CS716/IS703

Data Security/Information System Security

Week 1:

- Course Overview & What is Cryptography
- Historical Ciphers (& How to Break Them)
- Perfect Secrecy

Fall 2025

Assoc. Prof. Hisham Dahshan

hishamdahshan@aast.edu

Course Information Policies

- Grading:
  - Weeks 3:7th Midterm: **30%** (20% Midterm + 10% Assignments & Quiz)
  - Weeks 8:12th Midterm: **10%**
  - Presentation & paper review: **10%**
  - Project: **10%** (Discussion during last week of semester)
  - Final Exam: **40%**
- Attendance:
  - Attendance for lectures and labs is **mandatory**

Course Contents:

Week #	Description
1	Course Intro, One-Time Pad and Perfect Secrecy
2	Computational Security, Pseudorandomness and Stream Ciphers
3	CPA Security + PRFs, CCA Security
4	Stream Ciphers, Block Ciphers, DES, 3DES
5	Advanced Encryption Standard (AES)
6	Cryptographic Hash Function, HMACs
7	7th Mid Term
8	Public Key Cryptography
9	Key Management
10	Digital Signatures
11	Threshold Cryptography
12	12th Mid Term
13	Zero-Knowledge Proofs
14	Email Security, Web Security
15	Database Security, Firewalls
16	Final Exam

What Is This Course About?

What it IS about:

- theoretical cryptography;
- “replacing trust with mathematics”;
- exploring limits of what is possible *in principle;*
- fundamental tasks: encryption, authentication

What it is NOT about:

- practical IT security;
- hacking, spoofing, fishing, DOS attacks, etc.;
- real-world implementations;
- bleeding edge theory: obfuscation, quantum FHE

What we WILL do:

- define concepts rigorously, prove theorems
- analyze cryptosystems and attacks in terms of “possible in principle” vs “impossible, even in principle”

What we will NOT do:

- implement real cryptosystems or attacks
- analyze cryptosystems and attacks in terms of concrete costs (e.g., 20 minutes vs 2 hours on a four-core Xeon with 32GB RAM…?)

What Is This Course About?

What background should you refresh?

- Discrete probability: random variables and events, conditional probability, expectation, etc.;
- Theory of computation: basic algorithms and programming concepts, asymptotic analysis (O-notation), etc;
- Mathematical rigor: formal definitions, notation, theorems, proofs;

Basically, the stuff you (hopefully) did in discrete math.

I. The (Sketchy) History Of Crypto

Reading: xv – p.24.

What Is Cryptography?

What is cryptography?

How will we study it in this course?

Why will we do it that way?

To answer all this: need to first look at how crypto has been done for most of history.

This is not a "boring history lesson" you can ignore!

- people were very clever before computers too!
- develop intuition about what "good crypto" and "bad crypto" look like;
- learn basic techniques for breaking cryptosystems;
- understand *why* we now do crypto the way we do it;
- some historical schemes still crop up in modern crypto!

…and besides, history is awesome!

Historical Ciphers : Caesar Cipher

Caesar cipher

Goal: “send secret messages”

- shift each letter in the message, remove spaces
- Caesar himself used this; his key was 3, like this:

encrypt

A B C D E F G H I J K L M …

decrypt

message (plaintext)

ATTACK AT DAWN

encrypt *decrypt*

DWWDFNDWGDZR

ciphertext

- apparently, Caesar used this successfully for many years
- in 2011, used in a plot to attack airliners (no, really.)

Is it secure?

Historical Ciphers : Caesar Cipher

No! Brute force keysearch:

Suppose you see the message "dwwdfndwrqfh" (but you don't know Caesar's key.)

Try all possible decryption keys:

0	dwwdfndwrqfh
-1	cvvcemcvqpeg
-2	buubdlbupodf
-3	attackatonce
-4	zsszbjzsnmbd
-5	yrryaiyrmlac
⋮	⋮

Only 26 possibilities, so easy! (The 2011 plot failed and the plotters were caught.)

Must have: big keyspace.

Historical Ciphers : Substitution Cipher

Substitution cipher

- each letter of the alphabet is mapped to another, randomly selected letter
- for example:

key

$$A \mapsto X$$
$$B \mapsto F$$
$$C \mapsto D$$
$$D \mapsto L$$
$$E \mapsto P$$
$$\ldots$$
$$\ldots$$

BEE BED DEAD

encrypt *decrypt*

FPPFPLLPXL

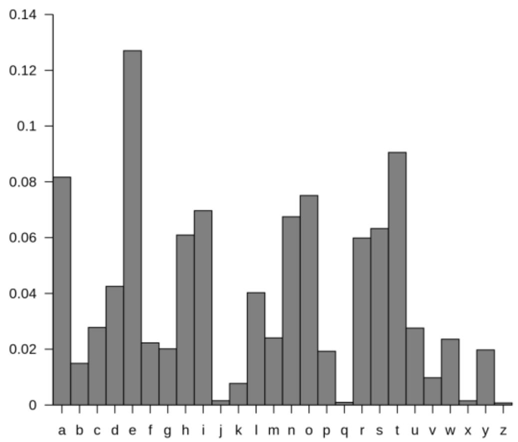
- used in 1586 plot by Mary, Queen of Scots to assassinate Queen Elizabeth and install Mary as queen;
- Mary used the cipher to instruct her collaborators to kill the queen!

Key space: $26! \approx 10^{26}$

Is it secure?

