# Perfectly Secret Encryption

ENCS4320 - Applied Cryptography

Dr. Ahmed I. A. Shawahna

Electrical and Computer Engineering Department

Birzeit University

# Presentation Outline

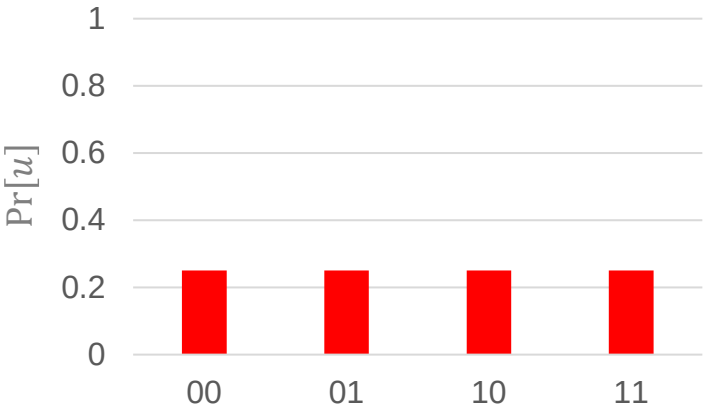
- **Basic Probability**
- Perfect Secrecy
- The One-Time Pad
- Crypto Requirements
- Crypto Taxonomy

# Discrete Probability

- Suppose that $\mathcal{U}$ is a finite set, e.g., $\mathcal{U} = \{0,1\}^n$

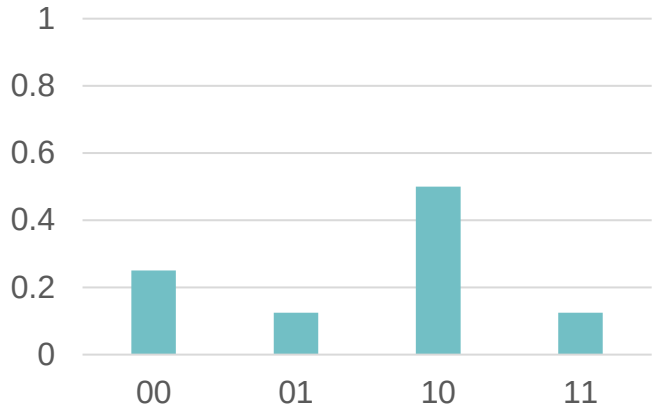
**Definition:** A **probability distribution** over $\mathcal{U}$ is a function $\Pr : \mathcal{U} \to [0,1]$ such that $\sum_{u \in \mathcal{U}} \Pr[u] = 1$

  - For example, $\mathcal{U} = \{0,1\}^2 = \{00, 01, 10, 11\}$ is the set of all possible outcomes

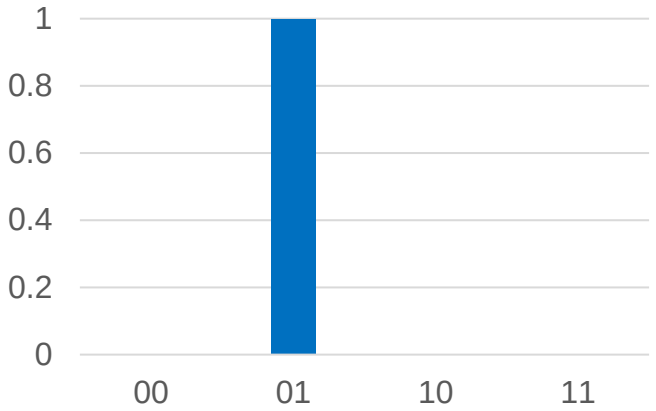



$\Pr[00] = 1/4$
$\Pr[01] = 1/4$
$\Pr[10] = 1/4$
$\Pr[11] = 1/4$

**Uniform distribution**




$\Pr[00] = 1/4$
$\Pr[01] = 1/8$
$\Pr[10] = 1/2$
$\Pr[11] = 1/8$

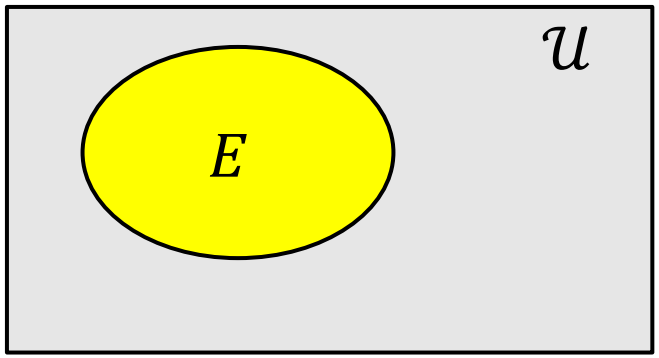



$\Pr[00] = 0$
$\Pr[01] = 1$
$\Pr[10] = 0$
$\Pr[11] = 0$

**Point distribution**

# Discrete Probability

- A subset $E \subseteq \mathcal{U}$ is called an **event**, and $\Pr[E] = \sum_{e \in E} \Pr[e]$




- If each outcome is equally likely, then the probability of event $E \subseteq \mathcal{U}$ is
  - $\Pr[E]$ = # elements in $E$ / # elements in $\mathcal{U}$
- For example, suppose we flip 2 coins, then $\mathcal{U} = \{hh, ht, th, tt\}$
  - Suppose $E$ = "at least one tail" = $\{ht, th, tt\}$
  - Then, $\Pr[E] = 3/4$

# Exercise - Discrete Probability

Suppose $\mathcal{U} = \{0,1\}^8$, and $E = \{x \in \mathcal{U} \mid x = 11\text{xx xxxx}\}$, i.e., $E \subset \mathcal{U}$. With the uniform distribution over $\mathcal{U}$, what is $\Pr[E]$?

