Perfect Secrecy, One Time Pad, Randomness
Recap

Consider the following encryption Schemes:

1. Shift Cipher: Crackable. Keyspace has only 26 elements.
2. Affine Cipher: Crackable. Keyspace has only 312 elements.
3. Vig Cipher: Crackable by repeats and letter freqs.
5. Matrix Cipher: Crackable if know (ENC,DEC)-pairs.
6. One-Time Pad: Uncrackable!
7. ElGamal: Uncrackable if we make hardness assumptions.
8. RSA: Uncrackable if we make hardness assumptions.
Recap

Consider the following encryption Schemes:

1. Shift Cipher: Crackable. Keyspace has only 26 elements.
2. Affine Cipher: Crackable. Keyspace has only 312 elements.
3. Vig Cipher: Crackable by repeats and letter freqs.
5. Matrix Cipher: Crackable if know (ENC,DEC)-pairs.
6. One-Time Pad: Uncrackable!
7. ElGamal: Uncrackable if we make hardness assumptions.
8. RSA: Uncrackable if we make hardness assumptions.

All of the above are true.
Recap

Consider the following encryption Schemes:

1. Shift Cipher: Crackable. Keyspace has only 26 elements.
2. Affine Cipher: Crackable. Keyspace has only 312 elements.
3. Vig Cipher: Crackable by repeats and letter freqs.
5. Matrix Cipher: Crackable if know (ENC,DEC)-pairs.
6. One-Time Pad: Uncrackable!
7. ElGamal: Uncrackable if we make hardness assumptions.
8. RSA: Uncrackable if we make hardness assumptions.

All of the above are true.
None of the above are rigorous!
Recap

Consider the following encryption Schemes:

1. Shift Cipher: Crackable. Keyspace has only 26 elements.
2. Affine Cipher: Crackable. Keyspace has only 312 elements.
5. Matrix Cipher: Crackable if know (ENC,DEC)-pairs.
6. One-Time Pad: Uncrackable!
7. ElGamal: Uncrackable if we make hardness assumptions.
8. RSA: Uncrackable if we make hardness assumptions.

All of the above are true.
None of the above are rigorous!
We make them ... more rigorous
Assumptions

1. Cryptography requires computational assumptions
2. Examples: Factoring is not in polynomial time.

Principle: Need assumptions to be explicit:

1. Allow researchers to (attempt to) validate assumptions by studying them
2. Allow meaningful comparison between schemes based on different assumptions
3. Useful to understand minimal assumptions needed
4. Practical implications if assumptions are wrong
5. Enable proofs of security
Proofs of Security/Limitations

Proofs give guarantee of sec
Proofs of Security/Limitations

Proofs give guarantee of security relative to the model of security!

1. If the model does not correspond to real-world threats.
2. If the attacker is "outside model" (e.g., Timing Attacks).
3. If the assumptions are invalid.
4. If the implementation is flawed.

Attacks can come from unexpected places: Look up the story of the Maginot Line, an immense wall that France built to deter a German Invasion. It didn't work.

This does not detract from the importance of having formal definitions in place.

This does not detract from the importance of proofs of security.
Proofs of Security/Limitations

Proofs give guarantee of security relative to the model of security!
Provably secure schemes can be broken!

1. If model does not correspond to real-world threats.
2. If attacker is "outside model" (e.g., Timing Attacks)
3. If the assumptions are invalid
4. If the implementation is flawed

Attacks can come from unexpected places: Look up the story of the Maginot Line, an immense wall that France built to deter a German Invasion. It didn’t work.

This does not detract from the importance of having formal definitions in place
This does not detract from the importance of proofs of security
Proofs of Security/Limitations

Proofs give guarantee of sec relative to the model of security! Provably secure schemes can be broken!

1. If model does not correspond to real-world threats.
2. If attacker is ”outside model” (e.g., Timing Attacks)
3. If the assumptions are invalid
4. If the implementation is flawed

Attacks can come from unexpected places: Look up the story of the Maginot Line, an immense wall that France build to deter a German Invasion. It didn’t work.

1. This does not detract from the importance of having formal definitions in place
2. This does not detract from the importance of proofs of security
Defining secure encryption
Crypto definitions (generally)

- Security guarantee/goal
  - What we want to achieve and/or what we want to prevent the attacker from achieving

- Threat model
  - What (real-world) capabilities the attacker is assumed to have
Threat models for encryption

- **Ciphertext-only attack (CTA).** As name indicates, Eve only has access to the ciphertext. Eve can crack Shift, Affine, Vig, Gen. Matrix might be an open problem.