Historical Ciphers : Substitution Cipher

Unfortunately for Mary, an Arab philosopher named Al-Kindi broke this cipher *over 700 years prior*.



Frequency analysis

- plot average frequency of letters in spoken English;
- do the same for the encrypted message;
- permute the letters to make the plots match up;
- the resulting permutation is (probably close to) the key!

- Mary's messages were intercepted and broken with frequency analysis;
- using the key, the messages were even changed to get her to reveal her conspirators *(authentication?);*
- based on this, Mary was found guilty and beheaded.

Crypto mattered a lot even in 1586!

Historical Ciphers : Vigenère Cipher

If Mary had a better cryptographer, she would have used Vigenère cipher (discovered a few years prior.)

	YOUCANEXPECTNOHELPFROMTHISSIDEOFTHERIVER
+	VICTORVICTORVICTORVICTORVICTORVICTORVICT
=	UXXWPFAGSYRLJXKYAHBARGIZEBVCSWKOWBTJEEHL

+ means add letters as numbers (mod 28)

Used by the Confederacy in the U.S. Civil War.

Is it secure?

Historical Ciphers : Vigenère Cipher

```
    YOUCANEXPECTNOHELPFROMTHISSIDEOFTHERIVER
  + VICTORVICTORVICTORVICTORVICTORVICTORVICT
  = UXXWPFAGSYRLJXKYAHBARGIZEBVCSWKOWBTJEEHL
```

Guess the length of the passphrase. Then split up ciphertext:

```
UXXWPF
AGSYRL
JXKYAH
BARGIZ
EBVCSW
KOWBTJ
EEHL
```

- each column is a Caesar cipher; 26 choices there, but $26^6 \approx 309$ million total! No good…
- instead, frequency analysis with a twist: plot of first column = English alphabet translated by V!

It took over 300 years for someone to figure this out and break Vigenère. (So Mary might have gotten away with it!)

What Went Wrong?

Lessons learned

- key space needs to be large (prevent brute force key search);
- scheme needs to resist frequency analysis, sometimes in non-obvious ways;
- what else? Is that enough?
- … as it turns out, it's not; throughout history, each attempt to "patch" was eventually circumvented.
- (fun read: Enigma in WW2.)

The first "unbreakable" cipher was not discovered until 1882!

- *why did it take so long?*
- people have been clever for a long time; that didn't start in 1882;
- modern crypto *seems to be* a lot more "stable" than the stuff we discussed above
- what changed?
- (also: if there's an unbreakable cipher, what is left to do? As we will see, a lot!)

What Do We Do Differently Now?

The modern (theoretical) approach (~1970s on)

"the algorithm" *"the key"*

- emphasis on mathematical rigor
- formal definitions : what is known to everyone, and what needs to stay secret?
- formal definitions : what exactly is the cryptosystem trying to achieve?
- formal definitions : when is a cryptosystem considered "secure"?
- security proofs: mathematical theorems establishing security (with important caveats!)

<u>Kerckhoffs's principle</u>
A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

<u>*... and lots and lots of clever cryptographic (design) work and cryptanalytic (attack) work!*</u>

These will be the ideas that we will explore in this course.

II. (SIMPLE) ENCRYPTION

Reading: Ch.2 (p.25-40)

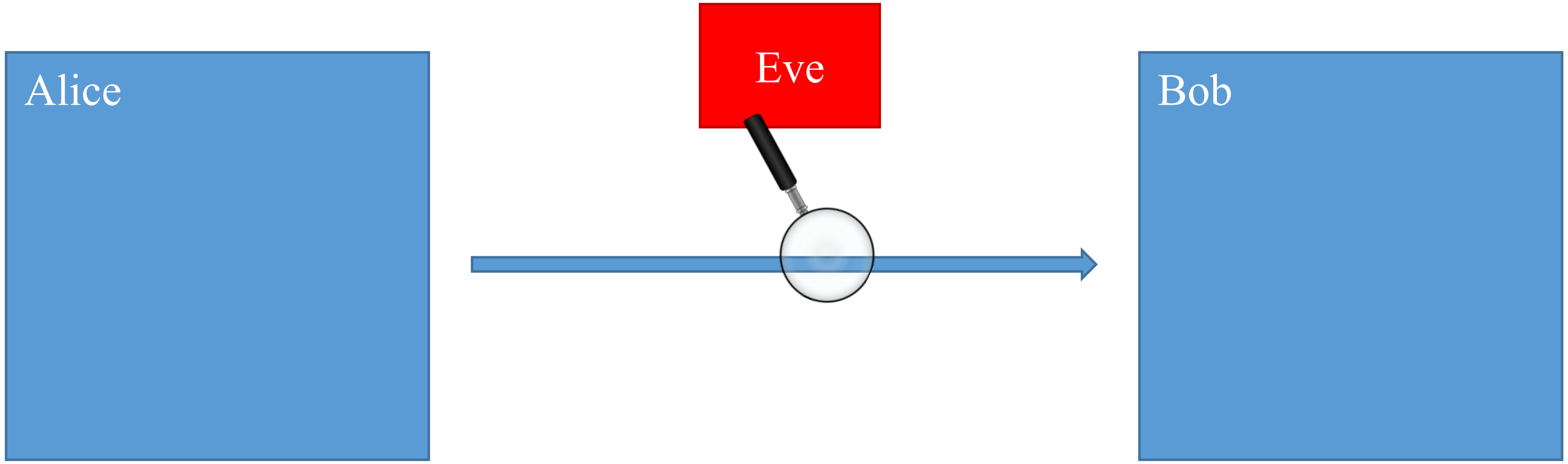
ENCRYPTION : THE SETTING

Task: Alice wants to send a single message to Bob, *but Eve is watching the channel.*

Assumptions:

- Alice and Bob can share a secret in advance;
- they have their own private spaces;
- Alice can send only one transmission, on a single channel;
- Eve (eavesdropper) can observe *everything* that is transmitted on that channel.
- *Eve cannot do anything else.*

Wait, why not just use this "assumption" to send the message?