**Solution**:

$$\Pr[E] = \Pr[1100\ 0000] + \Pr[1100\ 0001] + \cdots + \Pr[1111\ 1111]$$

$$= 2^6 / 2^8$$

$$= 1/2^2$$

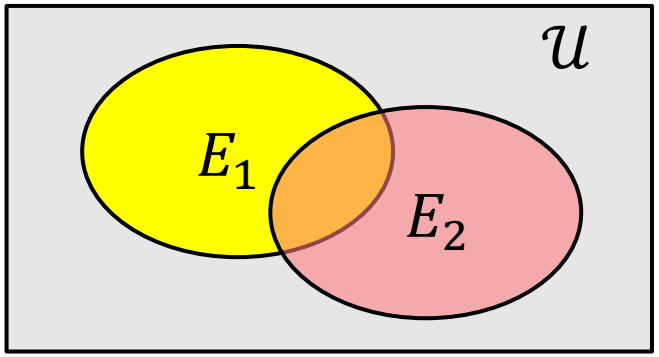
$$= 1/4$$

# Discrete Probability - Complement

- If $E$ is an event, the **complement** of $E$ is $\mathcal{U} \setminus E$ and denoted $\bar{E}$; i.e., $\bar{E}$ is the event that $E$ does *not* occur
  - Fact: $\Pr[\bar{E}] = 1 - \Pr[E]$
- Often, it's easier to compute $\Pr[E] = 1 - \Pr[\bar{E}]$
- Again, suppose we flip 2 coins, then $\mathcal{U} = \{hh, ht, th, tt\}$
  - Suppose $E$ = "at least one tail" = $\{ht, th, tt\}$
  - Complement of $E$ is "no tails" = $\{hh\}$
- Then,
  - $\Pr[E] = 1 - \Pr[\bar{E}] = 1 - 1/4 = 3/4$
- We make use of this trick often!

# Disjunction and Union Bound

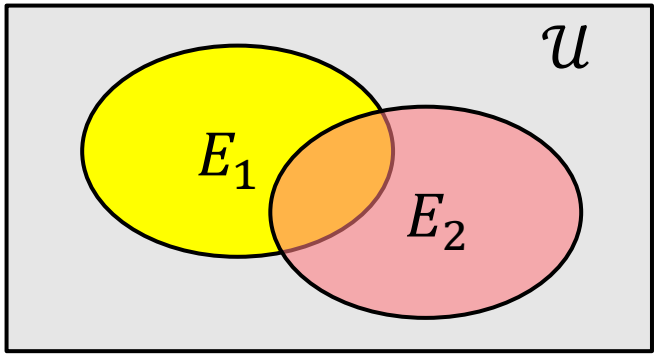
- If $E_1$ and $E_2$ are events, then $E_1 \cup E_2$ denotes the **disjunction** of $E_1$ and $E_2$; that is, $E_1 \cup E_2$ is the event that *either* $E_1$ *or* $E_2$ occurs
  - By definition, $\Pr[E_1 \cup E_2] \geq \Pr[E_1]$ and $\Pr[E_1 \cup E_2] \geq \Pr[E_2]$




- **Union bound**: For events $E_1$ and $E_2$ in $\mathcal{U}$:
  - $\Pr[E_1 \cup E_2] \leq \Pr[E_1] + \Pr[E_2]$
  - Repeated application of the union bound for any events $E_1$, …, $E_k$ gives $\Pr\left[\cup_{i=1}^{k} E_i\right] \leq \sum_{i=1}^{k} \Pr[E_i]$

# Conjunction and Independence

- If $E_1$ and $E_2$ are events, then $E_1 \cap E_2$ denotes their **conjunction**; i.e., $E_1 \cap E_2$ is the event that *both* $E_1$ *and* $E_2$ occur
  - By definition, $\Pr[E_1 \cap E_2] \leq \Pr[E_1]$ and $\Pr[E_1 \cap E_2] \leq \Pr[E_2]$

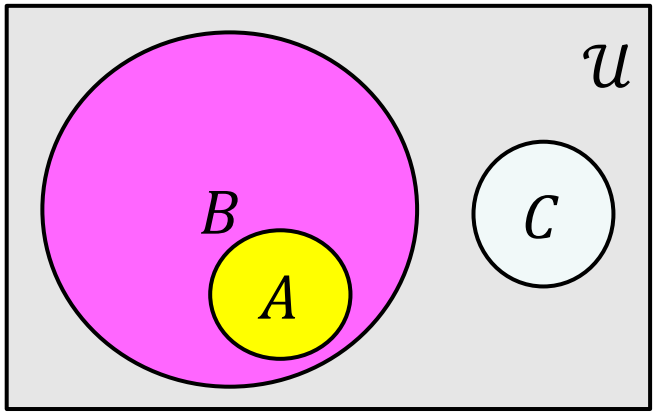



- Events $E_1$ and $E_2$ are said to be **independent** if
  - $\Pr[E_1 \cap E_2] = \Pr[E_1] \cdot \Pr[E_2]$