- **Known-plaintext attack (KPA).** Eve has access to previous plaintexts and what the ciphertext was. Matrix can be cracked this way if text is long enough.

- **Chosen-plaintext attack (CPA).** Eve can fool Alice into encoding a particular plaintext.

- **Chosen ciphertext attack (CCA).** Eve can fool Bob into telling her what a particular ciphertext decodes to.
Goal of secure encryption?

- How would you define what it means for encryption scheme $(\text{GEN}, \text{ENC}, \text{DEC})$ over message space $\mathcal{M}$ to be secure?
  - Against a (single) ciphertext-only attack
Secure encryption?

- “Impossible for the attacker to learn the key”
  - The key is a *means to an end*, not the end itself
  - Necessary (to some extent) but not sufficient
  - Easy to design an encryption scheme that hides the key completely, but is insecure
  - Can design schemes where most of the key is leaked, but the scheme is still secure
Secure encryption?

“Impossible for the attacker to learn the plaintext from the ciphertext”

What if the attacker learns 90% of the plaintext?
Secure encryption?

• “Impossible for the attacker to learn any character of the plaintext from the ciphertext”

  • What if the attacker is able to learn (other) partial information about the plaintext?

    • e.g. salary is greater than $75K

  • What if the attacker guesses a character correctly?
Perfect Secrecy
Perfect secrecy

“Regardless of any prior information the attacker has about the plaintext, the ciphertext should leak no additional information about the plaintext”

- The right notion!
- How to formalize?
Probability review

- **Event**: a particular occurrence in some experiment
  - \( \Pr[E] \): probability of event E

- Conditional probability: probability that one event occurs, *given that some other event occurred*

\[
\Pr[A|B] = \frac{\Pr[A \land B]}{\Pr[B]}
\]
Recall

A *private-key encryption scheme* is defined by a message space \( \mathcal{M} \) and algorithms (GEN, ENC, DEC):

- **GEN** (key generation algorithm) generates \( k \)
- **ENC** (encryption algorithm): takes key \( k \) and message \( m \in \mathcal{M} \) as input; outputs ciphertext \( c \)
  
  \[
  c \leftarrow ENC_k(m)
  \]

- **DEC** (decryption algorithm): takes key \( k \) and ciphertext \( c \) as input; outputs \( m \)
  
  \[
  m \leftarrow DEC_k(c)
  \]
Notation

- $\mathcal{K}$ (key space) — set of all possible keys
- $\mathcal{M}$ (message space) — set of all possible messages
- $\mathcal{C}$ (ciphertext space) — set of all possible ciphertexts
Perfect secrecy

Informal: Let $m$ be a message. Before Eve sees the ciphertext she knows $\Pr(M = m)$. After Eve sees the ciphertext we want her to not gain any knowledge whatsoever.

Formal: Encryption scheme (GEN, ENC, DEC) with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is perfectly secret if for every distribution over $\mathcal{M}$, every $m \in \mathcal{M}$, and every $c \in \mathcal{C}$ with $\Pr[C = c] > 0$, it holds that

$$\Pr[M = m | C = c] = \Pr[M = m]$$

The distribution of $M$ does not change when Eve sees the ciphertext.
Be Impressed!

In Mathematics often getting the right definition is the hard part!

That we have a way of formally defining Perfect Secrecy is very impressive!

This definition is one of the things that marks the line between Classical and Modern Cryptography.
Bayes’s theorem

\[ \Pr[A|B] = \Pr[B|A] \cdot \frac{\Pr[A]}{\Pr[B]} \]

Note: This is very useful in both this course and in life.
Example of Application of Bayes’s theorem

\[ \Pr[A|B] = \Pr[B|A] \cdot \frac{\Pr[A]}{\Pr[B]} \]. There are two coins:

1) Coin F is fair: \( \Pr(H) = \Pr(T) = \frac{1}{2} \).
2) Coin B is biased: \( \Pr(H) = \frac{3}{4}, \Pr(T) = \frac{1}{4} \).

Alice picks coin at random, flips 10 times, gets all H. Is the coin definitely biased?
Example of Application of Bayes’s theorem

\[
\Pr[A|B] = \Pr[B|A] \cdot \frac{\Pr[A]}{\Pr[B]}.
\]

There are two coins:

1) Coin F is fair: \( \Pr(H) = \Pr(T) = \frac{1}{2} \).
2) Coin B is biased: \( \Pr(H) = \frac{3}{4}, \Pr(T) = \frac{1}{4} \).