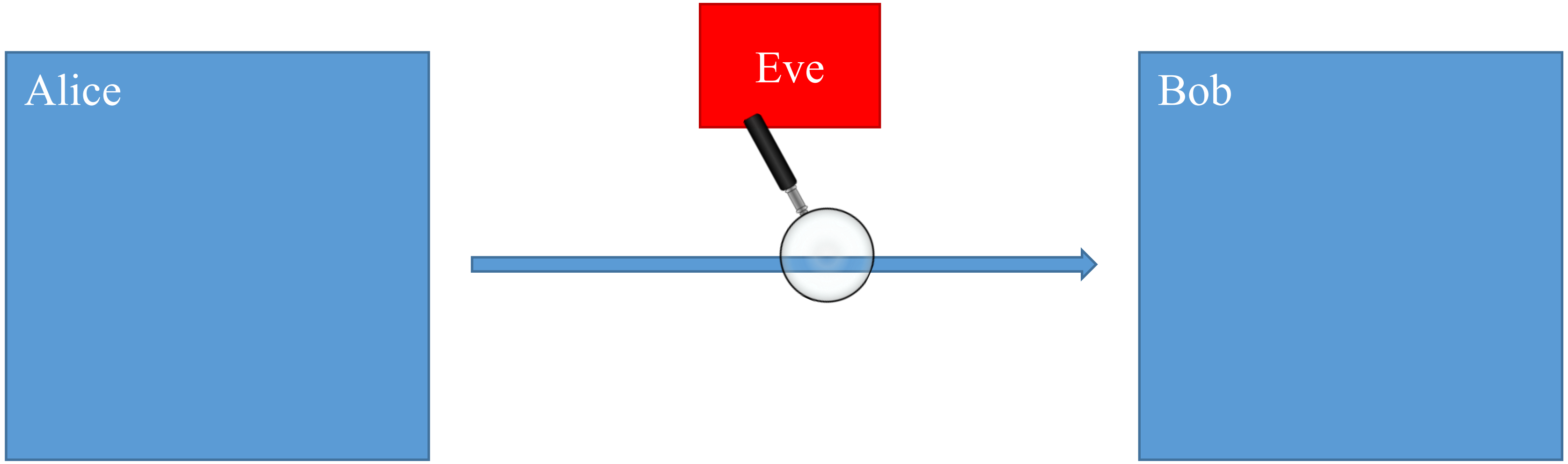
Encryption Schemes

Message-independent distribution.

Generic approach to this task:

- generate key via some algorithm: $k \leftarrow \mathbf{KeyGen}$
- encrypt via some algorithm: $c \leftarrow \mathbf{Enc}_k(m)$
- decrypt via some algorithm: $m \leftarrow \mathbf{Dec}_k(c)$

The triple $(\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ is called an *encryption scheme.*

