Perfect Security, One-Time Pad, Randomness

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

- *Random variable (r.v.):* variable that takes on (discrete) values with certain probabilities
- Probability distribution for a r.v. specifies the probabilities with which the variable takes on each possible value
  - Each probability must be between 0 and 1
  - The probabilities must sum to 1

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*
  - $\Pr[A|B] = \frac{\Pr[A \wedge B]}{Pr[B]}$
- To RV's X, Y are *independent* if for all x, y: $\Pr[X = x|Y = y] = \Pr[X = x]$

  Important: X, Y independent if Knowing that $Y = y$ does not help you figure out if $X = x$.
  Discuss: Why will this notion be important for defininig perfect security?

Probability distributions

Normally a RV returns a number. We allow it to be a message. Below are all the messages I could send, with the prob that I send them (unrelated to whether they are true).

1. Today 456 went well. $\Pr = \frac{1}{2}$.
2. Today 456 went badly. $\Pr = \frac{1}{100}$.
3. All 456 students submitted HW Monday. $\Pr = \frac{1}{100}$.
4. I proved a new result in crypto. $\Pr = \frac{1}{50}$.
5. I proved a new result about The Muffin Problem. $\Pr = \frac{9}{25}$.
6. I saw a student in office hours. $\Pr = \frac{1}{10}$

Note:

Should we assume that Eve knows this distribution? Discuss.

Probability distributions

Normally a RV returns a number. We allow it to be a message. Below are all the messages I could send, with the prob that I send them (unrelated to whether they are true).

1. Today 456 went well. $\Pr = \frac{1}{2}$.
2. Today 456 went badly. $\Pr = \frac{1}{100}$.
3. All 456 students submitted HW Monday. $\Pr = \frac{1}{100}$.
4. I proved a new result in crypto. $\Pr = \frac{1}{50}$.
5. I proved a new result about The Muffin Problem. $\Pr = \frac{9}{25}$.
6. I saw a student in office hours. $\Pr = \frac{1}{10}$

Note:

Should we assume that Eve knows this distribution? Discuss. YES.
Recall that we assume Eve knows distribution of letters in English.

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

Distribution on Keys

- Let K be the random variable denoting the key
  - K ranges over $\mathcal{K}$
- Fix some encryption scheme (Gen, Enc, Dec)
  - Gen defines a probability distribution for K:

$$\Pr[K = k] = \Pr[\text{Gen outputs key k}]$$

    Usually Uniform.

Message and Key Independent

- Random variables M and K are *independent*
  - i.e., the message that a party sends does not depend on the key used to encrypt that message

Distribution on Ciphertext

- Fix some encryption scheme (Gen, Enc, Dec) and some distribution for M
- Consider the following (randomized) experiment:
  1. Choose a message m, according to the given distribution
  2. Generate a key k using Gen
  3. Compute $c \leftarrow Enc_k(m)$
- This defines a distribution on the ciphertext!
- Let C be a random variable denoting the value of the ciphertext in this experiment

Perfect secrecy (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

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

- i.e. the distribution of M does not change conditioned on observing the ciphertext

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

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

$\Pr(B) = \frac{1}{2}$
$\Pr(H^{10}|B) = (\frac{3}{4})^{10}$
$\Pr(H^{10}) = \Pr(H^{10} \cap F) + \Pr(H^{10} \cap B)$
$\Pr(H^{10} \cap F) = \Pr(H^{10}|F)\Pr(F) + \Pr(H^{10}|B)\Pr(B) =$
$\frac{1}{2}\left((\frac{1}{2})^{10} + (\frac{3}{4})^{10}\right)$
Put it together to get

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

Example of Application of Bayes's theorem

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

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

Put it together to get

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

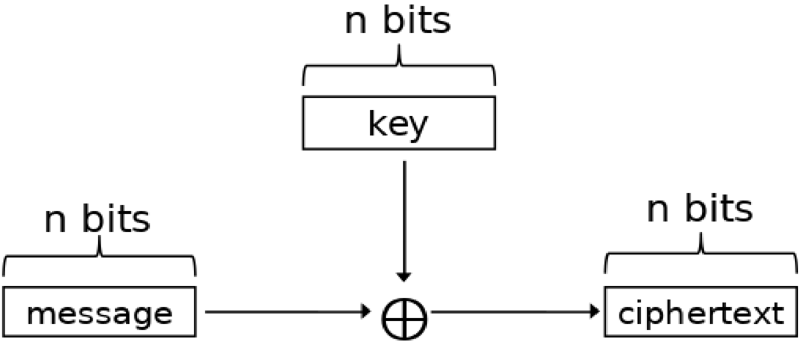
$$\mathrm{Pr}(B|H^{n}) = \frac{1}{1 + (2/3)^{n}}.$$

One-time pad

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

$$\begin{aligned} Dec_k(Enc_k(m)) &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m \\ &= m \end{aligned}$$

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 $\mathcal{M} = \{0,1\}^n$, 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.

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

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 $Dec_{k_1}(c), Dec_{k_2}(c), \ldots, Dec_{k_{1023}}(c)$
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!