# Conditional Probability

- The **conditional probability** of $E_1$ given $E_2$, denoted $\Pr[E_1 | E_2]$, represents the probability that event $E_1$ occurs, given that event $E_2$ has occurred, is defined as
  - $\Pr[E_1 | E_2] \stackrel{\text{def}}{=} \frac{\Pr[E_1 \cap E_2]}{\Pr[E_2]}$

    as long as $\Pr[E_2] \neq 0$ (If $\Pr[E_2] = 0$ then $\Pr[E_1 | E_2]$ is undefined)




$\Pr[A \mid B] > \Pr[A]$

$\Pr[A \mid C] = 0$

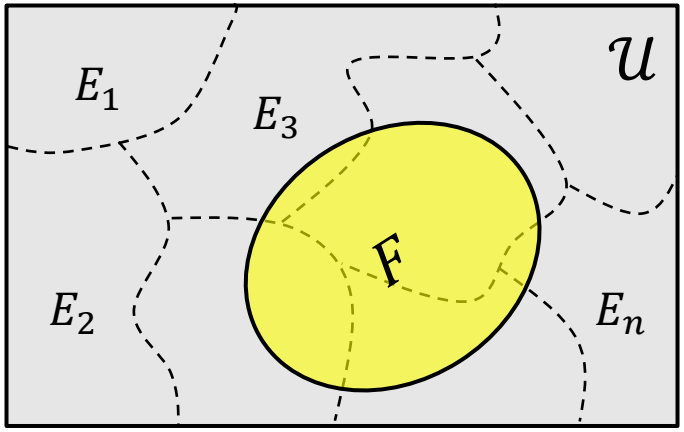
- It follows immediately from the definition that
  - $\Pr[E_1 \cap E_2] = \Pr[E_1 | E_2] \cdot \Pr[E_2]$
  - $\Pr[E_2 \cap E_1] = \Pr[E_2 | E_1] \cdot \Pr[E_1]$
  - But, $\Pr[E_1 \cap E_2] = \Pr[E_2 \cap E_1]$ !!

# Law of Total Probability

- **Bayes' Theorem**:
  - $\Pr[E_1 | E_2] = \frac{\Pr[E_1 \cap E_2]}{\Pr[E_2]} = \frac{\Pr[E_2 \cap E_1]}{\Pr[E_2]} = \frac{\Pr[E_2 | E_1] \cdot \Pr[E_1]}{\Pr[E_2]}$

- Let $E_1, \ldots, E_n$ be disjoint events, so that $\Pr[E_i \cap E_j] = 0$ for all $i \neq j$. That is, at most one of the $\{E_i\}$ occur. Assume further that $\Pr[E_i] > 0$ for all $i$. Then for any event $F$
  - $\Pr[F] = \Pr[F \mid E_1] \cdot \Pr[E_1] + \Pr[F \mid E_2] \cdot \Pr[E_2] + \ldots + \Pr[F \mid E_n] \cdot \Pr[E_n]$

$$= \sum_{i=1}^{n} \Pr[F \mid E_i] \cdot \Pr[E_i]$$



# Exercise - Probability Distribution

Consider the **shift cipher**, with the following distribution over $\mathcal{M}$:

$$\Pr[M = \text{"kim"}] = 0.5,$$

$$\Pr[M = \text{"ann"}] = 0.2, \text{ and}$$

$$\Pr[M = \text{"boo"}] = 0.3$$

1) What is the probability that the ciphertext is "DQQ"?
2) What is the probability that "ann" was encrypted, given that we observe ciphertext "DQQ"?

| Plaintext | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Position | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

# Exercise - Probability Distribution

**Solution:**

1) The only way the ciphertext "DQQ" can occur is if $M = \text{"ann"}$ and $K = 3$, or $M = \text{"boo"}$ and $K = 2$. By independence of $M$ and $K$, we have

$$\Pr[M = \text{"ann"} \cap K = 3] = \Pr[M = \text{"ann"}] \cdot \Pr[K = 3]$$
$$= 0.2 \cdot (1/26)$$

Similarly,

$$\Pr[M = \text{"boo"} \cap K = 2] = \Pr[M = \text{"boo"}] \cdot \Pr[K = 2]$$
$$= 0.3 \cdot (1/26)$$

Therefore,

$$\Pr[C = \text{"DQQ"}] = \Pr[M = \text{"ann"} \cap K = 3] + \Pr[M = \text{"boo"} \cap K = 2]$$
$$= 0.2 \cdot (1/26) + 0.3 \cdot (1/26) = 0.5 \cdot (1/26) = 1/52$$