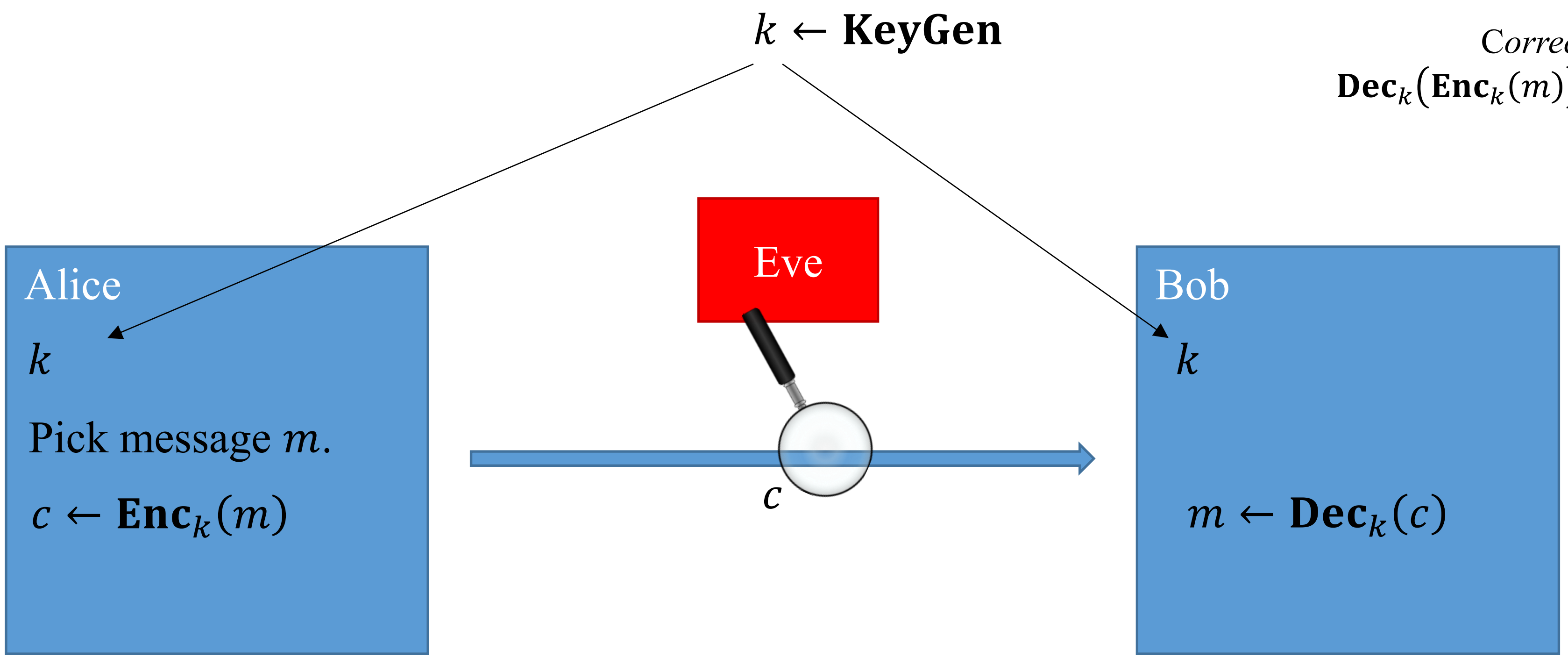


Encryption Schemes

Generic approach to this task:

- generate key via some algorithm: $k \leftarrow \mathbf{KeyGen}$
- encrypt via some algorithm: $c \leftarrow \mathbf{Enc}_k(m)$
- decrypt via some algorithm: $m \leftarrow \mathbf{Dec}_k(c)$

The triple $(\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ is called an *encryption scheme*.



Correctness:
$\mathbf{Dec}_k(\mathbf{Enc}_k(m)) = m$ for all m.

Encryption Schemes

Examples

Let's look at our initial Caesar's cipher example.

$3 \leftarrow$ **KeyGen**

Alice

3

Pick message **ATTACK AT DAWN**.

DWWDFNDWGDZR $\leftarrow$ **Enc**$_3$(**ATTACK AT DAWN**)

Eve

DWWDFNDWGDZR

Eve tries all 26 possible shifts, recovers key and message.

Bob

3

ATTACK AT DAWN $\leftarrow$ **Dec**$_3$(**DWWDFNDWGDZR**)

SCHEME BROKEN

Encryption Schemes: One-time Pad

Examples: one-time pad (Vernam cipher, ~1882)

- *Key generation* : sample uniformly random $k \in \{0,1\}^n$
- *Encryption* : $\mathbf{Enc}_k(m) = m \oplus k$
- *Decryption* : $\mathbf{Dec}_k(c) = c \oplus k$;

(note 1: messages are interpreted as bitstrings.)

(note 2: key length = message length = ciphertext length = n.)

Bitwise XOR (+ mod 2):

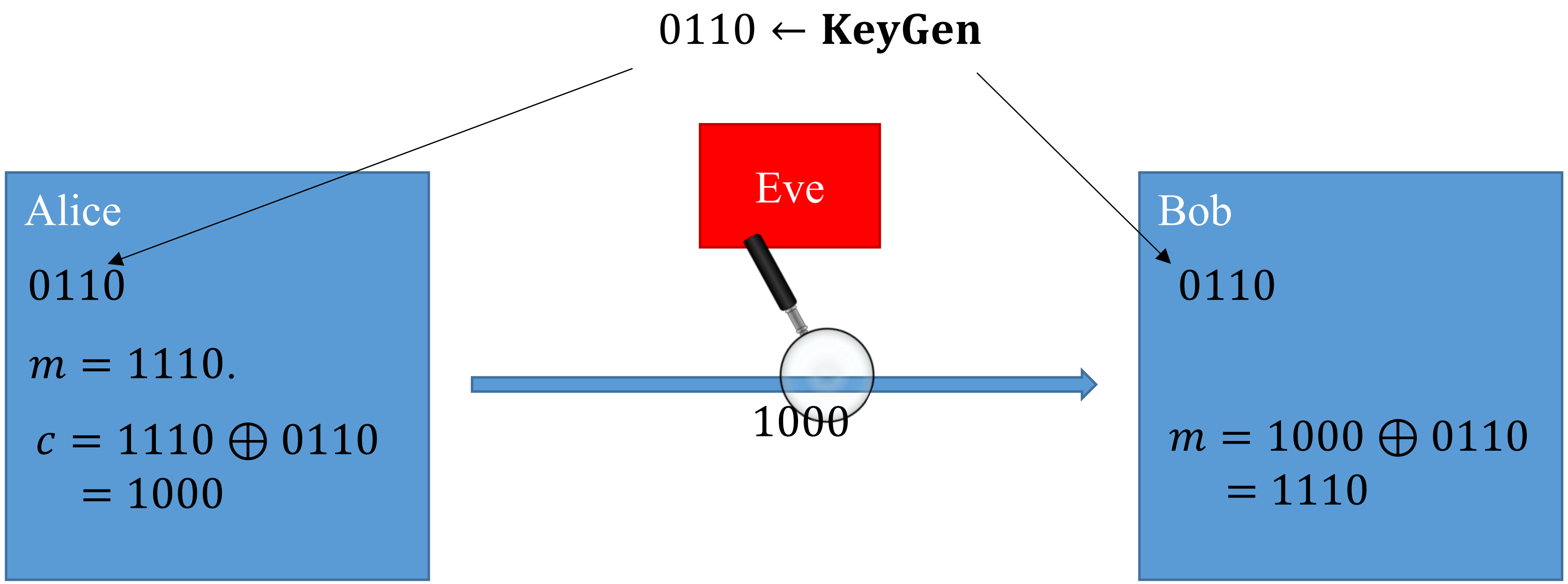
$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 1 = 0$

Check *correctness*:

$\mathbf{Dec}_k(\mathbf{Enc}_k(m)) = (m \oplus k) \oplus k = m$



Encryption

Is the one-time pad (OTP) secure?

