MIT 6.875/6.5620/18.425

Foundations of Cryptography
Lecture 1

Course website: ***https://mit6875.github.io/***

Course Staff

Instructor:

Vinod Vaikuntanathan
(vinodv@mit)

TAs:

Neekon Vafa
(nvafa@mit)

Hanshen Xiao
(hsxiao@mit)

Chirag Falor
(cfalor@mit)

Course Website

https://mit6875.github.io/

MIT 6.5620/6.875/18.425 (Fall 2023)

Foundations of Cryptography

Course Description

The field of cryptography gives us a technical language to *define* important real-world problems such as security, privacy and integrity, a mathematical toolkit to *construct* mechanisms such as encryption, digital signatures, zero-knowledge proofs, homomorphic encryption and secure multiparty computation, and a complexity-theoretic framework to *prove* security using reductions. Together, they help us *enforce the rules of the road* in digital interactions.

The last few years have witnessed dramatic developments in the foundations of cryptography, as well as its applications to real-world privacy and security problems. For example, cryptography is abuzz with solutions to long-standing open problems such as fully homomorphic encryption and software obfuscation that use an abundance of data for public good without compromising security.

The course will explore the rich theory of cryptography all the way from the basics to the recent developments.

Prerequisites: This is an introductory, but fast-paced, graduate course, intended for beginning graduate students and upper level undergraduates in CS and Math. We will assume fluency in algorithms (equivalent to 6.046), complexity theory (equivalent to 6.045) and discrete probability (equivalent to 6.042). Mathematical maturity and an ease with writing mathematical proofs will be assumed starting from the first lecture.

Course Information

INSTRUCTOR	Vinod Vaikuntanathan Email: vinodv at csail dot mit dot edu
LOCATION AND TIME	**Monday and Wednesday 1:00-2:30pm** in **1-190**
TAs	**Chirag Falor** Email: cfalor at mit dot edu Office hours: TBD. **Neekon Vafa** Email: nvafa at mit dot edu Office hours: TBD. **Hanshen Xiao** Email: hsxiao at mit dot edu Office hours: TBD

Crypto ≠ Cryptocurrencies

6.5620 is *not* about 

Crypto $\neq$ Cryptocurrencies

6.5620 is *not* about 

6.5620 *is* about foundations:

- Digital Signatures
- Zero-knowledge Proofs
- Public-key Encryption
- Homomorphic Encryption
- Threshold Cryptography
- Pseudorandomness

Crypto $\neq$ Cryptocurrencies

6.5620 is *not* about 

Blockchains/
Cryptocurrencies

"Trustworthy"
machine learning

6.5620 *is* about foundations:

Digital Signatures

Zero-knowledge
Proofs

Public-key Encryption

Homomorphic
Encryption

Threshold
Cryptography

Pseudorandomness

The Intellectual Origins

Claude E. Shannon

"Communication Theory of Secrecy Systems" (1945)

preceded

"A Mathematical Theory of Communication" (1948)

*which founded **Information Theory***

The Intellectual Origins

Claude E. Shannon

"Communication Theory of Secrecy Systems" (1945)

preceded

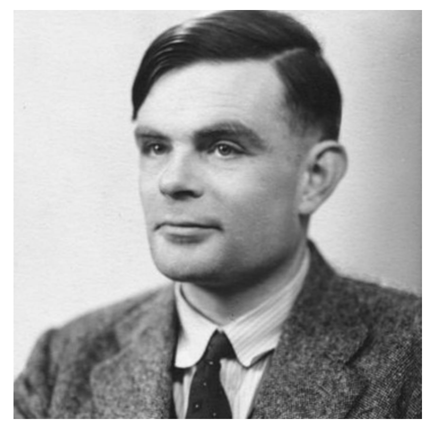
"A Mathematical Theory of Communication" (1948)

which founded ***Information Theory***

Cryptanalysis of the Enigma Machine (~1938-39)

"On Computable Numbers, with an Application to the Entscheidungsproblem" (1936)

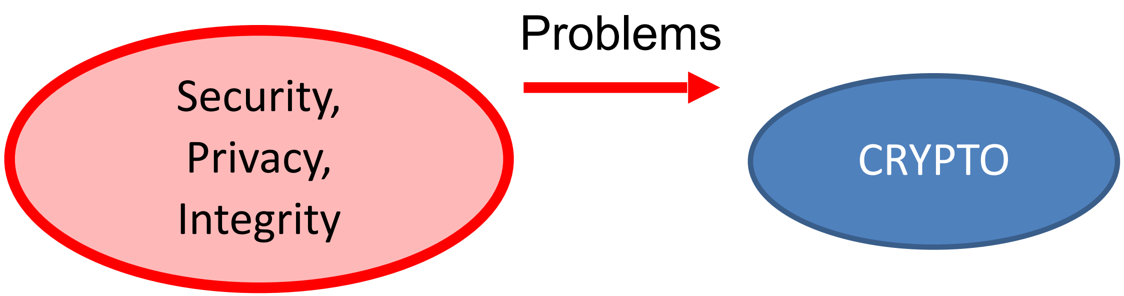
which gave birth to ***Computer Science***



Alan M. Turing

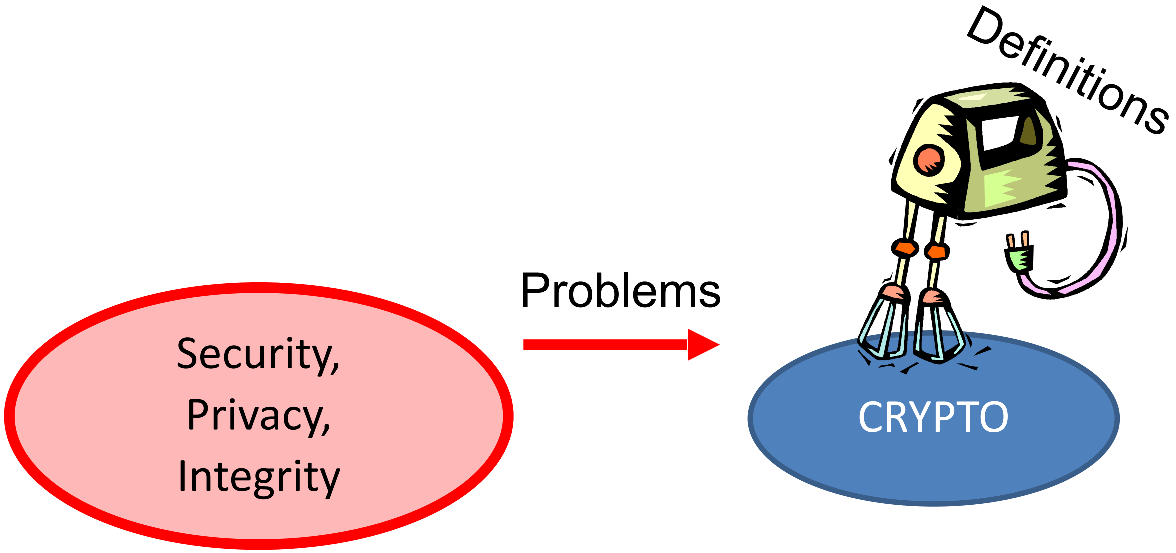
Modern Cryptography:

Practice to Theory and Back

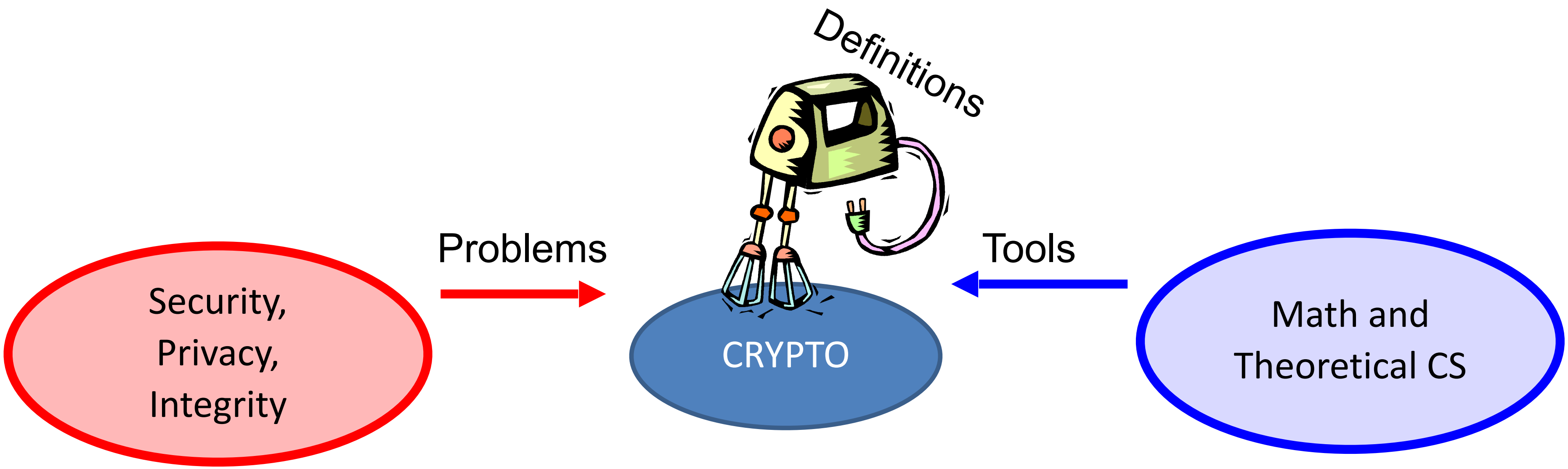


Modern Cryptography:

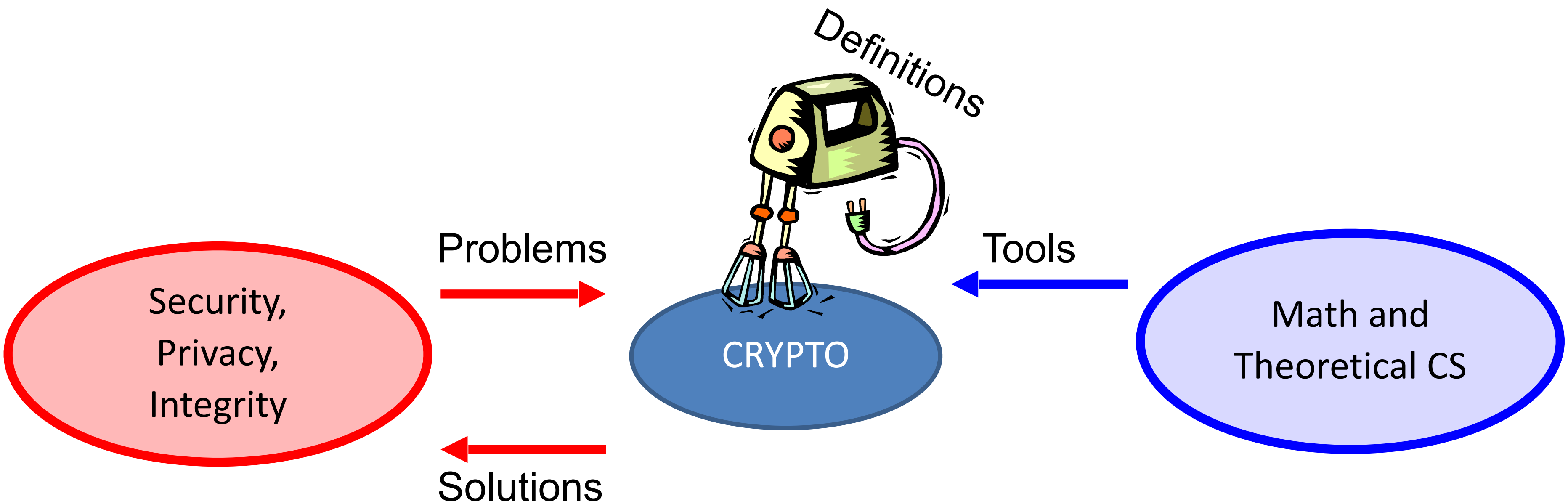
Practice to Theory and Back



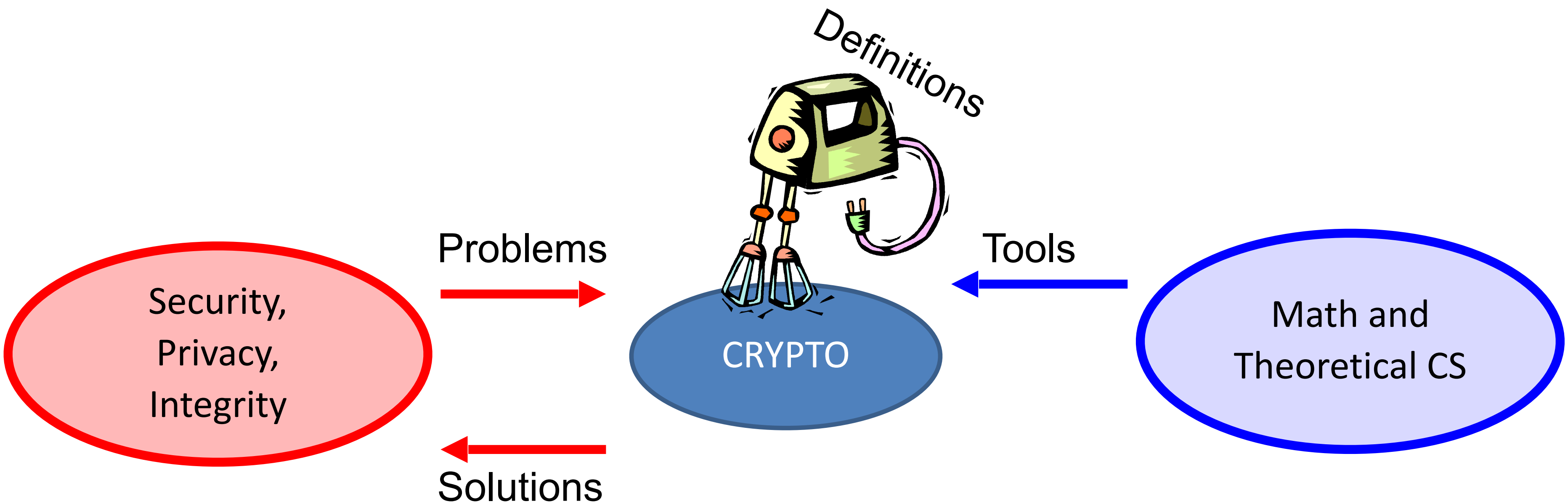
Modern Cryptography:
Practice to Theory and Back