# Exercise - Probability Distribution

**Solution:**

2) Using Bayes' Theorem, we have

$$\Pr[M = \text{"ann"} \mid C = \text{"DQQ"}]$$
$$= \frac{\Pr[C = \text{"DQQ"} \mid M = \text{"ann"}] \cdot \Pr[M = \text{"ann"}]}{\Pr[C = \text{"DQQ"}]}$$
$$= \frac{\Pr[C = \text{"DQQ"} \mid M = \text{"ann"}] \cdot 0.2}{1/52}$$

Note that, $\Pr[C = \text{"DQQ"} \mid M = \text{"ann"}] = 1/26$, since if $M = \text{"ann"}$ then the only way $C = \text{"DQQ"}$ can occur is if $K = 3$ (which occurs with probability $1/26$). We conclude that

$$\Pr[M = \text{"ann"} \mid C = \text{"DQQ"}] = \frac{(1/26) \cdot 0.2}{1/52} = 0.4$$

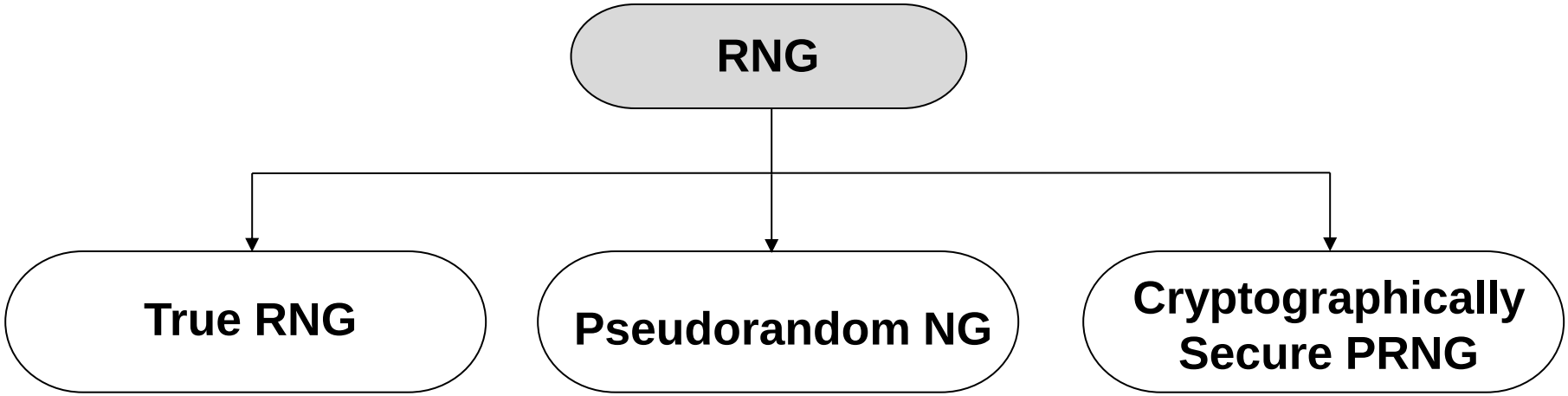
# Random Numbers

- Random numbers used to generate keys
  - E.g., *independent*, *unbiased* (i.e., *uniform*) bits for symmetric keys
- Random numbers used for nonces (used only-once values)
  - Sometimes a sequence is OK
  - But sometimes nonces must be random
- Random numbers also used in simulations, statistics, etc.
  - Such numbers need to be “statistically” random
- Two distinct and not necessarily compatible requirements for a sequence of random numbers are:
  - Randomness (irreproducible)
  - Unpredictability

# Random Numbers

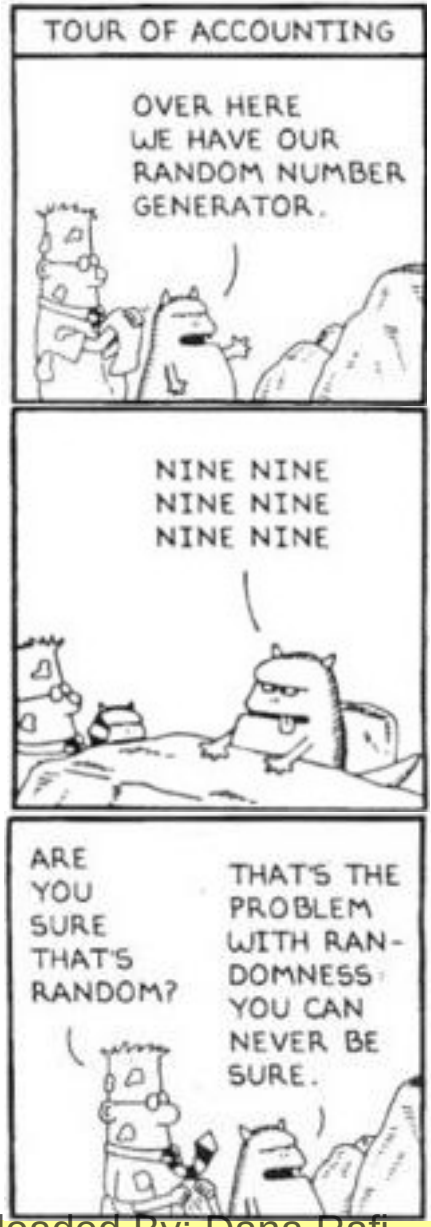
- ***Cryptographic*** random numbers must be **statistically random** and **unpredictable**
- Suppose server generates symmetric keys …
  - Alice: $K_A$
  - Bob: $K_B$
  - Charlie: $K_C$
  - Dave: $K_D$
- But Alice, Bob, and Charlie don't like Dave
- Alice, Bob, and Charlie working together must not be able to determine $K_D$