What does it mean to be secure?

- impossible to recover the key?

Consider this scheme:

- **KeyGen** outputs a random string $k \in \{0,1\}^n$.
- $\mathbf{Enc}_k(m) = m.$

totally insecure!

- impossible to recover message?

Consider a scheme like this:

$$\mathbf{Enc}_k(m) = \underbrace{m_1 m_2 m_3 m_4}_{\text{first 4 bits leak}}\underbrace{********}_{\text{rest are secret (somehow)}}.$$

Or something more insidious…

… like leaking the parity of m?

More generally: what do we mean by "impossible to recover"?

A Little Probability

Random variables

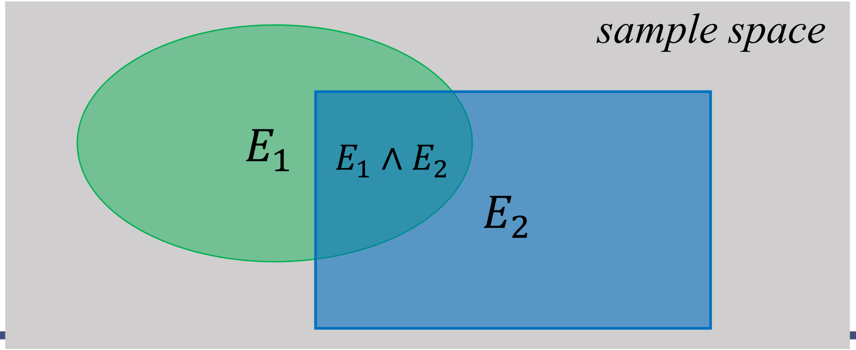
- outcome of some random experiment; denoted with capital letters: $X, Y, M, C, \dots$;
- comes with a probability distribution; denoted with calligraphic letters: $\mathcal{X}, \mathcal{Y}, \mathcal{M}, \mathcal{C}, \dots$;
- possible values (or samples) denoted with lowercase letters: $x, y, m, c, \dots$;
- **event**: a subset of the sample space of some random experiment.

Examples

Let X be uniformly random on $\{0,1\}^n$. Then $\Pr[\underbrace{\overset{RV}{\overset{\downarrow}{X}} = \overset{value}{\overset{\downarrow}{x}}}_{event}] = 2^{-n}$ for all $x \in \{0,1\}^n$.

Let X be uniformly random on $S = \{0,1,2,3,4\}$. Then $\mathbf{E}[X] = \sum_{s \in S} \Pr[X = s] \cdot s = \frac{1}{5}(0 + 1 + 2 + 3 + 4) = 2$.

Let E_1, E_2 be events. Then $\Pr[E_1 | E_2] \coloneqq \frac{\Pr[E_1 \wedge E_2]}{\Pr[E_2]}$.



Encryption: Secrecy: Candidate I

Secrecy: a good attempt.

"The adversary never learns anything *new* about the plaintext by looking at the ciphertext."

This is called *semantic security.* A very informal way to state it:

Definition 1. (very informal) An encryption scheme is **semantically secret** if, for all choices of:

- adversary $\boldsymbol{A}$,
- message m,
- "prior information" function g, and
- "target information" function f,

the following property holds:

$$\Pr\left[f(m) \leftarrow \boldsymbol{A}\left(g(m), \mathbf{Enc}_{\boldsymbol{k}}(m)\right)\right] = \Pr\left[f(m) \leftarrow \boldsymbol{A}\left(g(m)\right)\right].$$

"Look, I studied the ciphertext carefully and learned something interesting about the plaintext!"

"Actually, you could have learned it without looking at the ciphertext at all!"

Super complicated! And we haven't even properly formalized it…

Encryption: Secrecy: Candidate II

Secrecy: "perfect secrecy" (KL p.29)

Definition 2. An encryption scheme (**KeyGen**, **Enc**, **Dec**) is **perfectly secret** if, for every plaintext distribution $\mathcal{M}$, every plaintext m, and every ciphertext c,

$$\underbrace{\Pr[M = m \mid C = c]}_{\text{"The probability that the plaintext is some particular } m\text{, if you DID see the ciphertext."}} = \underbrace{\Pr[M = m]}_{\text{"The probability that the plaintext is some particular } m\text{, if you DID NOT see the ciphertext."}}.$$

What does the notation mean? This is the random experiment:

- Sample a uniformly random key $k \leftarrow$ **KeyGen**;
- Get a sample from the random variable M with distribution $\mathcal{M}$;
- Run encryption $\mathbf{Enc}_{\boldsymbol{k}}$ on the sample; the result is the random variable C;

Sounds like semantic secrecy, but without all the baggage. Good enough?

Encryption: Secrecy: Candidate III

Secrecy: what about this one?

Definition 3. An encryption scheme $(\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ is **perfectly secret** if, for every plaintext distribution $\mathcal{M}$, every plaintext pair m, m', and every ciphertext c,

$$\Pr_k[\mathbf{Enc}_k(m) = c] = \Pr_k[\mathbf{Enc}_k(m') = c]$$

Something like: "If the key is secret, then the distribution of ciphertexts is independent of the message."