Alice picks coin at random, flips 10 times, gets all H. Is the coin definitely biased? No.
**Example of Application of Bayes’s theorem**

\[ \Pr[A|B] = \Pr[B|A] \cdot \frac{\Pr[A]}{\Pr[B]} \]. There are two coins:

1) Coin F is fair: \( \Pr(H) = \Pr(T) = \frac{1}{2} \).
2) Coin B is biased: \( \Pr(H) = \frac{3}{4}, \Pr(T) = \frac{1}{4} \).

Alice picks coin at random, flips 10 times, gets all H. Is the coin definitely biased? No.

What is Prob that it is biased? VOTE:

1. Between 0.99 and 1.0
2. Between 0.98 and 0.99
3. Between 0.97 and 0.98
4. Less than 0.97
Example of Application of Bayes’s theorem

\[ \Pr[A|B] = \Pr[B|A] \cdot \frac{\Pr[A]}{\Pr[B]} \]. There are two coins:

1) Coin F is fair: \( \Pr(H) = \Pr(T) = \frac{1}{2} \).
2) Coin B is biased: \( \Pr(H) = \frac{3}{4}, \Pr(T) = \frac{1}{4} \).

Alice picks coin at random, flips 10 times, gets all H. Is the coin definitely biased? No.

What is Prob that it is biased? VOTE:

1. Between 0.99 and 1.0
2. Between 0.98 and 0.99
3. Between 0.97 and 0.98
4. Less than 0.97

We will see that it is 0.982954, so between 0.98 and 0.99.
Example of Application of Bayes’s theorem

\[
\Pr(B|H^{10}) = \frac{\Pr(B)\Pr(H^{10}|B)}{\Pr(H^{10})}
\]

\[
\Pr(B) = \frac{1}{2}
\]

\[
\Pr(H^{10}|B) = \left(\frac{3}{4}\right)^{10}
\]

\[
\Pr(H^{10}) = \Pr(H^{10} \cap F) + \Pr(H^{10} \cap B)
\]

\[
\Pr(H^{10} \cap F) = \Pr(H^{10}|F)\Pr(F) + \Pr(H^{10}|B)\Pr(B) = \frac{1}{2}\left(\left(\frac{1}{2}\right)^{10} + \left(\frac{3}{4}\right)^{10}\right)
\]

Put it together to get

\[
\Pr(B|H^{10}) = \frac{1}{1 + (2/3)^{10}} = 0.982954.
\]
Example of Application of Bayes’s theorem

$$\Pr(B|H^{10}) = \frac{\Pr(B)\Pr(H^{10}|B)}{P(H^{10})}$$

$$\Pr(B) = \frac{1}{2}$$

$$\Pr(H^{10}|B) = \left(\frac{3}{4}\right)^{10}$$

$$\Pr(H^{10}) = \Pr(H^{10} \cap F) + \Pr(H^{10} \cap B)$$

$$\Pr(H^{10} \cap F) = \Pr(H^{10}|F)\Pr(F) + \Pr(H^{10}|B)\Pr(B) =$$

$$\frac{1}{2}\left(\left(\frac{1}{2}\right)^{10} + \left(\frac{3}{4}\right)^{10}\right)$$

Put it together to get

$$\Pr(B|H^{10}) = \frac{1}{1 + (2/3)^{10}} = 0.982954.$$
One-time pad

- Let $m = \{0, 1\}^n$
- $Gen$: choose a uniform key $k \in \{0, 1\}^n$
- $Enc_k(m) = k \oplus m$
- $Dec_k(c) = k \oplus c$
- Correctness:

\[
Dec_k(Enc_k(m)) = k \oplus (k \oplus m) \\
= (k \oplus k) \oplus m \\
= m
\]
One-time pad

Diagram:
- n bits
- key
- n bits
- message
- n bits
- ciphertext

Diagram shows the process of encrypting a message using a one-time pad.
Perfect secrecy of one-time pad

- Note that *any* observed ciphertext can correspond to *any* message (why?)
  - (This is necessary, but not sufficient, for perfect secrecy)

- So, having observed a ciphertext, the attacker cannot conclude for certain which message was sent
Perfect secrecy of one-time pad for $n$-bit messages

Fix arbitrary distribution over $M = \{0, 1\}^n$, and arbitrary $m, c \in \{0, 1\}^n$

Want: $\Pr[M = m|C = c] = \Pr[M = m]$

By Bayes’s Theorem:
$\Pr[M = m|C = c] = \Pr[C = c|M = m] \cdot \frac{\Pr[M = m]}{\Pr[C = c]}$

So need

1. $\Pr[C = c|M = m] = \Pr[K = m \oplus c] = 2^{-n}$
2. $\Pr[M = m]$. DO NOT KNOW. Arbitrary Distribution!
3. $\Pr[C = c] = \Pr[c = K \oplus m] = \Pr[K = m \oplus c] = 2^{-n}$

Hence: $\Pr[M = m|C = c] = 2^{-n} \cdot \frac{\Pr[M = m]}{2^{-n}} = \Pr[M = m]$. 
One-time pad

1. The one-time pad achieves perfect secrecy!
2. One-time pad has historically been used in the real world E.g. red phone between DC and Moscow
3. It is not widely used today. Why not?
One-time pad

1. The one-time pad achieves perfect secrecy!
2. One-time pad has historically been used in the real world E.g. red phone between DC and Moscow
3. It is not widely used today. Why not?