Modern Cryptography:
Practice to Theory and Back



Modern Cryptography:
Practice to Theory and Back



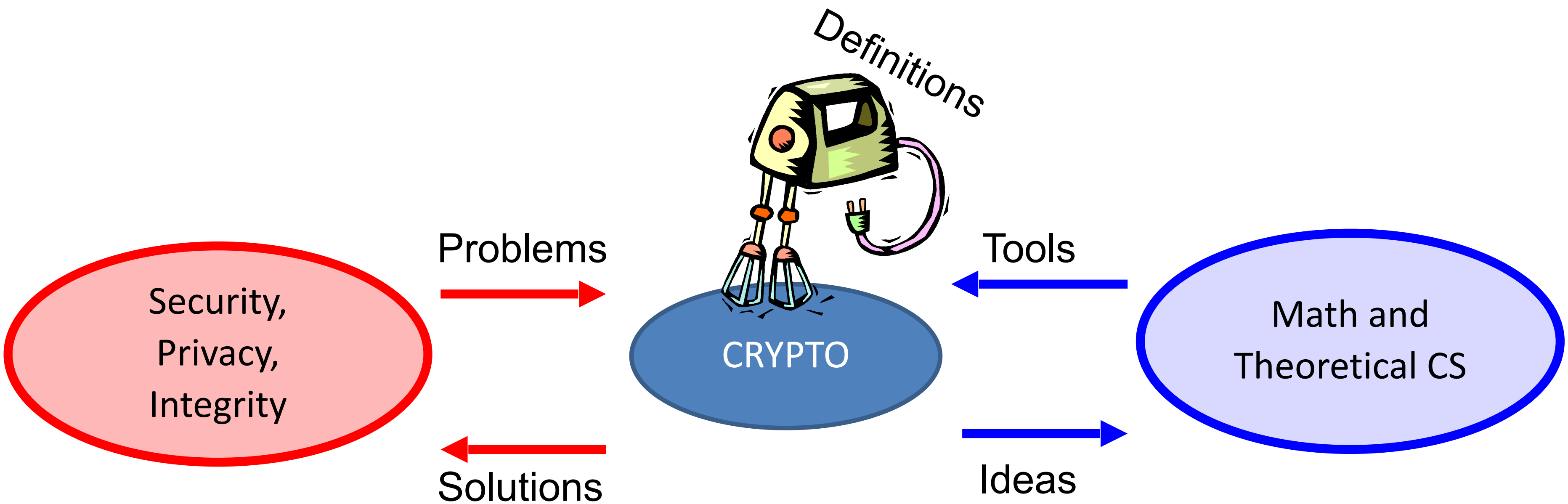
Encryption

Digital Signatures

Pseudorandom Functions

…

Modern Cryptography:
Practice to Theory and Back



Encryption

Digital Signatures

Pseudorandom Functions

…

Interactive Proofs

Probabilistically checkable Proofs

Locally decodable Codes

…

6.875/6.5620 Themes

1. The Omnipresent, Worst-case, Adversary. 

Central idea. model the adversary: what they know, what they can do, and what their goals are.

Definitions will be our friend.

If you cannot define something, you cannot achieve it.

6.875/6.5620 Themes

1. The Omnipresent, Worst-case, Adversary.

Central idea. model the adversary: what they know, what they can do, and what their goals are.

Definitions will be our friend.

If you cannot define something, you cannot achieve it.

A key takeaway from 6.875:

Cryptographic (or, adversarial) thinking.

6.875/6.5620 Themes

2. Computational Hardness will be our enabler.
(starting lecture 2)

Central theme: the cryptographic leash. Use computational hardness to “tame” the adversary.

6.875/6.5620 Themes

2. Computational Hardness will be our enabler.
(starting lecture 2)

Central theme: the cryptographic leash. Use computational hardness to "tame" the adversary.

A classical source of hard problems: number theory.

"Both Gauss and lesser mathematicians may be justified in rejoicing that there is one such science [number theory] at any rate, whose very remoteness from ordinary human activities should keep it gentle and clean"

[G. H. Hardy, "A Mathematician's Apology"]

6.875/6.5620 Themes

2. Computational Hardness will be our enabler.
(starting lecture 2)

Central theme: the cryptographic leash. Use computational hardness to “tame” the adversary.

A classical source of hard problems: number theory.

“Both Gauss and lesser mathematicians may be justified in rejoicing that there is one such science [number theory] at any rate, whose very remoteness from ordinary human activities should keep it gentle and clean”

[G. H. Hardy, “A Mathematician’s Apology”]

More recently: geometry, coding theory, combinatorics.

6.875/6.5620 Themes

2. Computational Hardness will be our enabler.
(starting lecture 2)

Central theme: the cryptographic leash. Use computational hardness to "tame" the adversary.

A classical source of hard problems: number theory.

"Both Gauss and lesser mathematicians may be justified in rejoicing that there is one such science [number theory] at any rate, whose very remoteness from ordinary human activities should keep it gentle and clean"

[G. H. Hardy, "A Mathematician's Apology"]

More recently: geometry, coding theory, combinatorics.

Cryptography is the science of useful hardness.

6.875 Themes

3. Security Proofs via Reductions.

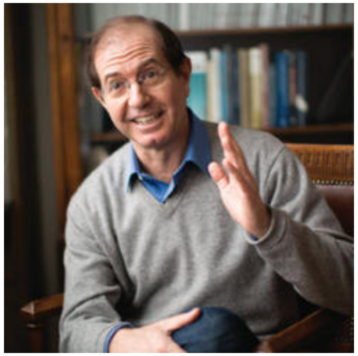
"If there is an (efficient) adversary that breaks scheme A w.r.t. definition D, then there is an (efficient) adversary that factors large numbers."

6.875 Themes

3. Security Proofs via Reductions.

"If there is an (efficient) adversary that breaks scheme A w.r.t. definition D, then there is an (efficient) adversary that factors large numbers."

"Science wins either way"

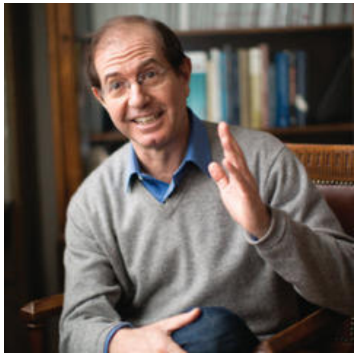


6.875 Themes

3. Security Proofs via Reductions.

"If there is an (efficient) adversary that breaks scheme A w.r.t. definition D, then there is an (efficient) adversary that factors large numbers."

"Science wins either way"



Our reductions will be probabilistic and significantly more involved than the NP-hardness reductions in, say, 6.045.

6.875 Topics

- Pseudorandomness
- Secret-key Encryption and Authentication
- Public-key Encryption and Digital Signatures
- Cryptographic Hashing
- Zero-knowledge Proofs
- Secure Multiparty Computation
- Private Information Retrieval
- Homomorphic Encryption
- Advanced topics:
  Threshold Cryptography, Program Obfuscation,
  Quantum Crypto, …

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

  Piazza for questions, Gradescope for psets.

  Piazza: https://piazza.com/class/lm5kwnurlj2573/

  Gradescope code: **B2BRD2**

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

  Piazza for questions, Gradescope for psets.

- **Homework (75%):** 6 psets, we will count your best 5.

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

  Piazza for questions, Gradescope for psets.

- **Homework (75%):** 6 psets, we will count your best 5.

  6.875 is on https://psetpartners.mit.edu

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

  Piazza for questions, Gradescope for psets.

- **Homework (75%):** 6 psets, we will count your best 5.
- **Midterm (20%):** Oct 25.

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

  Piazza for questions, Gradescope for psets.

- **Homework (75%):** 6 psets, we will count your best 5.
- **Midterm (20%):** Oct 25.
- **Class Participation (5%):** Lecture, Piazza.

Administrivia

- **Course website,** the central point of reference.

  https://mit6875.github.io

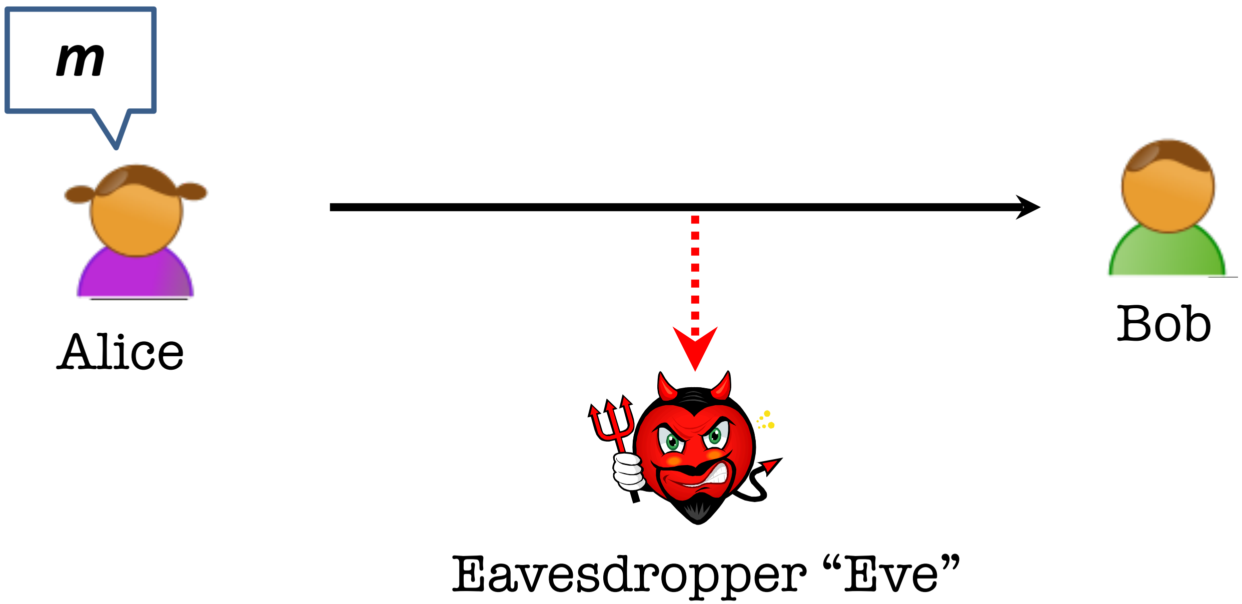
  Piazza for questions, Gradescope for psets.

- **Homework (75%):** 6 psets, we will count your best 5.
- **Midterm (20%):** Oct 25.
- **Class Participation (5%):** Lecture, Piazza.

- **Prereqs:** Algorithms, Probability & Discrete Math, but most of all, "mathematical maturity".

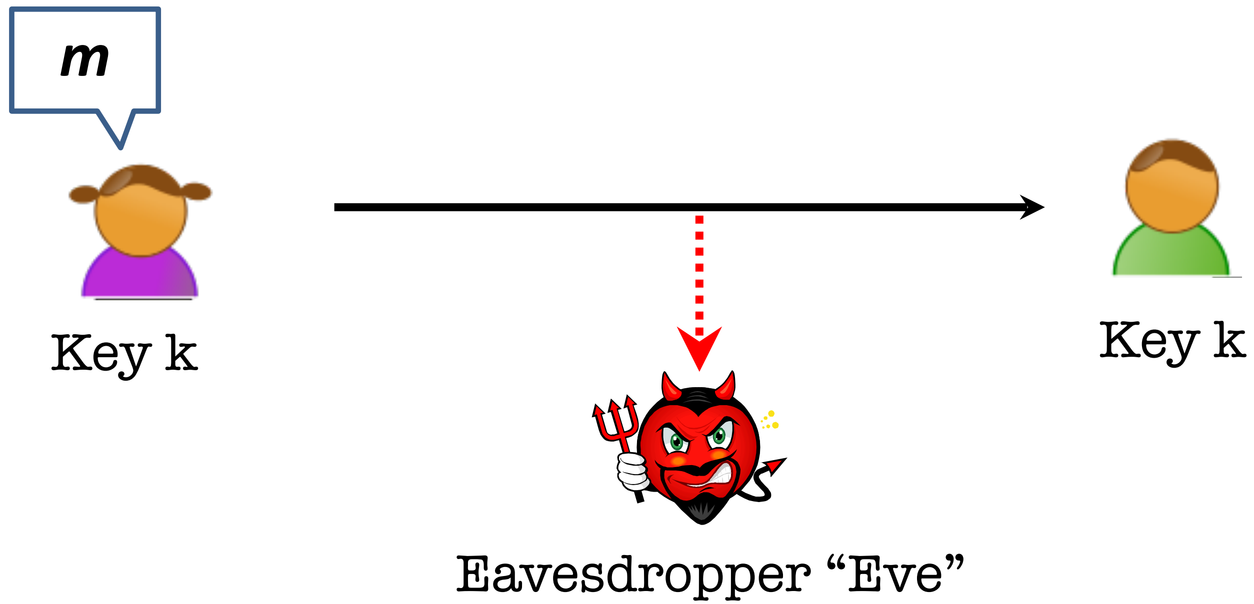
- **(Optional) special recitations:** 1. probability (this Friday), 2. basic complexity theory, 3. number theory.

Secure Communication



Alice wants to send a message m to Bob without revealing it to Eve.