# Random Number Generators (RNGs)



# True Random Number Generators (TRNGs)

- Based on physical random processes:
  - Delays between network events, hard-disk access times, keystrokes or mouse movements made by the users, thermal/shot noise, or radioactive decay
- However, high-entropy data is not necessarily uniform. Thus, post-processing is needed to obtain (nearly) independent and uniform key stream $k_i$:
  - $Pr(k_i = 0) = Pr(k_i = 1) = 50\%$
- **Entropy** is a measure of unpredictability
- Output can neither be **predicted** nor be **reproduced**
- True "randomness" hard to define



# Exercise - TRNGs Post-Processing

Imagine that a processor generates high-entropy data containing a sequence of *biased* bits, where 1 occurs with probability $p$ and 0 occurs with probability $(1 - p)$. Thousands of such bits have lots of entropy, but are not close to uniform. How can we obtain a uniformly distributed output from the initial high-entropy pool?

**Solution:**

We can obtain a uniform sequence of bits by taking the original bits in pairs: if we see a 1 followed by a 0 then we output 0, and if we see a 0 followed by a 1 then we output 1. (If we see two 0s or two 1s in a row we output nothing, and simply move on to the next pair.) The probability that any pair results in a 0 is $p \cdot (1 - p)$, which is exactly equal to the probability that any pair results in a 1. (Note that we do not even need to know the value of $p$ !)

# Pseudorandom Number Generator (PRNG)

- Generate sequences from initial seed value
- Typically, output stream has good statistical properties
- However, output can be **predicted** and can be **reproduced**
- Often computed in a recursive way:

$$s_0 = seed$$
$$s_{i+1} = F(s_i, s_{i-1}, s_{i-2}, \ldots, s_{i-t})$$

- Example, *rand()* function in the standard C library **<stdlib.h>**:

$$s_0 = 12345$$
$$s_{i+1} = (1103515245 * s_i + 12345) \bmod 2^{31}$$

- Most PRNGs have bad cryptographic properties!
  - Using them in cryptographic settings can have disastrous consequences

# Cryptanalyzing a Simple PRNG

- **Assume:**
  - Unknown $A$, $B$ and $s_0$ as key
  - Size of $A$, $B$ and $s_i$ to be 100 bit

Simple PRNG:

**Linear Congruential Generator**

$s_0 = seed$

$s_{i+1} = (A * s_i + B) \bmod m$

- **Solving**:

  Request 300 bit of output, i.e., $s_1$, $s_2$ and $s_3$

  $s_2 = (A * s_1 + B) \bmod m$

  $s_3 = (A * s_2 + B) \bmod m$

  … directly reveals $A$ and $B$. All $s_i$ can be computed easily!

- **Bad** cryptographic properties due to the linearity of most PRNGs
  - Bottom line: “The use of pseudo-random processes to generate secret quantities can result in pseudo-security”

# Exercise

- Suppose a server generates 3-bit symmetric keys ($m = 2^3 = 8$) using the ***Linear Congruential Generator***. The assigned keys for Alice, Bob, Charlie, and Dave are as follows:
  - Alice: $K_A = s_i = 7$
  - Bob: $K_B = s_{i+1} = 4$
  - Charlie: $K_C = s_{i+2} = 5$
  - Dave: $K_D = s_{i+3} = 2$

a) Can Alice determine the keys for Bob, Charlie, and Dave?

b) If Alice and Bob work together, will they be able to determine Charlie and Dave's keys?

c) If Alice, Bob, and Charlie work together, will they be able to determine Dave's key?

# Cryptographically Secure PRNG (CSPRNG)

- Special PRNG with additional property:
  - Output must be unpredictable
- More precisely: Given $n$ consecutive bits of output $s_i$, the following output bits $s_{n+1}$ *cannot* be predicted (in polynomial time)
- Needed in cryptography, in particular for stream ciphers
- Remark: There are almost no other applications that need unpredictability, whereas many, many (technical) systems need PRNGs

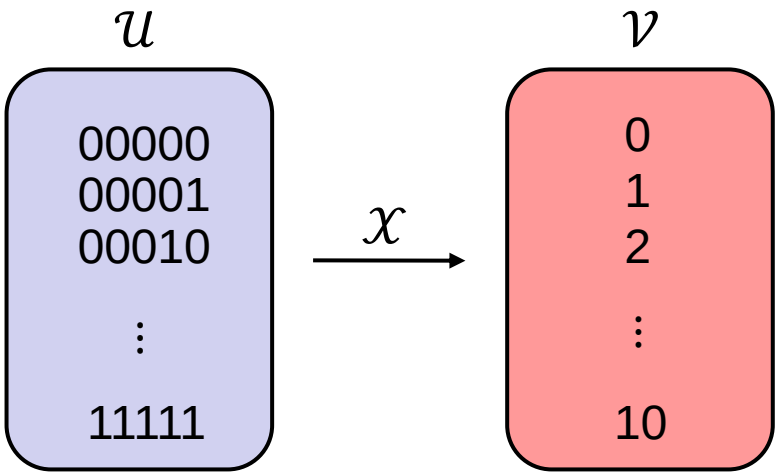
# Random Variables

- A **random variable** X is a function $\mathcal{X} : \mathcal{U} \to \mathcal{V}$

