictability ⇒ Secrecy

Unpredictability

Suppose that Eve can adaptively make q queries into our codebook.

We call the codebook **unpredictable** if Eve has a very small chance to predict Enc(X^*) for any unqueried X^*.

Note:

- An unpredictable codebook is *almost* secure against an Eve that conducts attack #4 (choose X adaptively, get Y).
- But, we have not addressed frequency analysis yet. Unpredictability doesn't allow Eve to observe same X twice

X	*Y*
aba	nrq
abs	mbk
ace	ybd
act	wxv
add	jen
ado	hhg
aft	uxv
age	zmx
ago	dgs
aha	ase
aid	ktf
⋮	⋮
zip	cyu
zoo	dux

"If an adversary *A* has not **explicitly** queried a [perfect codebook] ***R*** on some point *X*, then the value of ***R***(*X*) is **completely random**... at least as far as *A* is concerned."

–*Jon Katz and Yehuda Lindell*, Introduction to Modern Cryptography

A crypto “Manhattan project”

- Imagine society spends an enormous effort to make a single codebook ***R*** and its inverse (so Alice can decipher her original message later)
- Can Alice use this codebook to protect her messages from Eve?
- Intuitively: no!
  - Eve can use the codebook too
  - Codebook is too large for Alice to carry around
  - Codebook’s input + output lengths may not suffice to encode Alice’s message


CLAUD: GILBERT

Block cipher

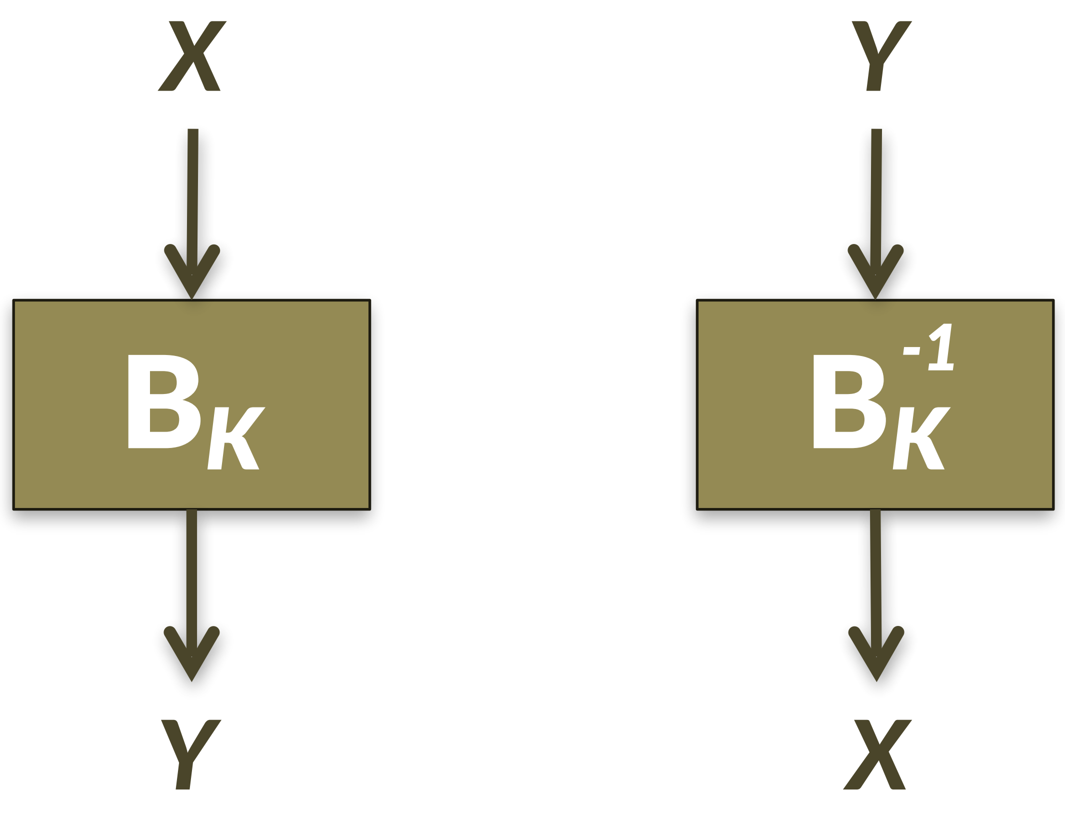
- Family of invertible permutations, indexed by a secret key

- Design goals

  1. **Simple** - built from native CPU operations like XOR, cyclic shifts, and table lookups

  2. **Makes no sense** - unpredictable

  3. **Simple to see that it makes no sense** - we have simple, convincing arguments that the cipher is unpredictable (remember Schneier's law!)