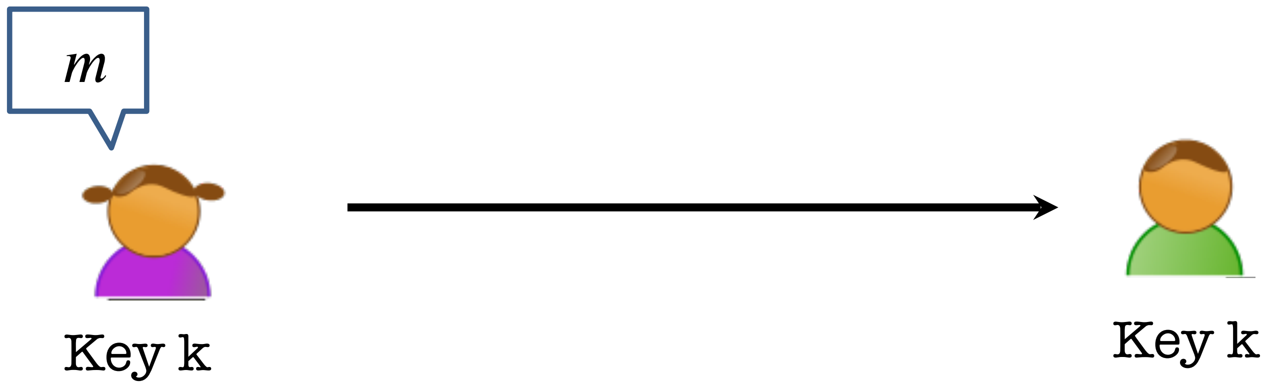
Secure Communication



SETUP: Alice and Bob meet beforehand to agree on a secret key k.

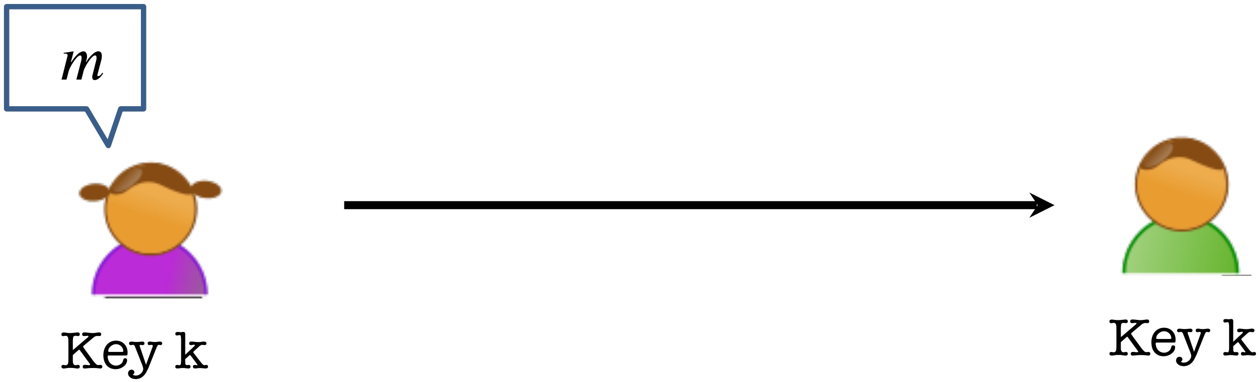
Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)



Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)

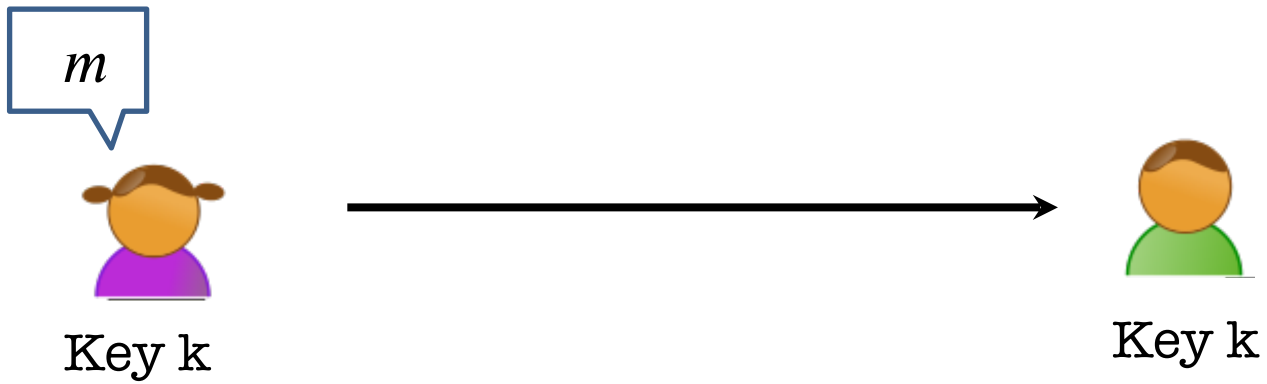


<u>Three (possibly probabilistic) polynomial-time algorithms:</u>

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}($

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)

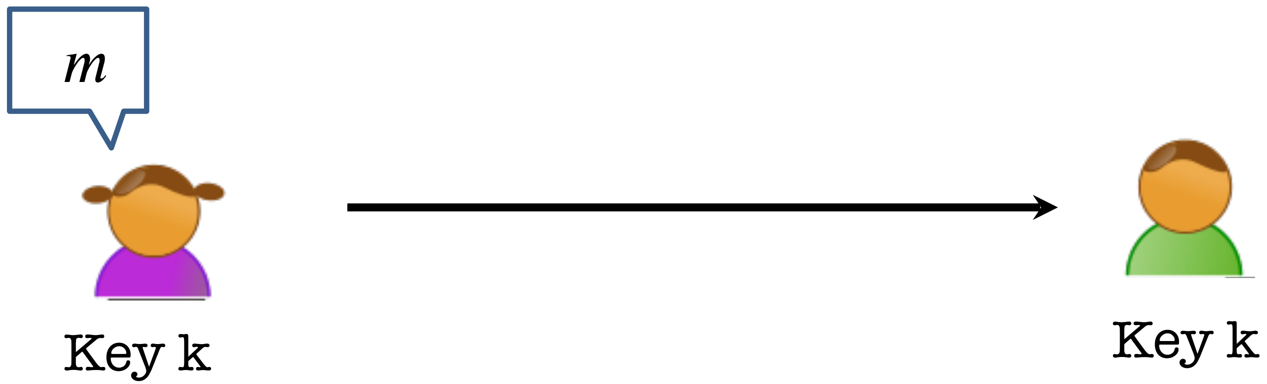


Three (possibly probabilistic) polynomial-time algorithms:

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}(1^n)$

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)



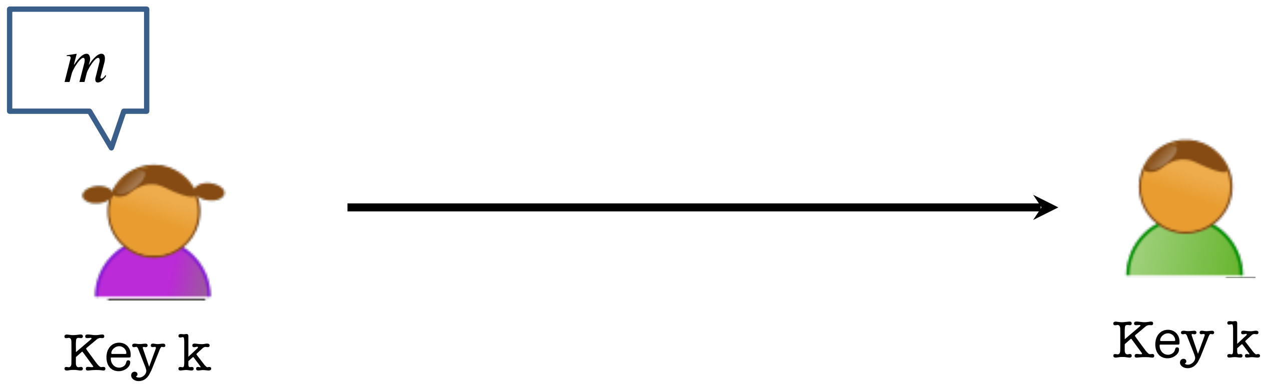
Three (possibly probabilistic) polynomial-time algorithms:

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}(1^n)$

  Has to be probabilistic

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)

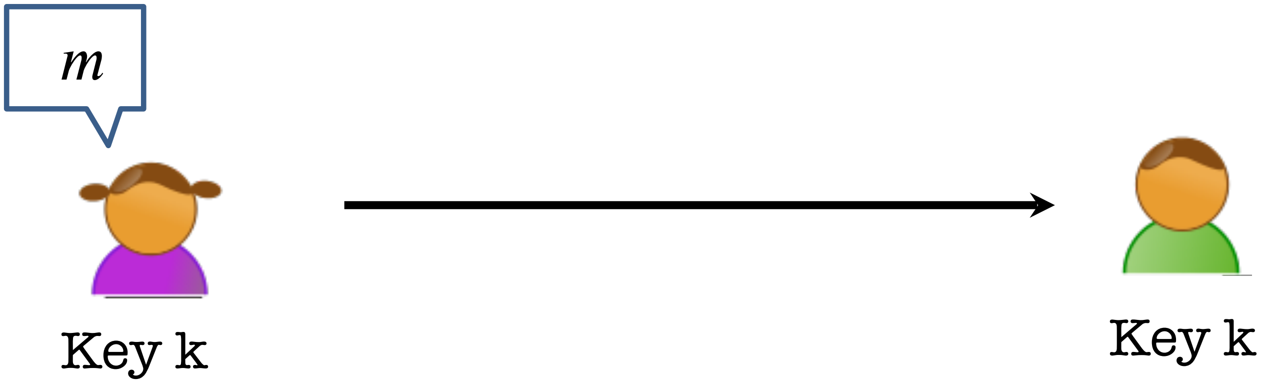


Three (possibly probabilistic) polynomial-time algorithms:

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}(1^n)$
  Has to be probabilistic
- **Encryption Algorithm Enc:** $c \leftarrow \mathrm{Enc}(k, m)$

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)

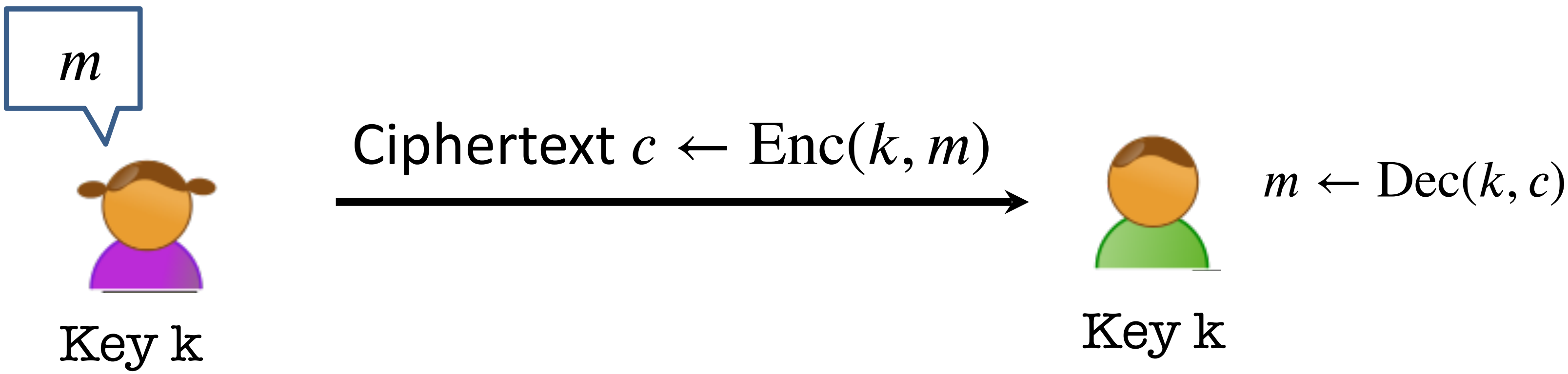


Three (possibly probabilistic) polynomial-time algorithms:

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}(1^n)$
  Has to be probabilistic
- **Encryption Algorithm Enc:** $c \leftarrow \mathrm{Enc}(k, m)$
- **Decryption Algorithm Dec:** $m \leftarrow \mathrm{Dec}(k, c)$

Key Notion: Secret-key Encryption

(or Symmetric-key Encryption)



Three (possibly probabilistic) polynomial-time algorithms:

- **Key Generation Algorithm Gen:** $k \leftarrow \mathrm{Gen}(1^n)$

  Has to be probabilistic

- **Encryption Algorithm Enc:** $c \leftarrow \mathrm{Enc}(k, m)$

- **Decryption Algorithm Dec:** $m \leftarrow \mathrm{Dec}(k, c)$

The Worst-case Adversary

- An arbitrary computationally *unbounded* algorithm **EVE**.*

The Worst-case Adversary

- An arbitrary computationally *unbounded* algorithm **EVE**.*

- Knows Alice and Bob's algorithms Gen, Enc and Dec but does not know the key nor their internal randomness.
  (*Kerckhoff's principle or Shannon's maxim*)

The Worst-case Adversary

- An arbitrary computationally *unbounded* algorithm **EVE**.*

- Knows Alice and Bob's algorithms Gen, Enc and Dec but does not know the key nor their internal randomness.
  (*Kerckhoff's principle or Shannon's maxim*)

- Can see the ciphertexts going through the channel
  (but cannot modify them… we will come to that later)

The Worst-case Adversary

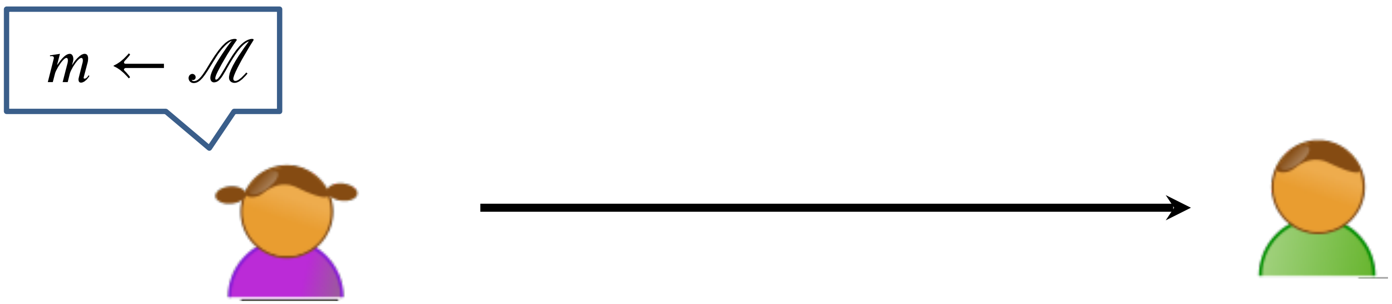
- An arbitrary computationally *unbounded* algorithm **EVE**.*

- Knows Alice and Bob's algorithms Gen, Enc and Dec but does not know the key nor their internal randomness.
  (*Kerckhoff's principle or Shannon's maxim*)

- Can see the ciphertexts going through the channel
  (but cannot modify them… we will come to that later)

Security Definition: What is she trying to learn?