- Example:

$$\mathcal{X} : \{0,1\}^5 \to \{0,1,2,\ldots,10\}$$

$$\mathcal{X}(x) \stackrel{\text{def}}{=} x_0 + x_1 + x_2 + x_3 + x_4$$



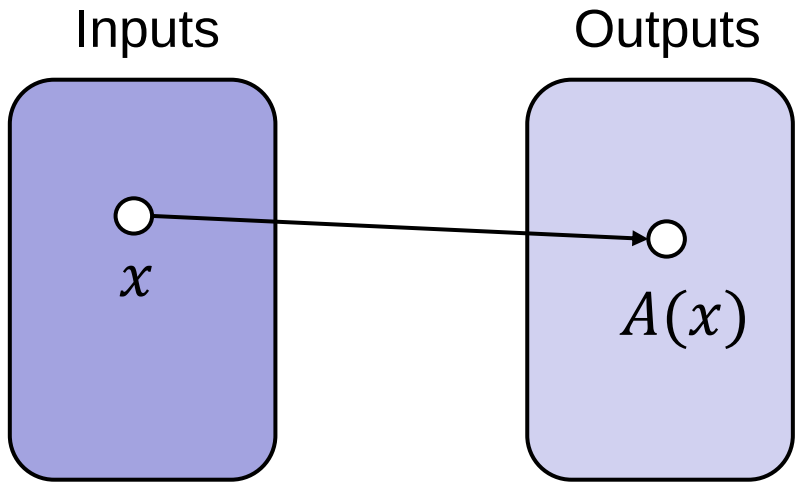

$$\Pr[X = 0] = 1/32$$
$$\Pr[X = 1] = 5/32$$
$$\Pr[X = 2] = \binom{5}{2} / 32 = 10/32$$
$$\vdots$$
$$\Pr[X = 10] = 0$$

Uniform distribution on $\mathcal{U}$ but not on $\mathcal{V}$!

# Deterministic and Randomized Algorithms

- Deterministic algorithm:

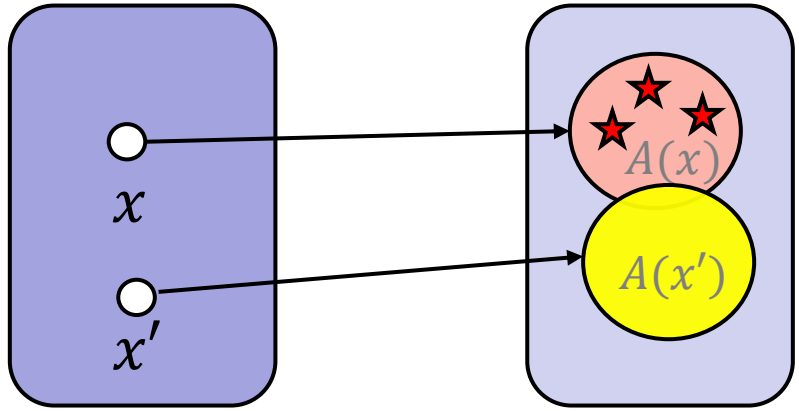
$$y \leftarrow A(x)$$




- Randomized algorithm:

$$y \leftarrow A(x, r) \quad \text{where } r \xleftarrow{\$} \{0,1\}^n$$

$$y \xleftarrow{\$} A(x')$$



# Symmetric Key Cryptography - Review

- An encryption scheme is defined by:
  - The *key-generation algorithm* (Gen): a probabilistic algorithm that outputs a key $k$, $k \in \mathcal{K}$, chosen according to some distribution. $\mathcal{K}$ is the set of all possible keys that can be output by Gen
  - The *encryption algorithm* (Enc): encrypt message $m$, $m \in \mathcal{M}$, using the key $k$

    $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ , $\mathsf{Enc}(k, m) = \mathsf{Enc}_k(m) = c$

    where, $\mathcal{C}$ denote the set of all possible ciphertexts that can be output by $\mathsf{Enc}_k(m)$
  - The *decryption algorithm* (Dec): decrypt ciphertext $c$, $c \in \mathcal{C}$, using the key $k$

    $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$, $\mathsf{Dec}(k, c) = \mathsf{Dec}_k(c) = m$

# Presentation Outline

- Basic Probability
- **Perfect Secrecy**
- The One-Time Pad
- Crypto Requirements
- Crypto Taxonomy

# Perfectly Secret

**Definition:** An encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ is *perfectly secret* if for every probability distribution for $M$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $\Pr[C = c] > 0$:

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

- For an encryption scheme to be **perfectly secret**, the ciphertext should have *no effect* on the adversary's knowledge regarding the actual plaintext that was sent
  - In other words, the ciphertext reveals nothing about the underlying plaintext