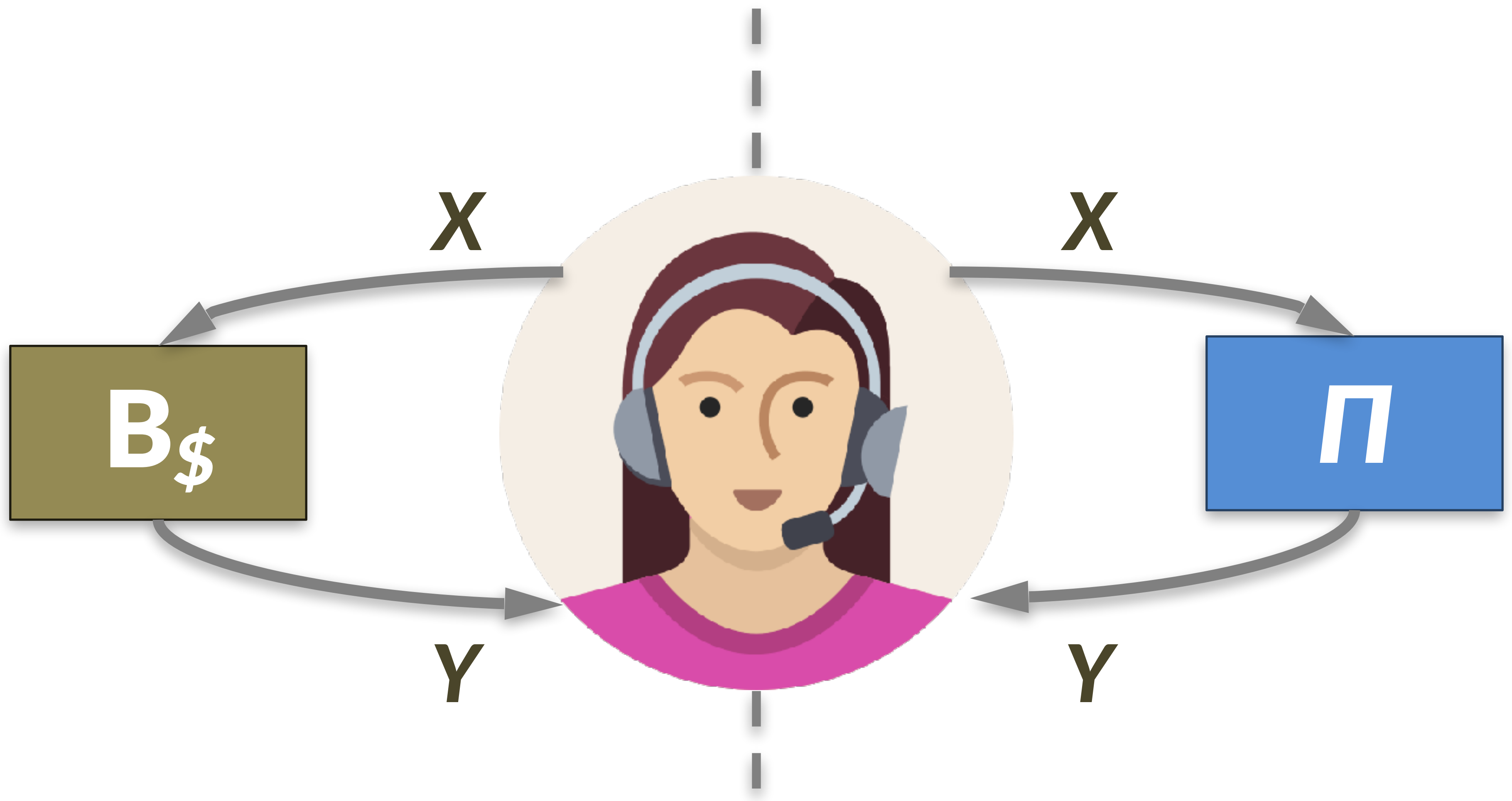
Cryptanalysis via brute force

- There is a large (but not infinite!) universe of possible keys
  - Alice's key  identifies the cipher that she decided to use
- *Brute force attack*:
  Eve can try all possible keys and test against an observed (X, Y) pair
- Alice's objectives
  - Make brute forcing infeasibly difficult
  - Ensure that cipher cannot be broken faster than brute force search

Game	Search size	Solved?
Connect 4	10^13	✓
Limit hold 'em	10^14	✓
Checkers	10^20	✓
Chess	10^50	
Modern crypto	*~10^70*	
No limit hold 'em	10^140	
Go (19 × 19)	10^171	

Security guarantee: pseudorandomness

- Let Π = truly random permutation made by a *secret* Manhattan project
- Goal: Eve cannot tell apart B_K and Π, so they are effectively the same



Security guarantee: strong pseudorandomness

- Let Π = truly random permutation made by a secret Manhattan project
- Goal: Eve cannot tell apart B_K and Π, so they are effectively the same

- …even if Eve gets access to both enciphering and deciphering
- Question: How do we build something that looks truly chaotic but isn't?

A crypto "Manhattan project"

- Imagine society spends an enormous effort to make a single codebook ***R*** and its inverse (so Alice can decipher her original message later)
- Can Alice use this codebook to protect her messages from Eve?
- ~~Intuitively: no~~! **Actually: Yes! (We can fix all of the problems below)**
  - Eve can use the codebook too
  - Codebook is too large for Alice to carry around
  - Codebook's input + output lengths may not suffice to encode Alice's message