Looks pretty good too. Is it right?

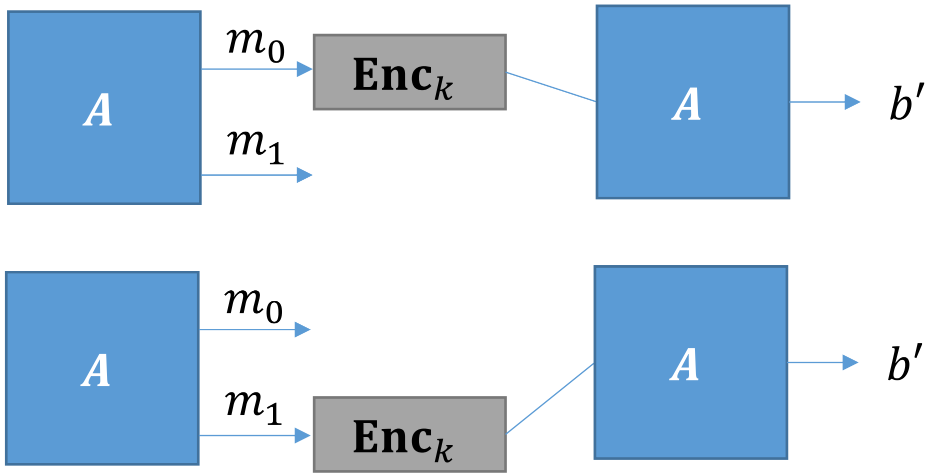
Encryption: Secrecy: Candidate IV

Secrecy: let's do it with an experiment (or game, if you like.)

Indistinguishability experiment (IND).

1. we sample a key $k \leftarrow \mathbf{KeyGen}$;
2. adversary (Eve) $\boldsymbol{A}$ outputs two messages m_0, m_1;
3. we flip a uniform coin $b \leftarrow \{0,1\}$;
4. we give $\boldsymbol{A}$ the ciphertext $c \leftarrow \mathbf{Enc}_k(m_b)$;
5. $\boldsymbol{A}$ outputs a bit b'.

We say $\boldsymbol{A}$ wins if $b = b'$.



Definition 4. An encryption scheme $(\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ has **perfectly indistinguishable ciphertexts** if, for every adversary $\boldsymbol{A}$,

$$\Pr_k[\boldsymbol{A} \text{ wins IND}] = \frac{1}{2}.$$

Encryption: Secrecy

Surprise: (I know, not really…)

Theorem 1. Definitions 1-4 are all equivalent. In particular,

$$\text{semantic secrecy} \iff \text{perfect secrecy} \iff \text{perfectly indistinguishable ciphertexts.}$$

- proof is not very hard; some parts in book, others in homework;
- studying how the proofs work is worthwhile.

This is awesome:

- each definition comes with some natural intuition: a secure scheme *should* satisfy it;
- that they are all equivalent is an indication that we are on to a *good notion;*
- the definitions are reasonably different in form; as a result, they will be useful in different situations;
- some have an explicit adversary, others do not!
- you can pick which one to use depending on context.

Encryption: Secrecy Of One-time Pad

Example: one-time pad.

Which definition should we use? Let's do **Definition 3**: $\Pr_k[\mathbf{Enc}_k(m) = c] = \Pr_k[\mathbf{Enc}_k(m') = c]$.

Simple argument:

- $k \leftarrow \{0,1\}^n$ is a uniformly random bitstring.
- for any fixed x, observe that $x \oplus k$ is also uniformly random in $\{0,1\}^n$.
- in particular, $\Pr_k[x \oplus k = c] = 2^{-n}$ for any $c \in \{0,1\}^n$.
- but this holds *for any fixed* x. In particular, it holds for both m and m' from the setup in Definition 3.

It follows that

$$\Pr_k[\mathbf{Enc}_k(m) = c] = \frac{1}{2^n} = \Pr_k[\mathbf{Enc}_k(m') = c]$$

So the one-time pad is *perfectly secret,* and (by **Theorem 1**) all those other great things too.

So we have *perfectly secure, unbreakable encryption!* Is the course over?

One Time Pad

Perfect Secrecy Limitations

Theorem: If (Gen,Enc,Dec) is a perfectly secret encryption scheme then

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

One Time Pad Limitations

- The key is as long as the message
  - How to exchange long messages?
  - Need to exchange/secure lots of one-time pads!
- OTPs can only be used once
  - As the name suggests
- VENONA project (US + UK)
  - Decrypt ciphertexts sent by Soviet Union which were mistakenly encrypted with portions of the same one-time pad over several decades

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'$$

Shannon's Theorem

Theorem: Let (Gen,Enc,Dec) be an encryption scheme with $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$. Then the scheme is perfectly secret if and only if:

1. Every key $k \in \mathcal{K}$ is chosen with (equal) probability $1/_{|\mathcal{K}|}$ by the algorithm Gen, and
2. For every $m \in \mathcal{M}$ and every $c \in \mathcal{C}$ there exists a unique key $k \in \mathcal{K}$ such that $\mathrm{Enc}_k(m)=c$.