Shannon's Perfect Secrecy Definition

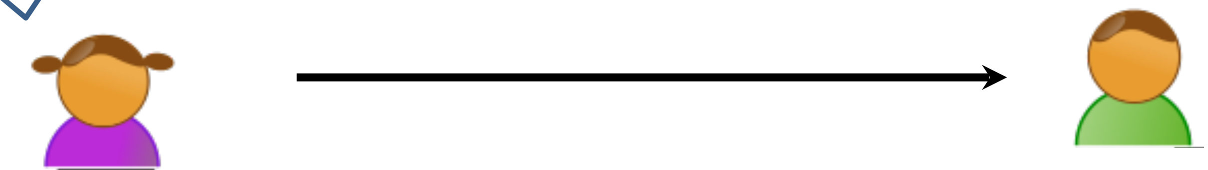


$\Pr[$

Shannon's Perfect Secrecy Definition

$m \leftarrow \mathscr{M}$

Message space (probability distribution) $\mathscr{M}$



$\Pr\big[\mathscr{M}$

Shannon's Perfect Secrecy Definition

$m \leftarrow \mathscr{M}$

Message space (probability distribution) $\mathscr{M}$

Key k $\leftarrow \mathscr{K}$

Key k $\leftarrow \mathscr{K}$

Key space $\mathscr{K}$

$\Pr[$

Shannon's Perfect Secrecy Definition

$m \leftarrow \mathscr{M}$

Message space (probability distribution) $\mathscr{M}$

$c \leftarrow \mathrm{Enc}(k, m)$

Ciphertext space $\mathscr{C}$

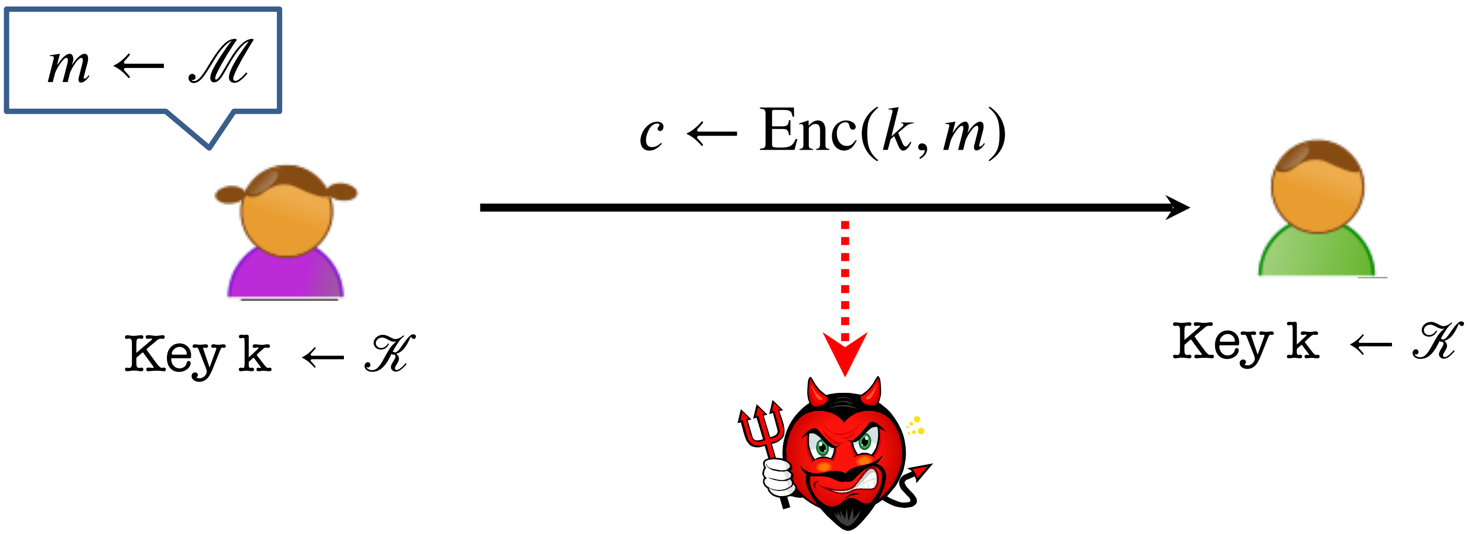
Key k $\leftarrow \mathscr{K}$

Key k $\leftarrow \mathscr{K}$

Key space $\mathscr{K}$

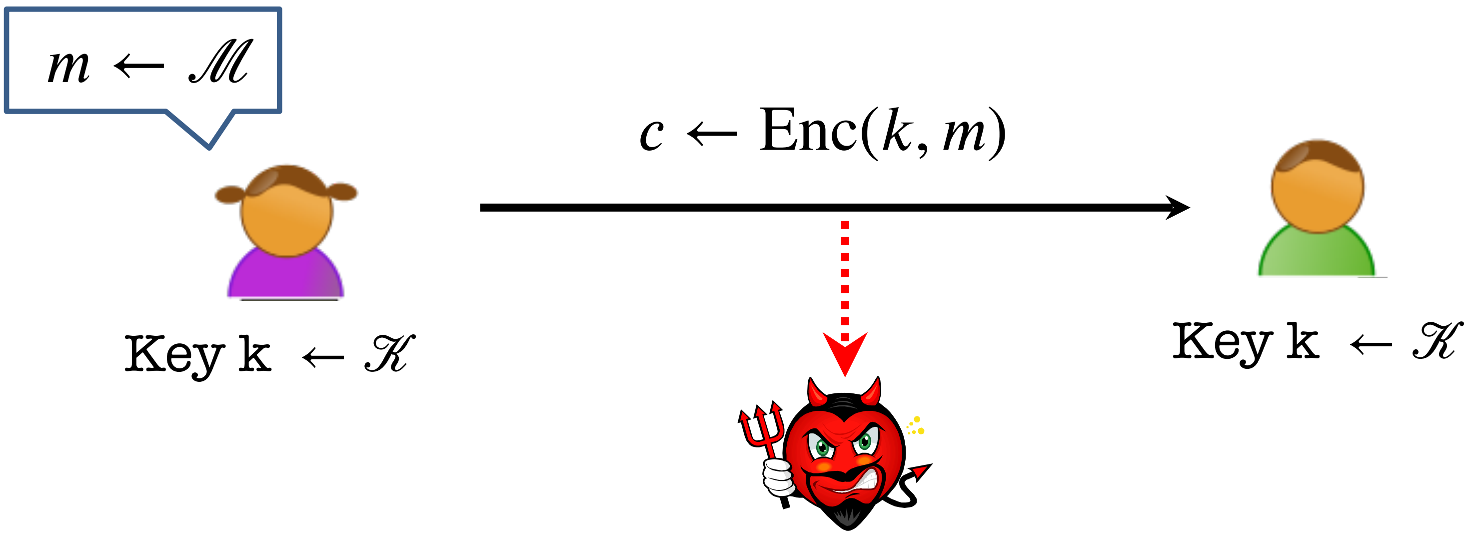
$\mathrm{Pr}\big[$

Shannon's Perfect Secrecy Definition



$\Pr\big[\mathcal{A}$

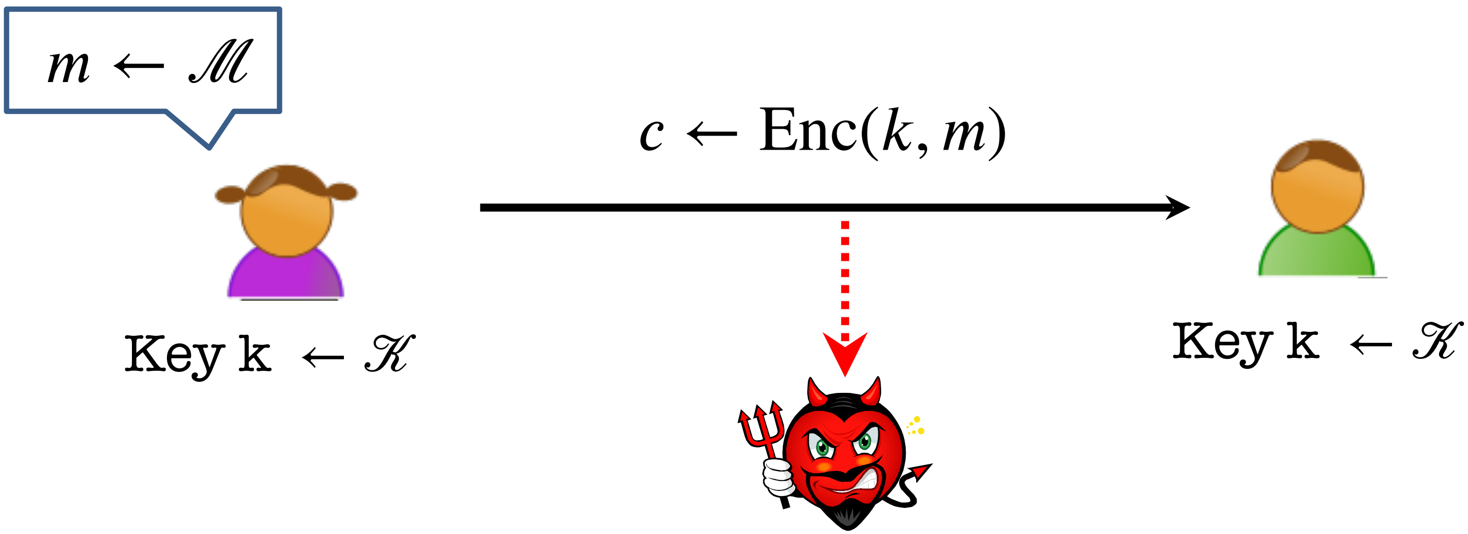
Shannon's Perfect Secrecy Definition



IDEA: A-posteriori = A-priori

$\Pr\Big[\mathscr{M}$

Shannon’s Perfect Secrecy Definition



IDEA: A-posteriori = A-priori

$$\underbrace{\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right]}_{\text{A-posteriori}} = \Pr\left[\mathscr{M}\right.$$

Shannon's Perfect Secrecy Definition

$m \leftarrow \mathscr{M}$

Key k $\leftarrow \mathscr{K}$

Key k $\leftarrow \mathscr{K}$

IDEA: A-posteriori = A-priori

$$\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right] = \Pr\left[\mathscr{M} = m\right]$$

A-posteriori A-priori

Shannon's Perfect Secrecy Definition

$m \leftarrow \mathscr{M}$

Key k $\leftarrow \mathscr{K}$

Key k $\leftarrow \mathscr{K}$

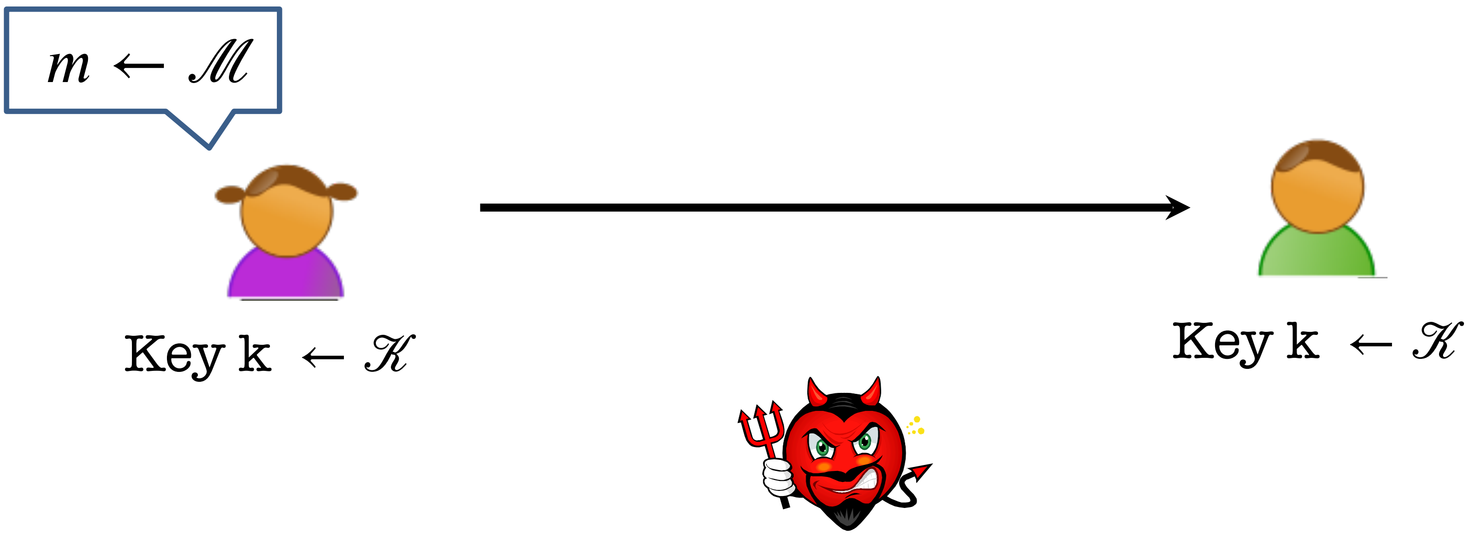
IDEA: A-posteriori = A-priori

$$\forall \mathscr{M}\ \forall m \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$$

$$\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right] = \Pr\left[\mathscr{M} = m\right]$$

A-posteriori A-priori

Shannon's Perfect Secrecy Definition



IDEA: A-posteriori = A-priori

$\forall \mathscr{M}\ \forall m \in \mathrm{Supp}(\mathscr{M}),\ \forall c \in \mathrm{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right] = \Pr\left[\mathscr{M} = m\right]$$