# Exercise - Perfectly Secret

Show that the shift cipher is *not* perfectly secret when used with the message space $\mathcal{M}$ consisting of all two-letter plaintexts

**Solution:**

The probability distribution over $\mathcal{M}$, for every message $m \in \mathcal{M}$, is $\Pr[M = m] = 1/(26)^2$

Consider the message is "hi" and the ciphertext is "XX" (i.e., $m =$ "hi" and $c =$ "XX"). Then, clearly $\Pr[M = \text{"hi"} \mid C = \text{"XX"}] = 0$, as there is no way that "XX" can ever result from the encryption of "hi" (In the shift cipher, the relative shift between characters is preserved). Therefore,

$$\Pr[M = \text{"hi"} \mid C = \text{"XX"}] \neq \Pr[M = \text{"hi"}]$$

$$0 \neq 1/(26)^2$$

*# the scheme is not perfectly secret*

# Perfectly Secret

**LEMMA:** An encryption scheme Σ = (Gen, Enc, Dec) with message space $\mathcal{M}$ is *perfectly secret* if and only if

$$\Pr[\mathrm{Enc}_K(m) = c] = \Pr[\mathrm{Enc}_K(m') = c]$$

for every two messages $m, m' \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$, with probability taken over the random choice $K \xleftarrow{\$} \mathcal{K}$ and the random coins used by $\mathrm{Enc}()$ (if any)

- For an encryption scheme to be **perfectly secret**, the distribution of the ciphertext *must not* depend on the plaintext
  - In other words, the distribution of the ciphertext when $m$ is encrypted should be identical to the distribution of the ciphertext when $m'$ is encrypted

# Exercise

Given the following encryption scheme, where $\mathbf{Enc}_k(m)$ returns **[$m + k$ mod 3]**, and $\mathbf{Dec}_k(c)$ returns **[$c - k$ mod 3]**, under which of the below message spaces and key spaces the encryption scheme is *perfectly secret*?

Scheme 1: The message space is $\mathcal{M} = \{0, 1\}$, and **Gen** chooses a uniform key from the key space $\mathcal{K} = \{0, 1\}$

Scheme 2: The message space is $\mathcal{M} = \{0, 1\}$, and **Gen** chooses a uniform key from the key space $\mathcal{K} = \{0, 1, 2\}$

Scheme 3: The message space is $\mathcal{M} = \{0, 1\}$, and **Gen** chooses a uniform key from the key space $\mathcal{K} = \{1, 2\}$

Scheme 4: The message space is $\mathcal{M} = \{0, 1, 2\}$, and **Gen** chooses a uniform key from the key space $\mathcal{K} = \{0, 1, 2\}$

a) Scheme 1
b) Scheme 2
c) Scheme 3
d) Scheme 4
e) None

# Perfect (Adversarial) Indistinguishability

**Definition:** An encryption scheme $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for every adversary $A$ it holds that

$$\Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma, A} = 1\right] = \Pr[b' = b] = \frac{1}{2}$$

- An encryption scheme is **perfectly indistinguishable** if no adversary $A$ can succeed with probability better than 1/2

**LEMMA:** An encryption scheme $\Sigma$ is *perfectly secret* if and only if it is *perfectly indistinguishable*

**$\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma}(A)$**

$\Sigma = (\text{Gen}, \text{Enc}, \text{Dec})$

*A:* an adversary, a stateful algorithm

---

1. $b \xleftarrow{\$} \{0,1\}$
2. $k \leftarrow \Sigma.\text{Gen}()$
3. $m_0 \leftarrow A,\ m_1 \leftarrow A$
4. $c \leftarrow \Sigma.\text{Enc}(k, m_b)$
5. $b' \leftarrow A(c)$
6. **return** $b' \stackrel{?}{=} b$

# Exercise - Perfectly Secret

Show that the Vigenère cipher is *not* perfectly indistinguishable, at least for certain parameters (e.g., for the message space of two-letter strings and the upper bound of the period is 2)

**Solution:**

Let $\Sigma$ denote the Vigenère cipher for the message space of two-letter strings, and where the period is chosen uniformly in $\{1, 2\}$, and adversary $A$ does:

1. Output $m_0 =$ "aa" and $m_1 =$ "ab"
2. Upon receiving the challenge ciphertext $c = c_0 c_1$, do the following: if $c_0 = c_1$ output 0; else output 1

$$\Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma, A} = 1\right]$$
$$= \Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma, A} = 1 \mid b = 0\right] \cdot \Pr[b = 0]$$
$$+ \Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma, A} = 1 \mid b = 1\right] \cdot \Pr[b = 1]$$

# Exercise - Perfectly Secret

$\Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma,A} = 1\right]$

$= \Pr[A \text{ outputs } 0 \mid b = 0] \cdot 1/2 + \Pr[A \text{ outputs } 1 \mid b = 1] \cdot 1/2$

When $b = 0$ (so $m_0 =$ "aa" is encrypted) then $c_0 = c_1$ (i.e., $A$ outputs 0) if either (1) a key of period 1 is chosen, or (2) a key of period 2 is chosen and both characters of the key are equal

$$\Pr[A \text{ outputs } 0 \mid b = 0] = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{26} \cdot 1 = \frac{27}{52}$$