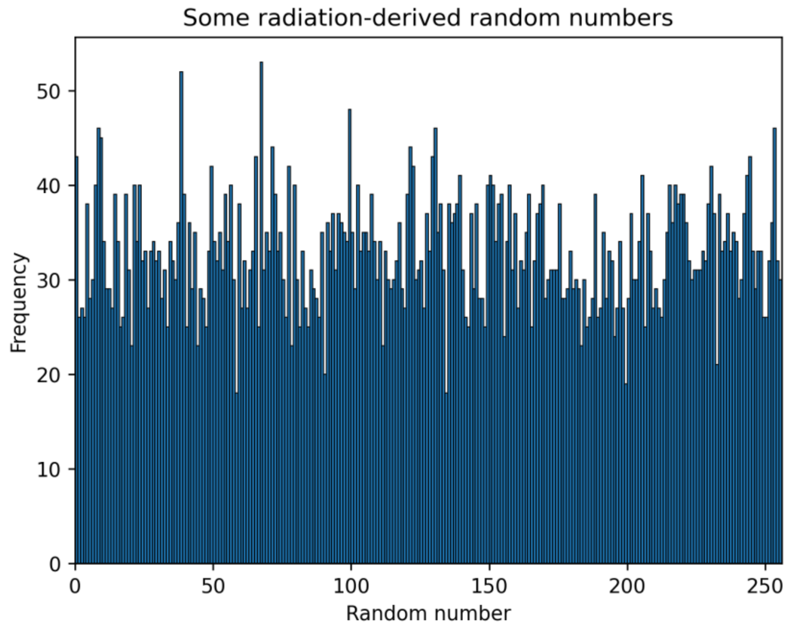
An Important Remark on Randomness

- In our analysis we have made (and will continue to make) a key assumption:
- We have access to true "randomness" to generate a secret key K

  Example: K = one time pad

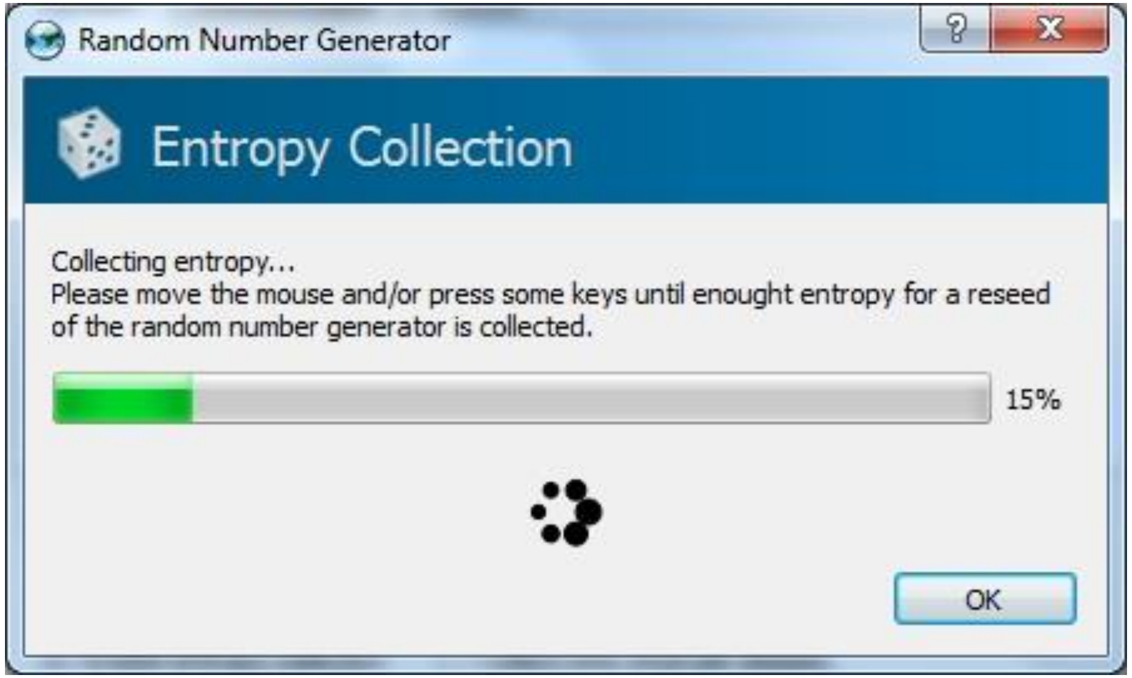
- Independent Random Bits
  - Unbiased Coin flips
  - Radioactive decay?

Radioactive decay is the process where unstable atomic nuclei lose energy by emitting radiation, transforming into more stable atoms.



In Practice

- Hard to flip thousands/millions of coins
- Mouse-movements/keys
  - Uniform bits?
  - Independent bits?
- Use Randomness Extractors
  - As long as input has high entropy, we can extract (almost) uniform/independent bits
  - Hot research topic in theory