A-posteriori A-priori

Perfect Indistinguishability Definition

Perfect indistinguishability: a Turing test

$$\forall \mathscr{M} \ \forall m,\ m' \in \ \mathrm{Supp}(\mathscr{M}),$$

Perfect Indistinguishability Definition

Perfect indistinguishability: a Turing test

$\forall \mathscr{M} \ \forall m, \ m' \in \ \mathrm{Supp}(\mathscr{M}),$

World 0:

k $\leftarrow$ $\mathcal{K}$

$c = E(k, m)$

World 1:

k $\leftarrow$ K

$c' = E(k, m')$

Perfect Indistinguishability Definition

Perfect indistinguishability: a Turing test

$\forall \mathscr{M} \ \forall m, \ m' \in \ \mathrm{Supp}(\mathscr{M}),$

World 0:

k $\leftarrow$ $\mathcal{K}$

$c = E(k, m)$

World 1:

k $\leftarrow$ K

$c' = E(k, m')$

is a **distinguisher** (that gets c and tries to guess which world she's in)

Perfect Indistinguishability Definition

<u>Perfect indistinguishability</u>: a Turing test

$$\forall \mathscr{M} \;\; \forall m,\; m' \in \mathrm{Supp}(\mathscr{M}),\;\; c \in \mathrm{Supp}(\mathscr{C}):$$

$$\Pr[E(\mathsf{K}, m) = c] = \Pr[E(\mathsf{K}, m') = c]$$

World 0:

$\mathsf{k} \leftarrow \mathcal{K}$

$c = E(k, m)$

World 1:

$\mathsf{k} \leftarrow \mathrm{K}$

$c' = E(k, m')$

is a **distinguisher** (that gets c and tries to guess which world she's in)

Perfect Indistinguishability Definition

Perfect indistinguishability: a Turing test

$$\forall \mathcal{M}\ \forall m,\ m' \in \text{Supp}(\mathcal{M}),\quad c \in \text{Supp}(\mathcal{C}):$$
$$\Pr[E(\mathsf{K}, m) = c] = \Pr[E(\mathsf{K}, m') = c]$$

World 0:

$\mathsf{k} \leftarrow \mathcal{K}$

$c = E(k, m)$

World 1:

$\mathsf{k} \leftarrow \mathrm{K}$

$c' = E(k, m')$

is a **distinguisher** (that gets c and tries to guess which world she's in)

The Two Definitions are Equivalent

THEOREM: An encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ satisfies perfect secrecy IFF it satisfies perfect indistinguishability.

The Two Definitions are Equivalent

THEOREM: An encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ satisfies perfect secrecy IFF it satisfies perfect indistinguishability.

PROOF: Simple use of conditional probabilities.

A simple observation

(SEC): $\forall \mathscr{M}\ \forall m \in \mathrm{Supp}(\mathscr{M}),\ \forall c \in \mathrm{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right] = \Pr\left[\mathscr{M} = m\right]$$

A simple observation

(SEC): $\forall \mathscr{M}\ \forall m \in \mathrm{Supp}(\mathscr{M}),\ \forall c \in \mathrm{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \mid Enc\left(\mathscr{K}, \mathscr{M}\right) = c\right] = \Pr\left[\mathscr{M} = m\right]$$

Observation: SEC is equivalent to saying that the random variables $\mathscr{M}$ and $\mathscr{C} := Enc\left(\mathscr{K}, \mathscr{M}\right)$ are independent.

Proof Part 1. Indistinguishability $\Longrightarrow$ Secrecy

WE KNOW (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \ \mathrm{Supp}(\mathscr{M}),\ \forall c \in \mathrm{Supp}(\mathscr{C}),$

$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right] \qquad \boldsymbol{= \alpha}$$

WE WANT (SEC): $\forall \mathscr{M}\ \forall m \in \ \mathrm{Supp}(\mathscr{M}),\ \forall c \in \mathrm{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \mid \mathscr{C} = c\right] = \Pr\left[\mathscr{M} = m\right]$$

Proof Part 1. Indistinguishability $\Longrightarrow$ Secrecy

WE KNOW (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right] \qquad = \boldsymbol{\alpha}$$

WE WANT (SEC): $\forall \mathscr{M}\ \forall m \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \,|\, \mathscr{C} = c\right] = \Pr\left[\mathscr{M} = m\right]$$

Proof: By the observation from last slide, SEC is true if and only if
$\Pr\left[\mathscr{C} = c \,|\, \mathscr{M} = m\right] = \Pr\left[\mathscr{C} = c\right]$ (by independence of $\mathscr{M}$ and $\mathscr{C}$.)

Proof Part 1. Indistinguishability $\Longrightarrow$ Secrecy

WE KNOW (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C})$,

$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right] \qquad = \boldsymbol{\alpha}$$

WE WANT (SEC): $\forall \mathscr{M}\ \forall m \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C})$,

$$\Pr\left[\mathscr{M} = m \mid \mathscr{C} = c\right] = \Pr\left[\mathscr{M} = m\right]$$

Proof: By the observation from last slide, SEC is true if and only if $\Pr\left[\mathscr{C} = c \mid \mathscr{M} = m\right] = \Pr\left[\mathscr{C} = c\right]$ (by independence of $\mathscr{M}$ and $\mathscr{C}$.)

This means that $\Pr\left[\mathscr{C} = c \mid \mathscr{M} = m\right] = \Pr\left[Enc\left(\mathscr{K}, m\right) = c\right]$ is a number (say α_c) that does not depend on m.

Proof Part 1. Indistinguishability $\Longrightarrow$ Secrecy

WE KNOW (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \text{ Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right] \qquad = \boldsymbol{\alpha}$$

WE WANT (SEC): $\forall \mathscr{M}\ \forall m \in \text{ Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \,|\, \mathscr{C} = c\right] = \Pr\left[\mathscr{M} = m\right]$$

Proof: By the observation from last slide, SEC is true if and only if $\Pr\left[\mathscr{C} = c \,|\, \mathscr{M} = m\right] = \Pr\left[\mathscr{C} = c\right]$ (by independence of $\mathscr{M}$ and $\mathscr{C}$.)

This means that $\Pr\left[\mathscr{C} = c \,|\, \mathscr{M} = m\right] = \Pr\left[Enc\left(\mathscr{K}, m\right) = c\right]$ is a number (say α_c) that does not depend on m.

So, $\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right]$ $(= \alpha_c)$ for all m and m', giving us **IND**.

Proof Part 2. Secrecy $\Longrightarrow$ Indistinguishability

WE KNOW (SEC): $\forall \mathscr{M}\ \forall m \in \operatorname{Supp}(\mathscr{M}),\ \forall c \in \operatorname{Supp}(\mathscr{C}),$

$$\Pr\left[\mathscr{M} = m \mid \mathscr{C} = c\right] = \Pr\left[\mathscr{M} = m\right]$$

WE WANT (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \operatorname{Supp}(\mathscr{M}),\ \forall c \in \operatorname{Supp}(\mathscr{C}),$

$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] = \Pr\left[Enc\left(\mathscr{K}, m'\right) = c\right]$$

Proof: As before, SEC is true if and only if $\Pr\left[\mathscr{C} = c \mid \mathscr{M} = m\right] = \Pr\left[\mathscr{C} = c\right]$ for all m and c.

Proof Part 2. Secrecy $\Longrightarrow$ Indistinguishability

WE KNOW (SEC): $\forall \mathscr{M}\ \forall m \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr[\mathscr{M} = m \,|\, \mathscr{C} = c] = \Pr[\mathscr{M} = m]$$

WE WANT (IND): $\forall \mathscr{M}\ \forall m,\ m' \in \text{Supp}(\mathscr{M}),\ \forall c \in \text{Supp}(\mathscr{C}),$

$$\Pr[Enc(\mathscr{K}, m) = c] = \Pr[Enc(\mathscr{K}, m') = c]$$

Proof: As before, SEC is true if and only if $\Pr[\mathscr{C} = c \,|\, \mathscr{M} = m] = \Pr[\mathscr{C} = c]$ for all m and c.

$$\begin{aligned}\Pr[Enc(\mathscr{K}, m) = c] &= \Pr[\mathscr{C} = c \,|\, \mathscr{M} = m] \\ &= \Pr[\mathscr{C} = c \,|\, \mathscr{M} = m'] \\ &= \Pr[Enc(\mathscr{K}, m') = c]\end{aligned}$$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c):$ Output $m = c \bigoplus k$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c):$ Output $m = c \bigoplus k$

$\oplus$: bitwise exclusive OR (or XOR)

$0 \oplus 0 = 1 \oplus 1 = 0$

$0 \oplus 1 = 1 \oplus 0 = 1$

$a \oplus b = a + b \pmod 2$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c):$ Output $m = c \bigoplus k$

Correctness: $c \bigoplus k = \left(m \bigoplus k\right) \bigoplus k = m$.

Perfect Secrecy is Achievable

The One-time Pad Construction:

Gen: Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c)$: Output $m = c \bigoplus k$

<u>Claim</u>: One-time Pad achieves Perfect Indistinguishability (and therefore perfect secrecy).

<u>Proof</u>: For any $m, c \in \{0,1\}^n$,

$$\Pr[\mathrm{Enc}(\mathrm{K}, m) = c] = \Pr[m \bigoplus \mathrm{K} = c]$$

$$= \Pr[\mathrm{K} = c \bigoplus m] = 1/2$$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c)$: Output $m = c \bigoplus k$

Claim: One-time Pad achieves Perfect Indistinguishability (and therefore perfect secrecy).

Proof: For any $m, c \in \{0,1\}^n$,

$$\Pr[\mathrm{Enc}(\mathrm{K}, m) = c] = \Pr\left[m \bigoplus \mathrm{K} = c\right]$$

$$= \Pr\left[\mathrm{K} = c \bigoplus m\right] = 1/2^n$$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c):$ Output $m = c \bigoplus k$

Claim: One-time Pad achieves Perfect Indistinguishability (and therefore perfect secrecy).

Proof: For any $m, c \in \{0,1\}^n$,

$$\Pr[\mathrm{Enc}(\mathrm{K}, m) = c] = \Pr[m \bigoplus \mathrm{K} = c]$$

$$= \Pr[\mathrm{K} = c \bigoplus m] = 1/2^n$$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k,m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k,c):$ Output $m = c \bigoplus k$

Claim: One-time Pad achieves Perfect Indistinguishability (and therefore perfect secrecy).

Proof: For any $m, c \in \{0,1\}^n$,

$$\Pr[\mathrm{Enc}(\mathrm{K}, m) = c] = \Pr\left[m \bigoplus \mathrm{K} = c\right]$$

$$= \Pr\left[\mathrm{K} = c \bigoplus m\right] = 1/2^n$$

Perfect Secrecy is Achievable

The One-time Pad Construction:

$Gen:$ Choose an n-bit string k at random, i.e. $k \leftarrow \{0,1\}^n$

$Enc(k, m)$, where M is an n-bit message: Output $c = m \bigoplus k$

$Dec(k, c)$: Output $m = c \bigoplus k$

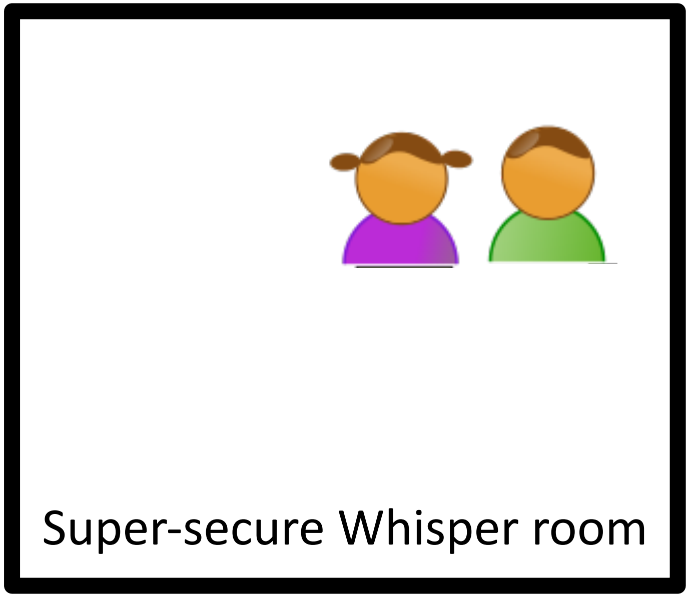
Claim: One-time Pad achieves Perfect Indistinguishability (and therefore perfect secrecy).

Proof: For any $m, m', c \in \{0,1\}^n$,

$$\text{So, } \Pr[\mathrm{Enc}(\mathrm{K}, m) = c] = \Pr[\mathrm{Enc}(\mathrm{K}, m') = c].$$

QED.

Reusing a One-time Pad?



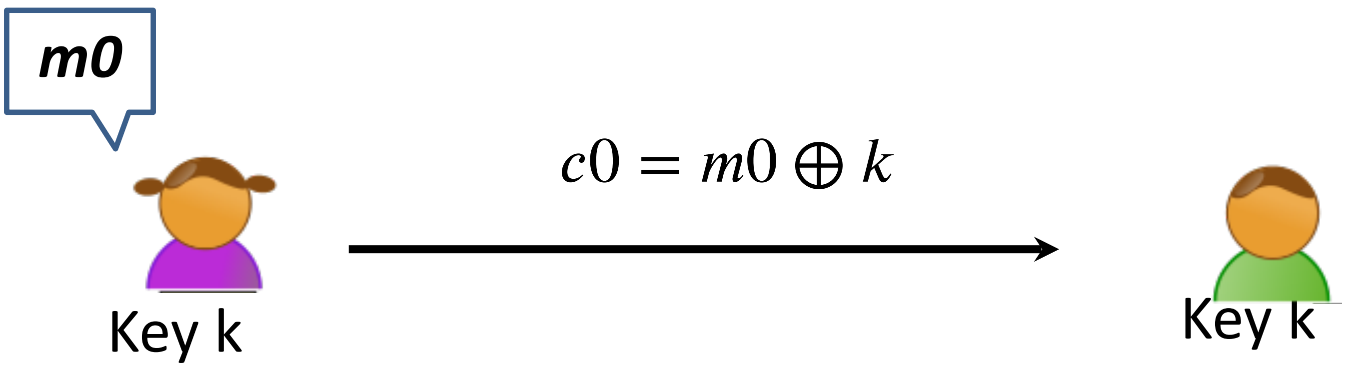
Reusing a One-time Pad?