Drawbacks:
1. Key as long as the message
2. Only secure if each key is used to encrypt once
3. Generating perfectly random bits is hard!
One-time pad

1. The one-time pad achieves perfect secrecy!
2. One-time pad has historically been used in the real world E.g. red phone between DC and Moscow
3. It is not widely used today. Why not?

Drawbacks:
1. Key as long as the message
2. Only secure if each key is used to encrypt once
3. Generating perfectly random bits is hard!

Are there any other schemes that are perfectly secure! Vote:

1. YES
2. NO
3. OTHER
One-time pad

1. The one-time pad achieves perfect secrecy!
2. One-time pad has historically been used in the real world E.g. red phone between DC and Moscow
3. It is not widely used today. Why not?

Drawbacks:
1. Key as long as the message
2. Only secure if each key is used to encrypt once
3. Generating perfectly random bits is hard!

Are there any other schemes that are perfectly secure? Vote:

1. YES
2. NO
3. OTHER

NO.
Alice wants to send 10-bit message to Bob. Use (Gen, Enc, Dec).
Assume number of keys $< 2^{10} = 1024$. Say 1023.
Will information be leaked? Discuss
Optimality of the one-time pad – Example

Alice wants to send 10-bit message to Bob. Use (Gen, Enc, Dec). Assume number of keys $< 2^{10} = 1024$. Say 1023.
Will information be leaked? Discuss YES
Optimality of the one-time pad – Example

Alice wants to send 10-bit message to Bob. Use (Gen, Enc, Dec). Assume number of keys $< 2^{10} = 1024$. Say 1023. Will information be leaked? Discuss YES

Eve sees Alice send Bob $c$. Eve knows $\mathcal{K} = \{k_1, \ldots, k_{1023}\}$. Eve computes $\text{Dec}_{k_1}(c), \text{Dec}_{k_2}(c), \ldots, \text{Dec}_{k_{1023}}(c)$ Let $m'$ be the one message that Eve did NOT get. Eve knows $m \neq m'$. This is a leak!

Hence $\mathcal{K}$ must be of size $2^{10}$ to avoid having a leak!
Theorem: If \((\text{Gen}, \text{Enc}, \text{Dec})\) with message space \(\mathcal{M}\) is perfectly secret, then \(|\mathcal{K}| \geq |\mathcal{M}|\).

Proof: Similar to last slide. Might be HW.

Upshot: If \((\text{Gen}, \text{Enc}, \text{Dec})\) has perfect secrecy then \(|\mathcal{K}| \geq |\mathcal{M}|\). Hence is 1-time pad or variant (omit proof).
1-Time Pad is the Gold Standard

The 1-time pad is hard to really do.

However, it gives us a target.

In future we will ask

Is this encryption system 1-time-pad-like?
Where do we stand?

- Defined perfect secrecy
- One-time pad achieves it!
- One-time pad is optimal!
- Are we done...?
Perfect secrecy

- Requires that \textit{absolutely no information} about the plaintext is leaked, even to eavesdroppers \textit{with unlimited computational power}
  - Has some inherent drawbacks
  - Seems unnecessarily strong

Two directions to go

1. Try to generate random bits so can use 1-time pad (do now).
2. Try to relax definition of \textit{Perfect Secrecy} so that achievable and secure (do later).
A brief detour: randomness generation
Key generation

- When describing algorithms, we assume access to uniformly distributed bits/bytes
- Where do these actually come from?
- Random-number generation
Random-number generation

- Precise details depend on the system
  - Linux or unix: /dev/random or /dev/urandom
  - **Do not use rand() or java.util.Random**
    Not as random as the name would indicate!
  - Use crypto libraries instead
Random-number generation

- Two steps:
  1. Continually collect ‘unpredictable” data.
  2. Correct biases in it to make it more random. Called smoothing.

Unpredictable: Different models.

1. There is a $0 < p < 1$ such that each bit has
   $$\Pr(1) = p, \Pr(0) = 1 - p.$$  
   Note that bits are independent. $p$ is not known. We will only deal with this.