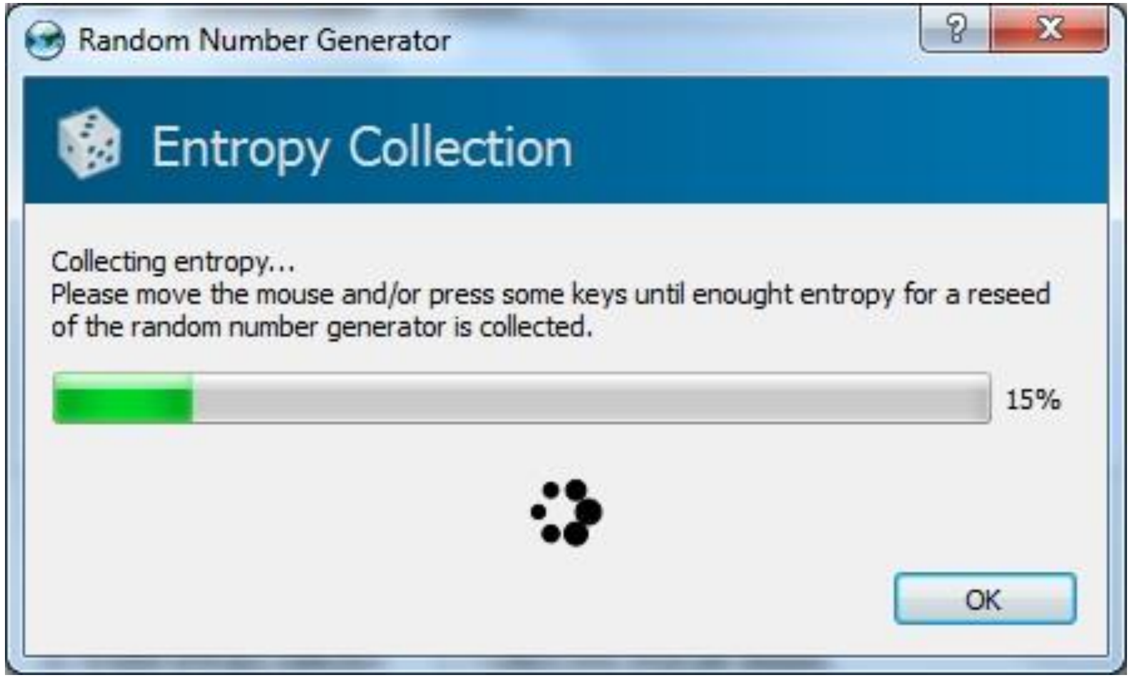


In Practice

- Hard to flip thousands/millions of coins

- Mouse-movements/keys

- Customized Randomness Chip?



Caveat: Don't do this!

- Rand() in C stdlib.h is no good for cryptographic applications

- Source of many real world flaws

How to Read a Paper

First Pass:

- Title, Abstract
- Figures (illustrations? important results?)
- skim intro & conclusions
- References

Second Pass

- Intro in details
- Overview, related work, or background sections
- Figures in details

Third pass:

- Read in detail
- Mark references for future read

How to Review a Paper

How to think when reviewing a paper?

1) Motivation
Is this an important problem?

- New problem? → Worthwhile or artificial?
- Existing problem? (i.e., have others worked on it) → Does it improve over prior work?

2) Related Work

- Does it really outperform prior work?
- Does it accurately represent prior work?
- Do you know past work? If not, search Google Scholar to get a sense of past work

3) Techniques

- Are they novel? intellectually interesting?
- Are they technically sound? Is there a key technical flaw?

How to Review a Paper

How to think when reviewing a paper?

4) Implementation

- Significant effort?
- Matches the motivation?
- Simulation or real-world prototype?

5) Evaluation

- Comprehensive? Convincing?
- Does the system deliver what it promises?

How to Review a Paper

How to think when reviewing a paper?

1) Motivation

2) Related Work

3) Techniques

4) Implementation

5) Evaluation

How to Review a Paper

How to write a review?

1) Summary

2) Strengths & Weaknesses

3) Comments to authors

How to Review a Paper

How to write a review?

1) Summary

- 5-10 sentences
- If someone hasn't read the paper at all, they should understand what it's about
- Should sound like a "brutally honest and straightforward abstract"

Rough structure:

This paper presents XXX, a system that does YYY. **The goal is to** XXX. The **main challenge** the authors try to address is YYY.

The key idea is to do XXX. The authors do this by introducing/proposing ZZZ

The authors implement (or simulate) their system and **demonstrated** (results) that it outperforms the baseline?

How to Review a Paper

How to write a review?

1) Summary

- 5-10 sentences
- If someone hasn't read the paper at all, they should understand what it's about
- Should sound like a "brutally honest and straightforward abstracT"

2) Strengths & Weaknesses

- Use your answers to the questions of "How to think when reviewing"
- List 2-4 pros/cons
- Each should be a direct statement about the paper

Rough structure:

Pros:
+ Statement 1
+ Statement 2

Cons:
-
-

How to Review a Paper

How to write a review?

1) Summary

2) Strengths & Weaknesses

3) Comments to authors

- Detailed comments to authors
- Elaborate on your pros/cons, areas for improvement, key concerns
- Ask questions about techniques, figures, results, etc.
- Based on the 5 points from how to think as well as technical details

Examples:

- If you listed a weaknesses small delta over prior work, specify in details why with references
- If experimental details are missing, state exactly what is missing and why it is problematic
- Include typos/grammar mistakes, potential suggestions to correct

How to Review a System Paper

How to write a review?

1) Summary

2) Strengths & Weaknesses

3) Comments to authors

- Detailed comments to authors
- Elaborate on your pros/cons, areas for improvement, key concerns
- Ask questions about techniques, figures, results, etc.
- Based on the 5 points from how to think as well as technical details

Examples:

- If you listed a weaknesses small delta over prior work, specify in details why with references
- If experimental details are missing, state exactly what is missing and why it is problematic
- Include typos/grammar mistakes, potential suggestions to correct

For the sake of this class, we will drop "comments to authors".

- If you could improve this paper, how
- How do you envision your proposed technique will improve the work

How to Review a Paper

How to write a review? (for this class)

1) Summary

2) Strengths & Weaknesses

3) Suggestions for Improvement

Questions?