Key k

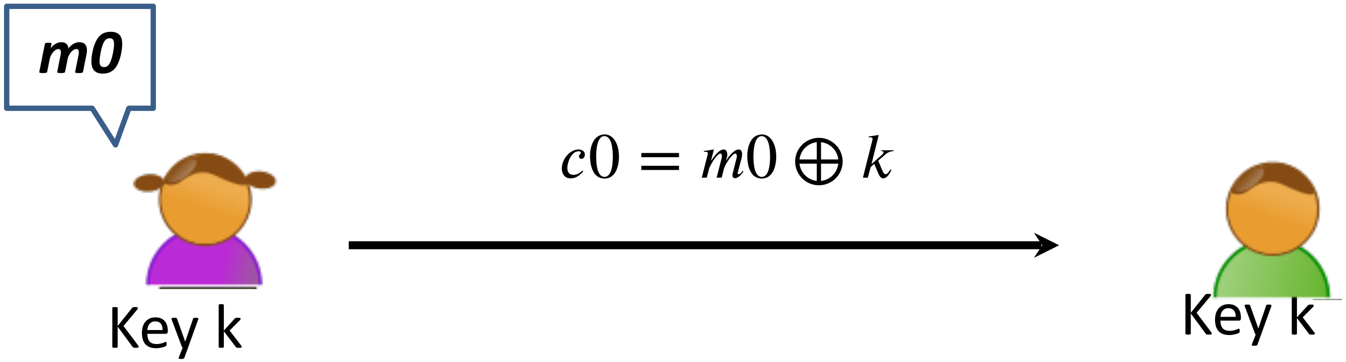
Key k

Reusing a One-time Pad?

Reusing a One-time Pad?

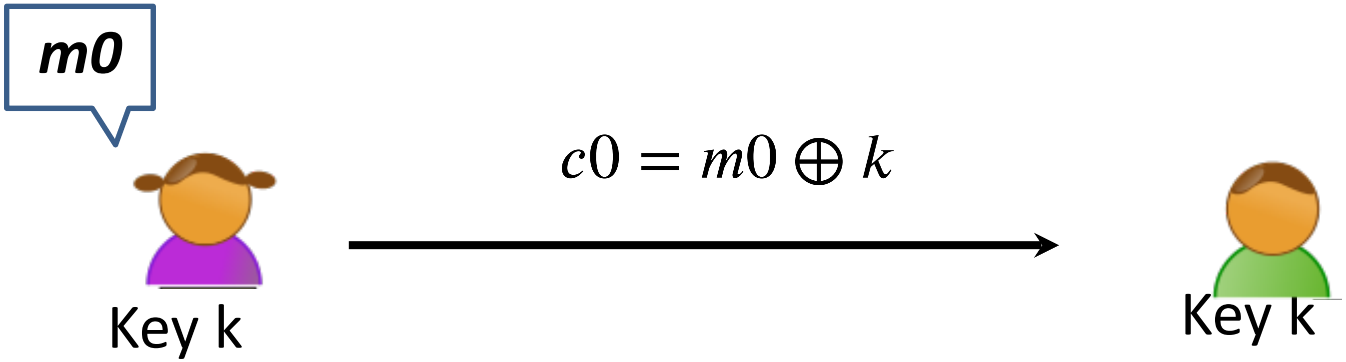


Reusing a One-time Pad?

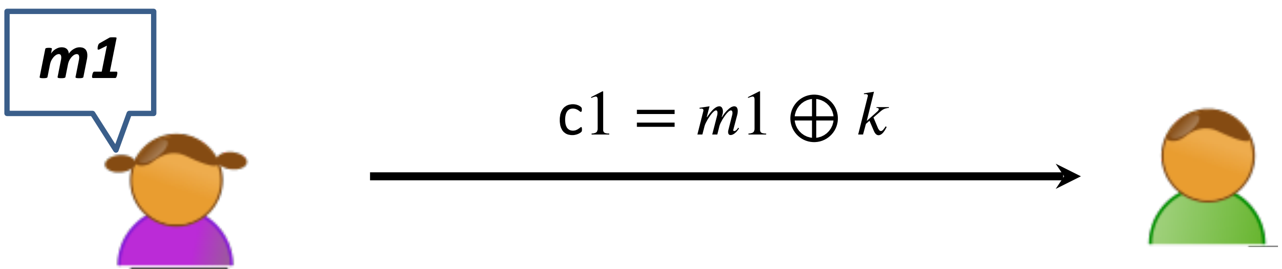


A week later:

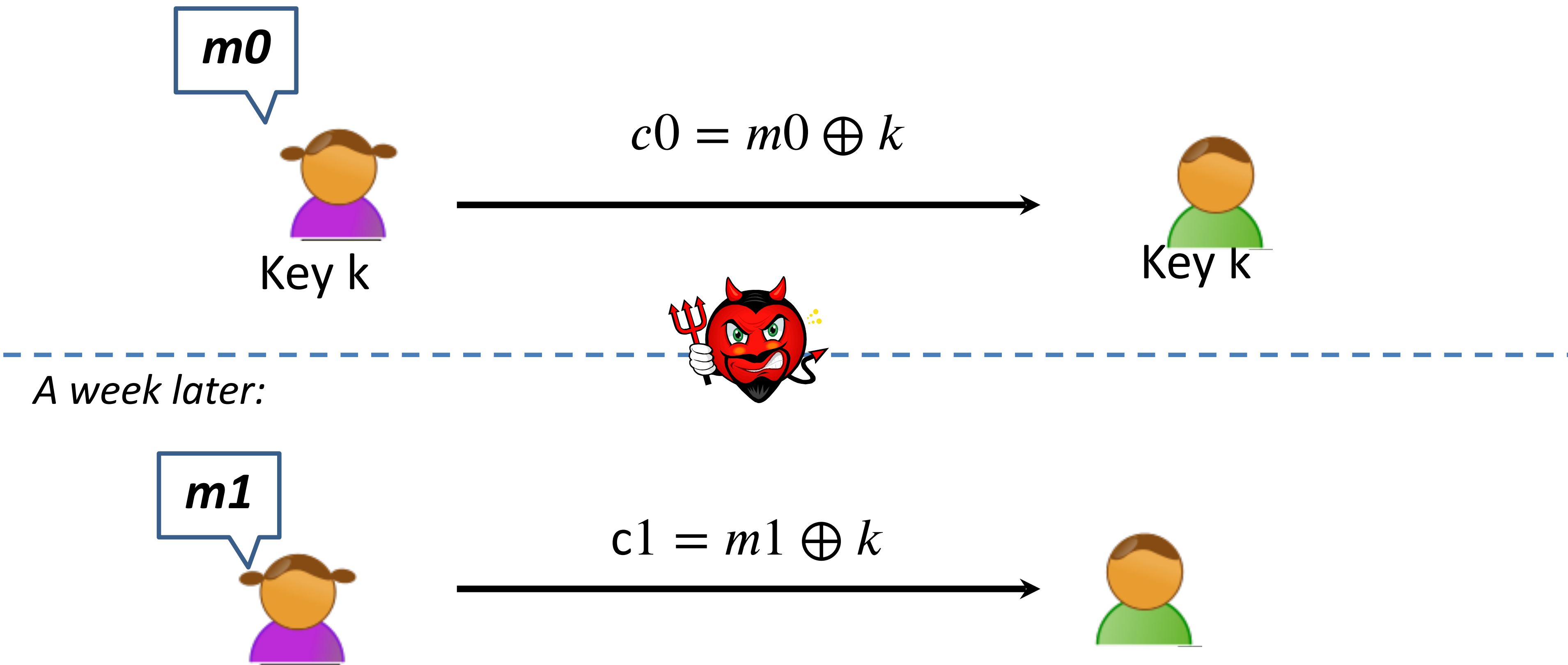
Reusing a One-time Pad?



A week later:



Reusing a One-time Pad?



Is this *still* perfectly secret?

Reusing a One-time Pad?

<u>Claim</u>: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: Perfect indistinguishability requires that for all pairs $(m0, m1), (m0', m1'), (c0, c1) \in \{0,1\}^{2n}$:

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: Perfect indistinguishability requires that for all pairs $(m0, m1), (m0', m1'), (c0, c1) \in \{0,1\}^{2n}$:

$$\Pr[\mathrm{Enc}\left(\mathrm{k}, m0\right) = c0 \; and \; \mathrm{Enc}(\mathrm{k}, m1\right) = c1]$$

$$= \Pr[\mathrm{Enc}\left(\mathrm{k}, m0'\right) = c0 \; and \; \mathrm{Enc}(k, m1'\right) = c1]$$

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: We want to pick $(m0,m1),(m0',m1'),(c0,c1) \in \{0,1\}^{2n}$ s.t.

$$\Pr[\mathrm{Enc}\big(\mathrm{k}, m0) = c0 \; and \; \mathrm{Enc}(k, m1\big) = c1]$$
$$\neq \Pr[\mathrm{Enc}\big(\mathrm{k}, m0') = c0 \; and \; \mathrm{Enc}(k, m1'\big) = c1]$$

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: We want to pick $(m0,m1),(m0',m1'),(c0,c1) \in \{0,1\}^{2n}$ s.t.

$$\Pr[\mathrm{Enc}\big(\mathrm{k}, m0) = c0 \; and \; \mathrm{Enc}(k, m1\big) = c1]$$
$$\neq \Pr[\mathrm{Enc}\big(\mathrm{k}, m0') = c0 \; and \; \mathrm{Enc}(k, m1'\big) = c1]$$

Pick $\boldsymbol{m0 = m1 = m}$, $\boldsymbol{m0' \neq m1'}$ **and** $\boldsymbol{c0 = c1 = c}$.

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: We want to pick $(m0, m1), (m0', m1'), (c0, c1) \in \{0,1\}^{2n}$ s.t.

$$\Pr[\mathrm{Enc}\left(\mathrm{k}, m0\right) = c0 \; and \; \mathrm{Enc}(k, m1\right) = c1]$$
$$\neq \Pr[\mathrm{Enc}\left(\mathrm{k}, m0'\right) = c0 \; and \; \mathrm{Enc}(k, m1'\right) = c1]$$

Pick $\boldsymbol{m0 = m1 = m}$, $\boldsymbol{m0' \neq m1'}$ and $\boldsymbol{c0 = c1 = c}$.

$$\Pr[\mathrm{Enc}\left(\mathrm{k}, m0\right) = c0 \; and \; \mathrm{Enc}(k, m1\right) = c1]$$

Reusing a One-time Pad?

<u>Claim</u>: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

<u>Proof</u>: We want to pick $\underline{(m0, m1), (m0', m1'), (c0, c1) \in \{0,1\}^{2n}}$ s.t.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c0 \; and \; \mathrm{Enc}(k, m1) = c1]$$
$$\neq \Pr[\mathrm{Enc}(\mathrm{k}, m0') = c0 \; and \; \mathrm{Enc}(k, m1') = c1]$$

Pick $\boldsymbol{m0 = m1 = m}$, $\boldsymbol{m0' \neq m1'}$ **and** $\boldsymbol{c0 = c1 = c}$.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c0 \; and \; \mathrm{Enc}(k, m1) = c1]$$
$$= \Pr[\mathrm{Enc}(k, m) = c] = 1/2^n$$

Reusing a One-time Pad?

<u>Claim</u>: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

<u>Proof</u>: We want to pick $\underline{(m0,m1),(m0',m1'),(c0,c1) \in \{0,1\}^{2n}}$ s.t.

$$\Pr[\mathrm{Enc}\big(\mathrm{k}, m0) = c0 \textit{ and } \mathrm{Enc}(k, m1\big) = c1]$$
$$\neq \Pr[\mathrm{Enc}\big(\mathrm{k}, m0') = c0 \textit{ and } \mathrm{Enc}(k, m1'\big) = c1]$$

$\textbf{Pick } \boldsymbol{m0 = m1 = m, m0' \neq m1'} \textbf{ and } \boldsymbol{c0 = c1 = c}.$

$$\Pr[\mathrm{Enc}\big(\mathrm{k}, m0) = c = \mathrm{Enc}(k, m1\big)] = 1/2^n$$

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: We want to pick $(m0,m1),(m0',m1'),(c0,c1) \in \{0,1\}^{2n}$ s.t.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c0 \ and \ \mathrm{Enc}(k, m1) = c1]$$
$$\neq \Pr[\mathrm{Enc}(\mathrm{k}, m0') = c0 \ and \ \mathrm{Enc}(k, m1') = c1]$$

Pick $\boldsymbol{m0 = m1 = m}$, $\boldsymbol{m0' \neq m1'}$ **and** $\boldsymbol{c0 = c1 = c}$.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c = \mathrm{Enc}(k, m1)] = 1/2^n$$

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0') = \mathrm{c} = \mathrm{Enc}(k, m1')] = 0$$

Reusing a One-time Pad?

Claim: Two-time Pad does ***not*** achieve Perfect Indistinguishability (and therefore ***not*** perfect secrecy).

Proof: We want to pick $(m0, m1), (m0', m1'), (c0, c1) \in \{0,1\}^{2n}$ s.t.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c0 \textit{ and } \mathrm{Enc}(k, m1) = c1]$$
$$\neq \Pr[\mathrm{Enc}(\mathrm{k}, m0') = c0 \textit{ and } \mathrm{Enc}(k, m1') = c1]$$

Pick $m0 = m1 = m$, $m0' \neq m1'$ **and** $c0 = c1 = c$.