2. Not independent but simple dependency. For example, if $b_i = 1$ then $\Pr(b_{i+1} = 1) = p$.

3. Complicated dependencies. Depends on last $x$ bits.
Random-number generation
Smoothing via Von Neumann Technique (VN)

- Need to eliminate both *bias* and *dependencies*

- VN technique for eliminating bias:
  - Collect two bits per output bit
    - $01 \mapsto 0$
    - $10 \mapsto 1$
    - $00, 11 \mapsto$ skip
  - Note that this assumes *independence* (as well as constant bias)
Assume that $\Pr(b = 0) = p$ and $\Pr(b = 1) = 1 - p$.

If flip 2 coins then

$$\Pr(01) + \Pr(10) = p(1 - p) + (1 - p)p = 2p(1 - p).$$

If flip $2n$ coins then expected number of random bits is $2np(1 - p)$. 
How Good is VN Method?

If flip 14 coins ($n = 7$) then we get the following graph:
Step 2: Smoothing via Elias. Prepossess

1. Of the $\binom{7}{3} = 35$ elts of $\{0, 1\}^7$ with 4 0’s and 3 1’s, toss 3 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^5$.

2. Of the $\binom{7}{3} = 35$ elts of $\{0, 1\}^7$ with 3 0’s and 4 1’s, toss 3 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^5$.

3. Of the $\binom{7}{2} = 21$ elts of $\{0, 1\}^7$ with 5 0’s and 2 1’s, toss 5 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^4$.

4. Of the $\binom{7}{2} = 21$ elts of $\{0, 1\}^7$ with 2 0’s and 5 1’s, toss 5 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^4$.

5. Of the $\binom{7}{1} = 7$ elts of $\{0, 1\}^7$ with 6 0’s and 1 1’s, toss 3 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^2$.

6. Of the $\binom{7}{1} = 7$ elts of $\{0, 1\}^7$ with 1 0’s and 6 1’s, toss 3 of them out. Let $B$ be a bijection from what’s left to $\{0, 1\}^2$.

Sequences tossed out are called **bad**
Step 2: Smoothing via Elias

Assume that \( \Pr(b = 0) = p \) and \( \Pr(b = 1) = 1 - p \).

1. Flip 7 coins. Let the sequence be \( s \).
2. If \( s \) is bad then goto step 1.
3. Output \( B(s) \). (could be 2,4, or 5 bits).

Let \( X \) be the number of bits.
Expected Number of Random Bits

\[ E(X) = 5\text{Pr}(X = 5) + 4\text{Pr}(X = 4) + 2\text{Pr}(X = 2) \]

\[ 5\text{Pr}(X = 5) = 5 \times (32p^4(1 - p)^3 + 32p^3(1 - p)^4) = 160p^3(1 - p)^3 \]

\[ 4\text{Pr}(X = 4) = 4 \times (16p^5(1 - p)^2 + 16p^2(1 - p)^5) = 64p^2(1 - p)^2(p^3 + (1 - p)^3) \]

\[ 2\text{Pr}(X = 2) = 2 \times (4p^6(1 - p) + 4p(1 - p)^6) = 8p(1 - p)(p^5 + (1 - p)^5) \]

\[ E(X) = -8p^6 + 24p^5 - 40p^3 + 16p^3 + 8p \]
How good is Elias Method

If flip 14 bits:

Much better than VN. Can we do better? Discuss.
VN vs GMS

If we flip 14 bits:
Is Elias Actually Used?

No

Discuss why
Is Elias Actually Used?

No

Discuss why

1. Assumes independent bits with constant bias.
2. Need to wait for all 7 flips to get some bits.
3. If \( p = 0.3 \) then 14 flips yields only \( \sim 4 \) random bits.
Is Elias Actually Used?

No
Discuss why

1. Assumes independent bits with constant bias.
2. Need to wait for all 7 flips to get some bits.
3. If $p = 0.3$ then 14 flips yields only $\sim 4$ random bits. Can improve this (HW).
Is Elias Actually Used?

No

Discuss why

1. Assumes independent bits with constant bias.
2. Need to wait for all 7 flips to get some bits.
3. If $p = 0.3$ then 14 flips yields only $\sim 4$ random bits.
   Can improve this (HW).
4. Perfect randomness not really needed
Is Elias Actually Used?

No
Discuss why

1. Assumes independent bits with constant bias.
2. Need to wait for all 7 flips to get some bits.
3. If $p = 0.3$ then 14 flips yields only $\sim 4$ random bits. Can improve this (HW).
4. Perfect randomness not really needed
5. Pseudorandomness good enough. We will discuss later.