Optimality of the one-time pad

Theorem: If (Gen, Enc, 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 (Gen, Enc, 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 *absolutely no information* about the plaintext is leaked, even to eavesdroppers *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 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

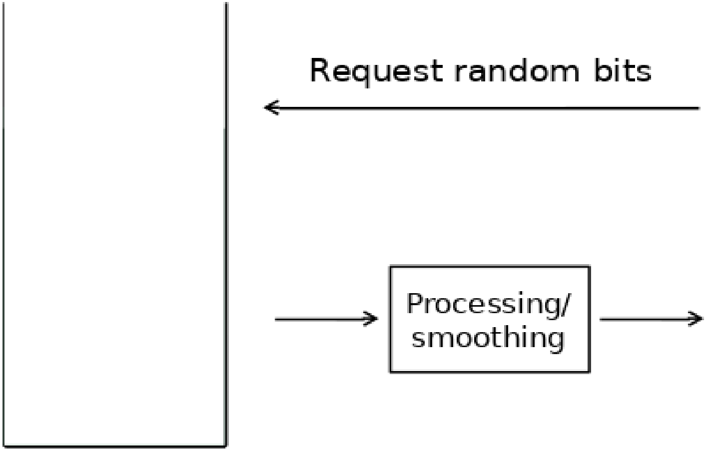
Random-number generation

- 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)

How Many Random Bits Can We Expect?

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

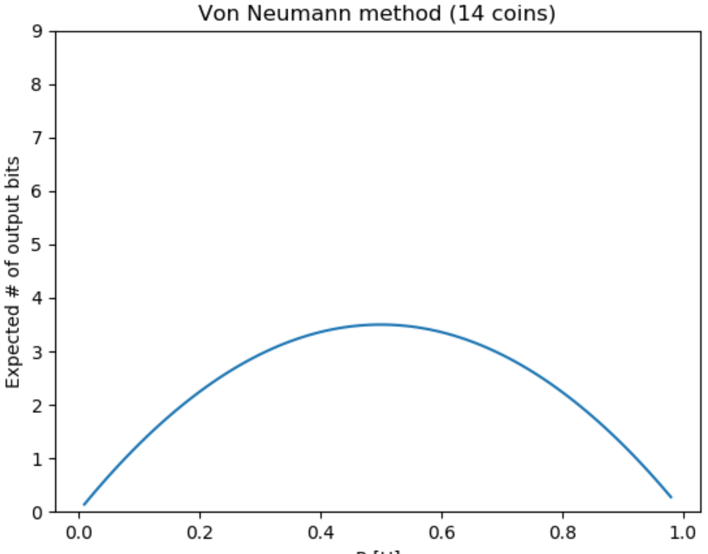
If flip 2 coins then

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

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 whats 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 whats 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 whats 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 whats 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 whats 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 whats 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

$$E(X) = 5\Pr(X = 5) + 4\Pr(X = 4) + 2\Pr(X = 2)$$

$$5\Pr(X = 5) = 5 \times (32p^4(1 - p)^3 + 32p^3(1 - p)^4) = 160p^3(1 - p)^3$$

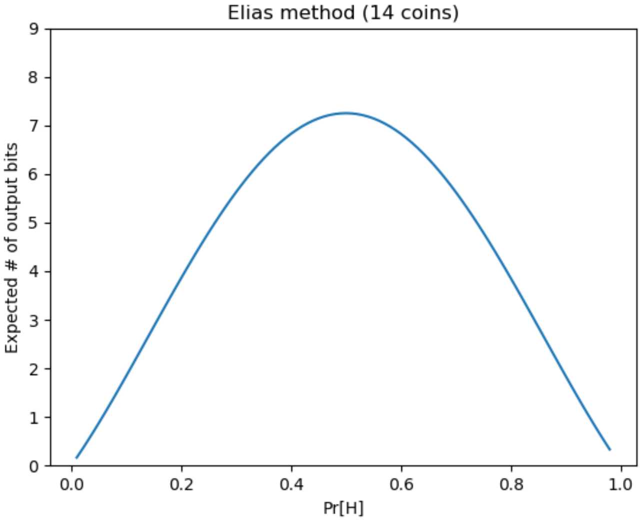
$$4\Pr(X = 4) = 4\times(16p^5(1-p)^2+16p^2(1-p)^5) = 64p^2(1-p)^2(p^3+(1-p)^3$$

$$2\Pr(X = 2) = 2\times(4p^6(1-p)+4p(1-p)^6) = 8p(1-p)(p^5+(1-p)^5)$$

$$E(X) = -8p^6 + 24p^5 - 40p^3 + 16p^3 + 8p$$

How good is Elias Method

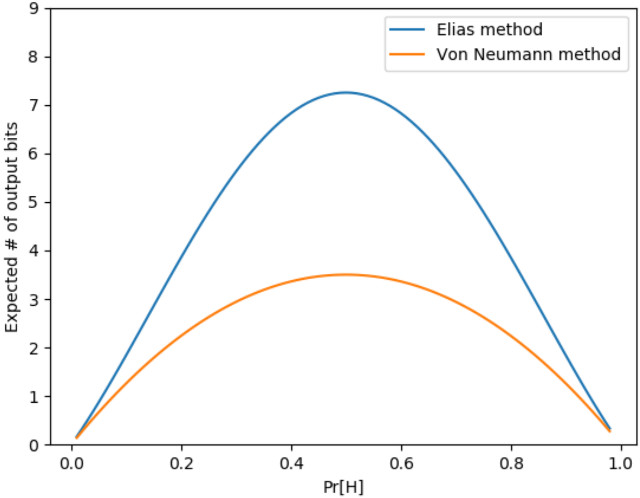
If flip 14 bits:



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 ~ 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 ~ 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 ~ 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 ~ 4 random bits. Can improve this (HW).
4. Perfect randomness not really needed
5. Pseudorandomness good enough. We will discuss later.