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0) = c = \mathrm{Enc}(k, m1)] = 1/2^n$$

$$\Pr[\mathrm{Enc}(\mathrm{k}, m0') = \mathrm{c} = \mathrm{Enc}(k, m1')] = 0$$

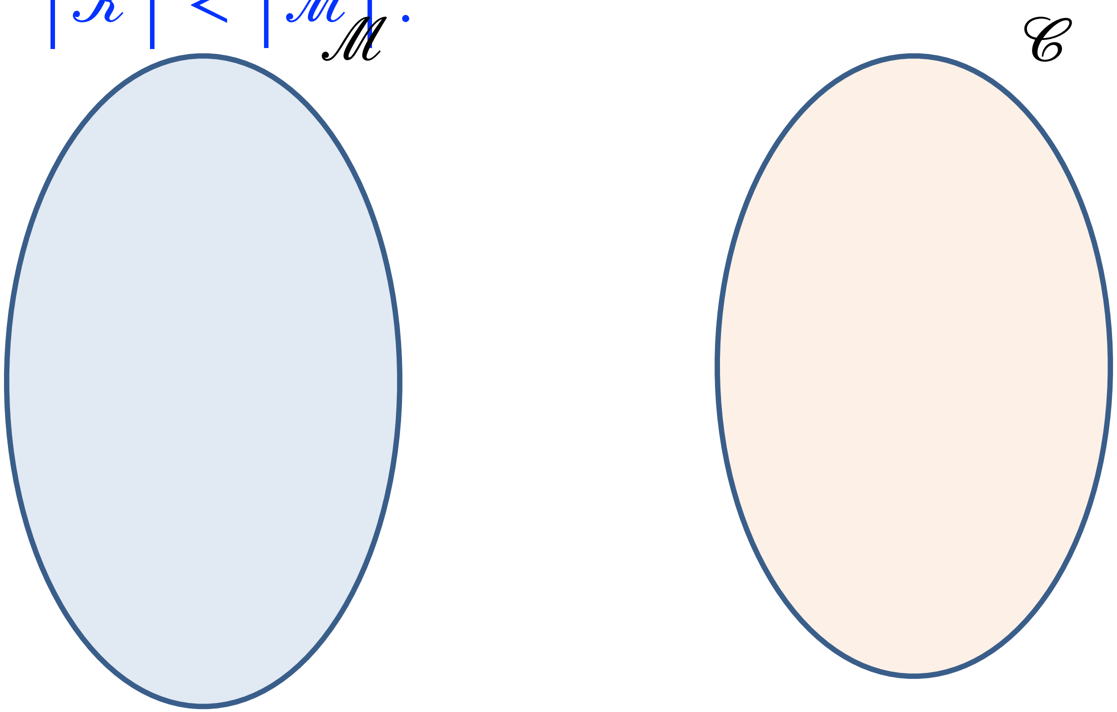
$\blacksquare$

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.

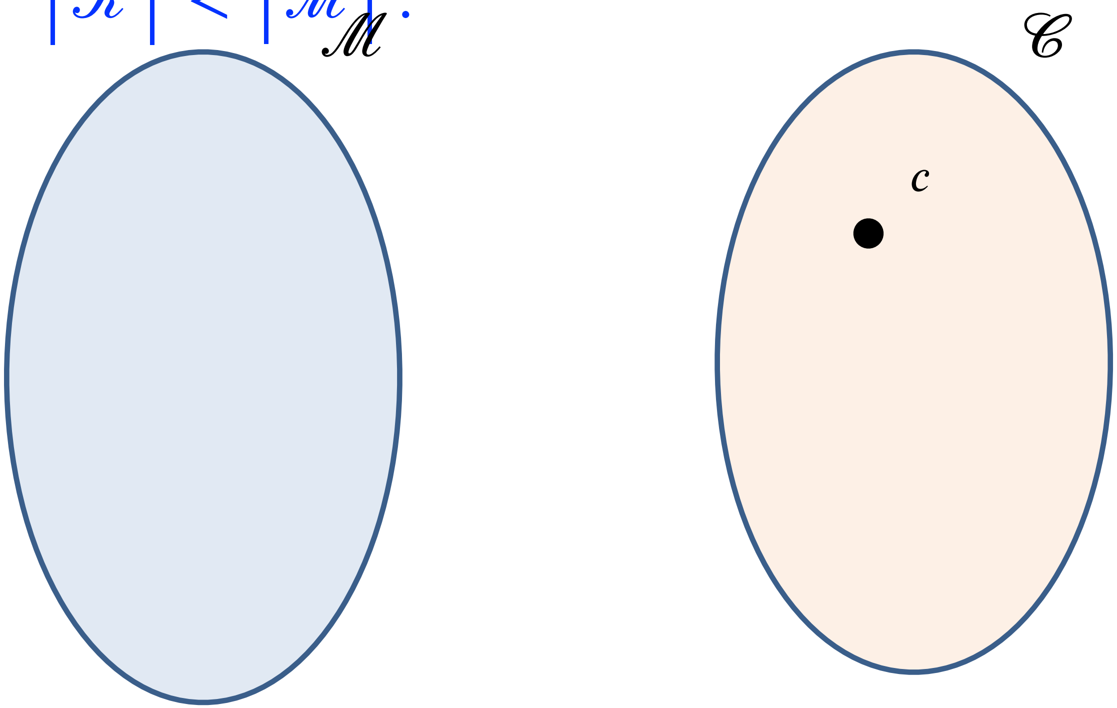


Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.



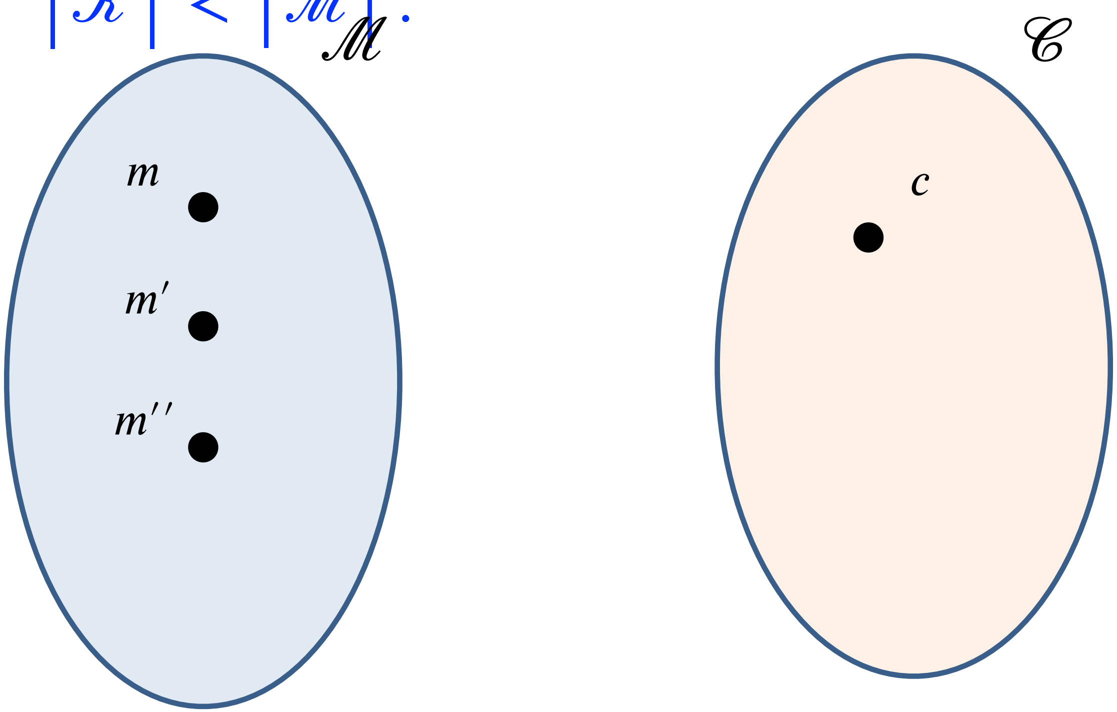
Pick any $c \in \mathscr{C}$

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.



Pick any $c \in \mathscr{C}$

Look at the set of possible msgs

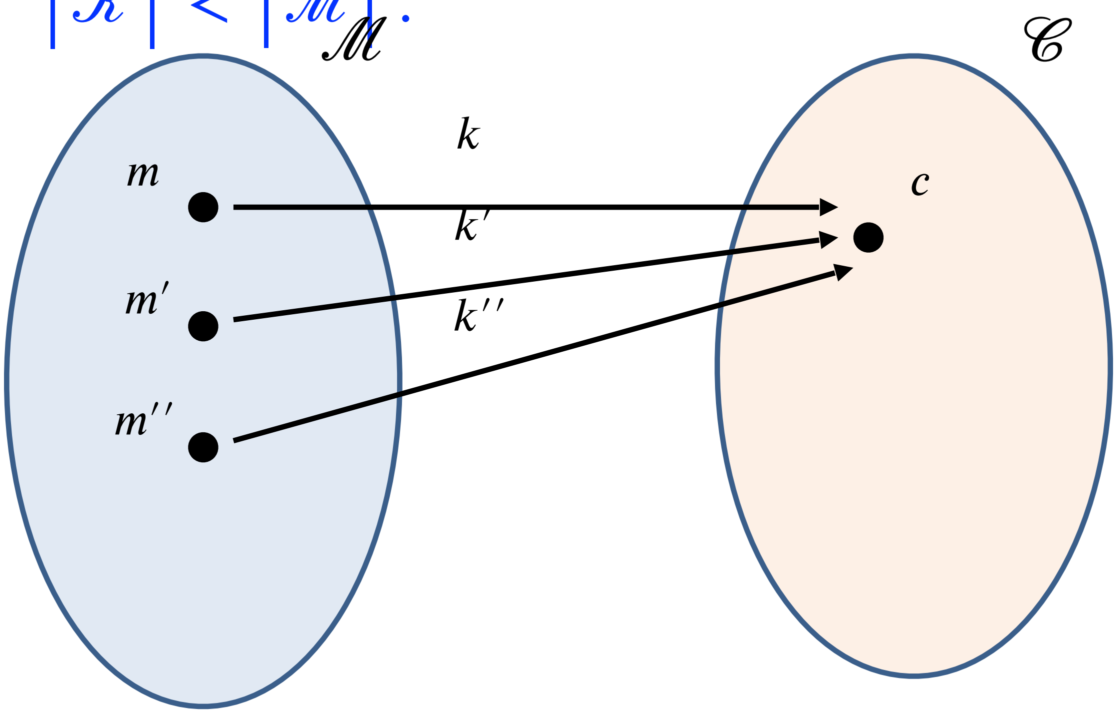
($m = Dec(k, c)$ etc.)

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.



Pick any $c \in \mathscr{C}$

Look at the set of possible msgs

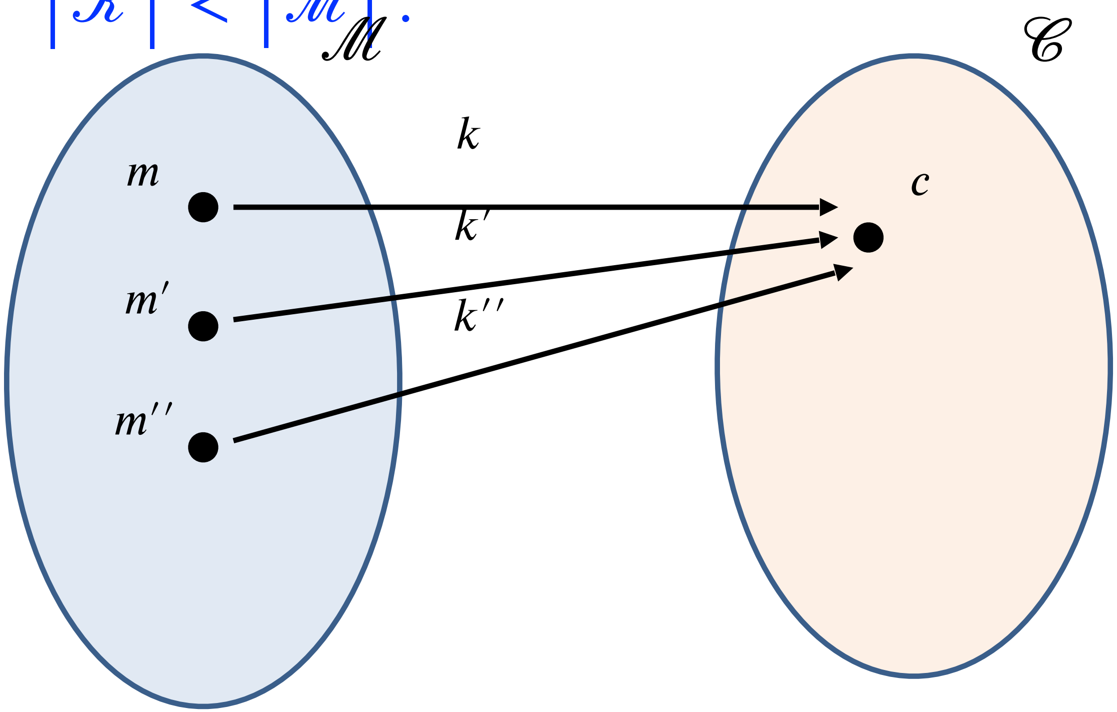
($m = Dec(k, c)$ etc.)

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.



Pick any $c \in \mathscr{C}$

Look at the set of possible msgs ($m = Dec(k, c)$ etc.)

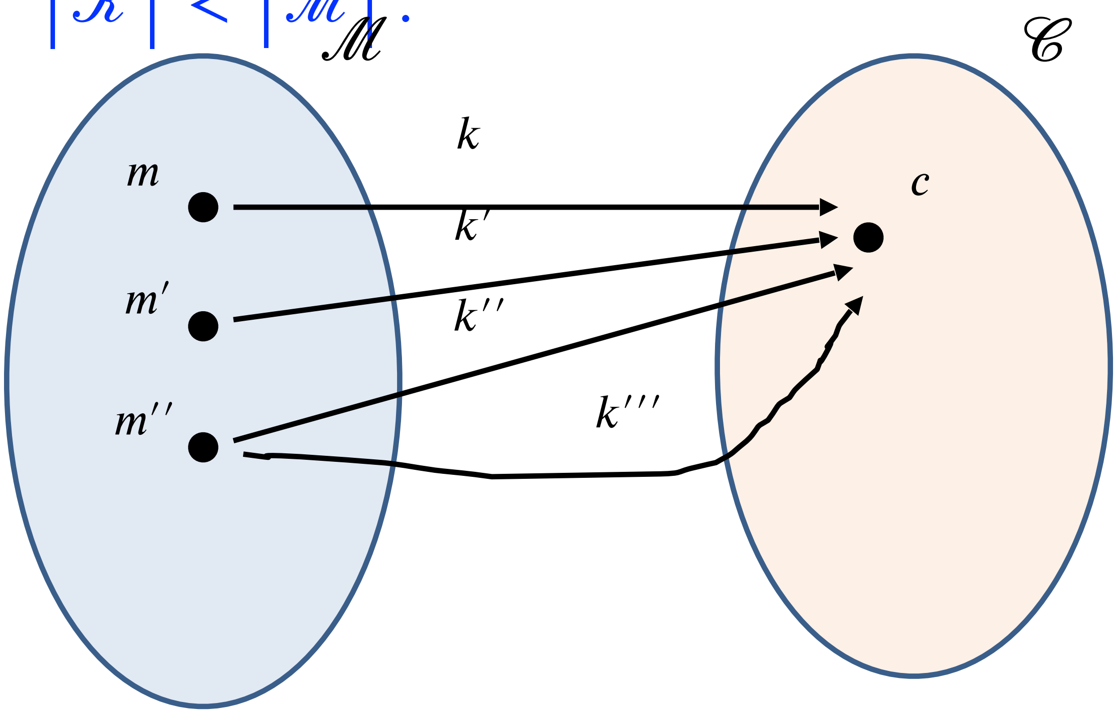
Distinct keys!

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathscr{K}| < |\mathscr{M}|$.



Pick any $c \in \mathscr{C}$

Look at the set of possible msgs

($m = Dec(k, c)$ etc.)

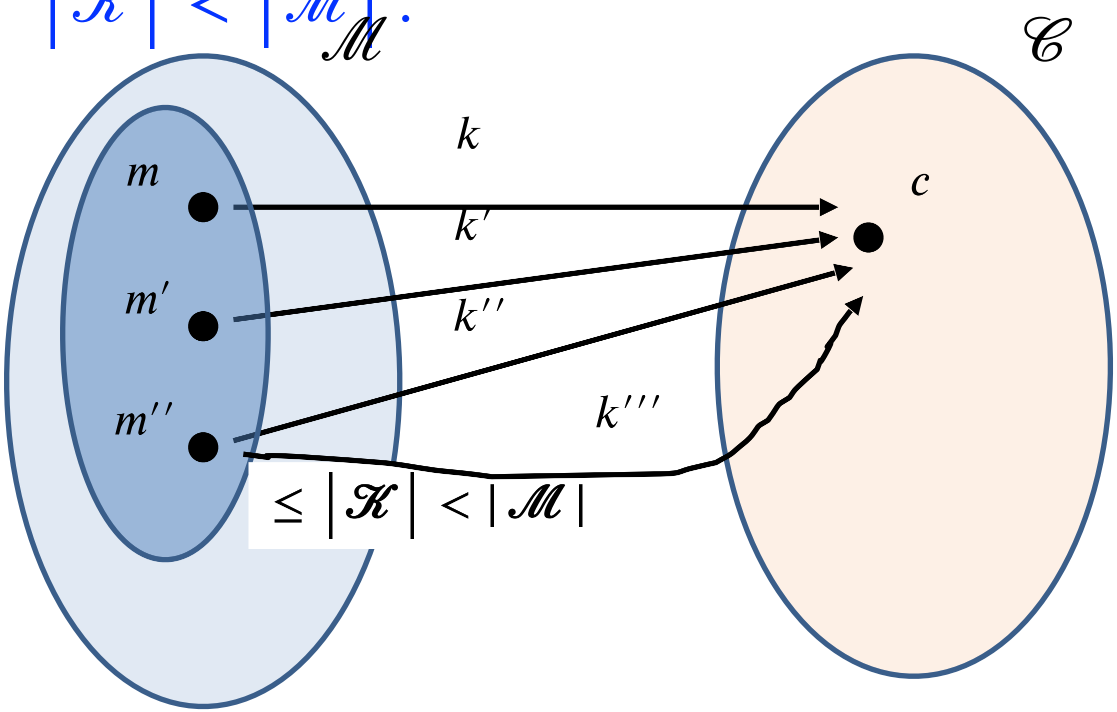
Distinct keys!

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathcal{K}| \geq |\mathcal{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathcal{K}| < |\mathcal{M}|$.



Pick any $c \in \mathcal{C}$

Look at the set of possible msgs ($m = Dec(k, c)$ etc.)

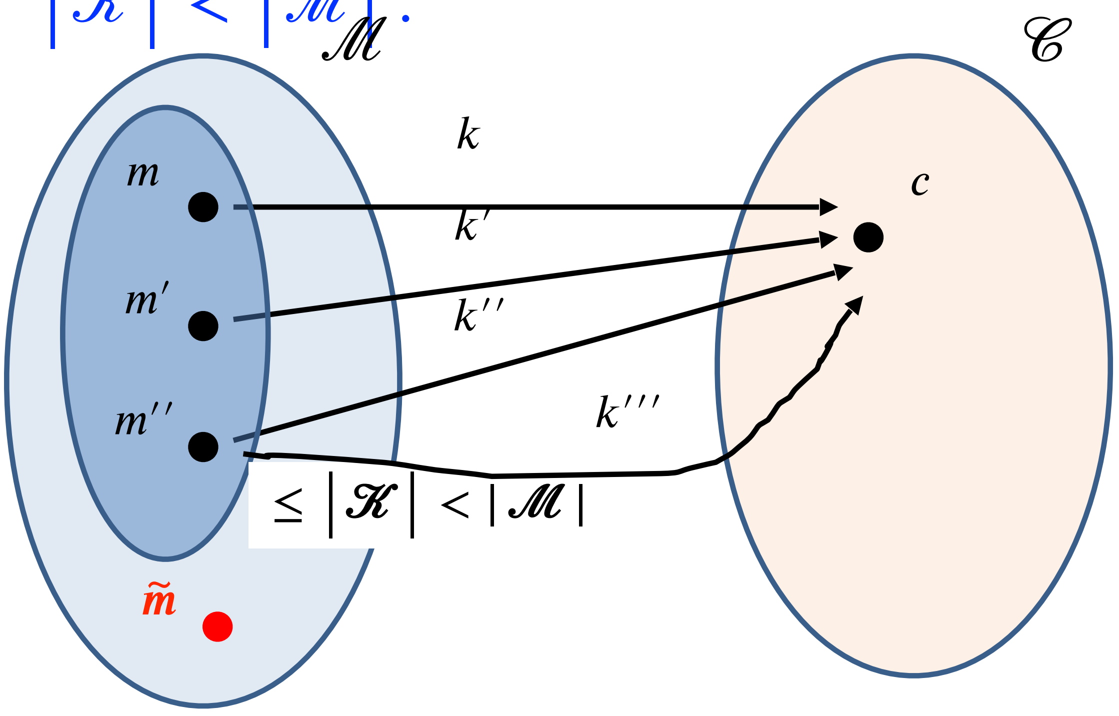
Distinct keys!

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathcal{K}| \geq |\mathcal{M}|$$

PROOF (by picture): Assume for contradiction that $|\mathcal{K}| < |\mathcal{M}|$.



Pick any $c \in \mathcal{C}$

Look at the set of possible msgs ($m = Dec(k, c)$ etc.)

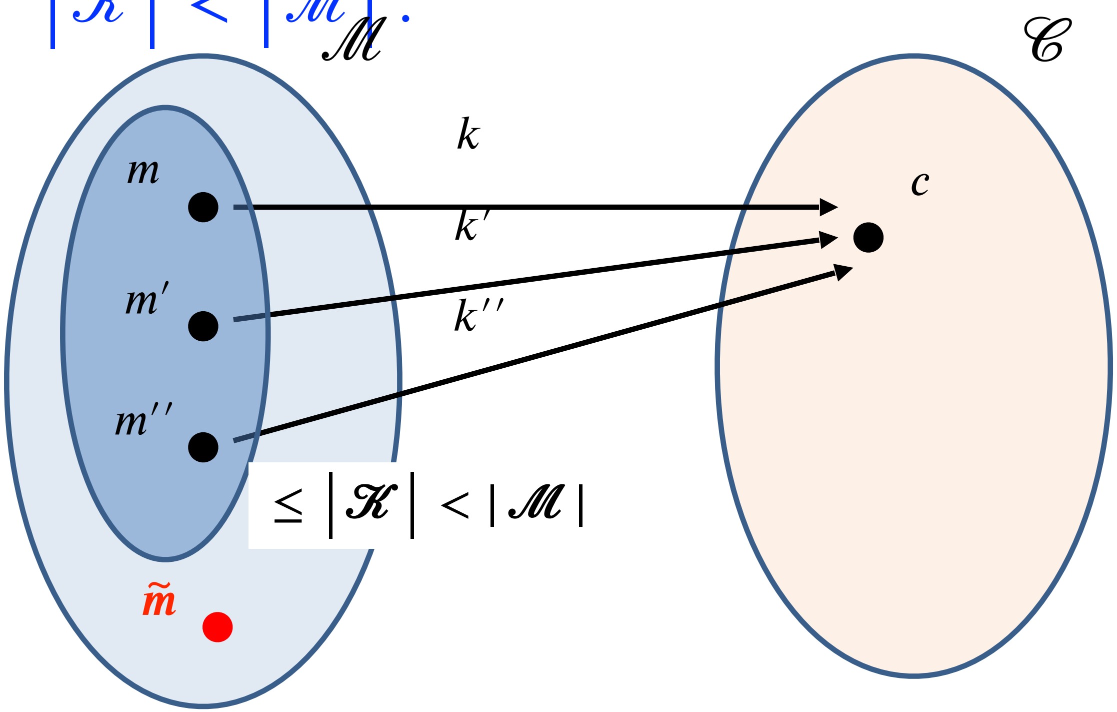
Distinct keys!

Perfect Secrecy has its Price

THEOREM: For any perfectly secure encryption scheme,

$$|\mathscr{K}| \geq |\mathscr{M}|$$

PROOF (by picture): Assume for contradiction that $\left|\mathscr{K}\right| < \left|\mathscr{M}\right|$.



$$\Pr\left[Enc\left(\mathscr{K}, m\right) = c\right] > 0$$

$$\Pr\left[Enc\left(\mathscr{K}, \tilde{m}\right) = c\right] = 0$$

$\boldsymbol{QED}$.

So, what are we to do?

So, what are we to do?

RELAX the definition:

EVE is an arbitrary *computationally bounded* algorithm.

So, what are we to do?

RELAX the definition:

EVE is an arbitrary *computationally bounded* algorithm.

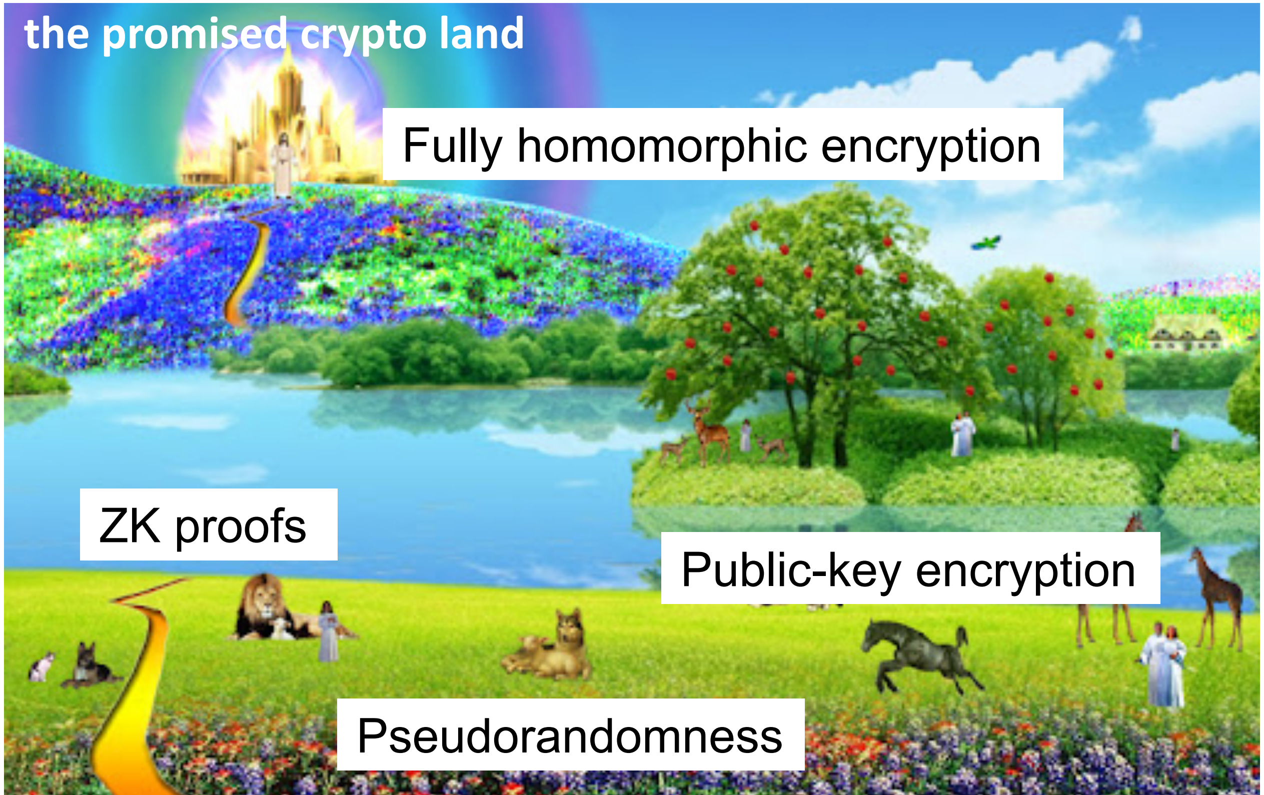
+ number theory/geometry/combinatorics

So, what are we to do?

RELAX the definition:

EVE is an arbitrary *computationally bounded* algorithm.

+ number theory/geometry/combinatorics



To Summarize…

- **Secure Communication**: a quintessential problem in cryptography.

- We saw two equivalent definitions of security:

  Shannon's perfect indistinguishability and perfect secrecy

- **One-time pad achieves perfect secrecy.**

- **A Serious Limitation:** Any perfectly secure encryption scheme needs keys that are at least as long as the messages.

- **Next Lecture: Overcoming the limitation** with Computationally Bounded Adversaries.