When $b = 1$ (so $m_1 =$ "ab" is encrypted) then $c_0 = c_1$ (i.e., $A$ outputs 0) only if a key of period 2 is chosen and the first character of the key is one more than the second character of the key

$$\Pr[A \text{ outputs } 1 \mid b = 1] = 1 - \Pr[A \text{ outputs } 0 \mid b = 1] = 1 - \frac{1}{2} \cdot \frac{1}{26} \cdot 1 = \frac{51}{52}$$

Plugging into the main Equation, then gives

$$\Pr\left[\mathbf{Exp}^{\text{Per-Indist}}_{\Sigma,A} = 1\right] = \frac{1}{2} \cdot \left(\frac{27}{52} + \frac{51}{52}\right) = 0.75 > \frac{1}{2}$$

*# the scheme is not perfectly indistinguishable*

# Presentation Outline

- Basic Probability
- Perfect Secrecy
- **The One-Time Pad**
- Crypto Requirements
- Crypto Taxonomy

# One-Time Pad

- A **perfectly secret** encryption scheme proposed in 1917
- Encryption: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

  $\mathcal{C} \leftarrow M \oplus K$

  - $\text{Enc}_k(m_1 \cdots m_L) = c_1 \cdots c_L$ , where $c_i = m_i \oplus k_i$

- Decryption: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

  $M \leftarrow \mathcal{C} \oplus K$

  - $\text{Dec}_k(c_1 \cdots c_L) = m_1 \cdots m_L$ , where $m_i = c_i \oplus k_i$

Where, $\oplus$ denote the *bitwise exclusive-or* (XOR) operation, and $\text{Dec}_k(\text{Enc}_k(m)) = k \oplus k \oplus m = m$

$$\mathcal{K} = \{0,1\}^n \qquad \mathcal{M} = \{0,1\}^n \qquad \mathcal{C} = \{0,1\}^n$$

# One-Time Pad: Encryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

❖ **Encryption: Plaintext ⊕ Key = Ciphertext**

| | h | e | i | l | h | i | t | l | e | r |
|---|---|---|---|---|---|---|---|---|---|---|
| Plaintext: | 001 | 000 | 010 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| Key: | 111 | 101 | 110 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| Ciphertext: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| | s | r | l | h | s | s | t | h | s | r |

# One-Time Pad: Decryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

❖ **Decryption: Ciphertext ⊕ Key = Plaintext**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | s | r | l | h | s | s | t | h | s | r |
| Ciphertext: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| Key: | 111 | 101 | 110 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| Plaintext: | 001 | 000 | 010 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| | h | e | i | l | h | i | t | l | e | r |

# One-Time Pad: Decryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

- Double agent claims sender used following "**key**"

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | s | r | l | h | s | s | t | h | s | r |
| Ciphertext: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| **Key**: | 101 | 111 | 000 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| Plaintext: | 011 | 010 | 100 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| | k | i | l | l | h | i | t | l | e | r |

# One-Time Pad: Decryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

- Or sender is captured and claims the key is…

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | s | r | l | h | s | s | t | h | s | r |
| Ciphertext: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| Key: | 111 | 101 | 000 | 011 | 101 | 110 | 001 | 011 | 101 | 101 |
| Plaintext: | 001 | 000 | 100 | 010 | 011 | 000 | 110 | 010 | 011 | 000 |
| | h | e | l | i | k | e | s | i | k | e |

# Exercise – OTP Encryption

What is the ciphertext that results when the plaintext **0x012345** (written in *hex*) is encrypted using the one-time pad with key **0xFFEEDD**?

**Solution:**

Encryption: Plaintext ⊕ Key = Ciphertext

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Plaintext: | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 |
| Key: | 1111 | 1111 | 1110 | 1110 | 1101 | 1101 |
| Ciphertext: | 1111 | 1110 | 1100 | 1101 | 1001 | 1000 |
| | F | E | C | D | 9 | 8 |

# One-Time Pad – Security

**Theorem:** The one-time pad encryption scheme has **one-time** *perfect privacy*

- From adversary's POV, the ciphertext is *uniformly* distributed over $\mathcal{C}$ ($C$ *cannot* give any information about $M$)

| Prob | $K$ | $C = K \oplus 101$ |
|---|---|---|
| 1/8 | 000 | 101 |
| 1/8 | 001 | 100 |
| 1/8 | 010 | 111 |
| 1/8 | 011 | 110 |
| 1/8 | 100 | 001 |
| 1/8 | 101 | 000 |
| 1/8 | 110 | 011 |
| 1/8 | 111 | 010 |

| Prob | $K$ | $C = K \oplus 001$ |
|---|---|---|
| 1/8 | 000 | 001 |
| 1/8 | 001 | 000 |
| 1/8 | 010 | 011 |
| 1/8 | 011 | 010 |
| 1/8 | 100 | 101 |
| 1/8 | 101 | 100 |
| 1/8 | 110 | 111 |
| 1/8 | 111 | 110 |

# Proof of OTP One-Time Perfect Privacy

**Theorem:** The one-time pad encryption scheme has **one-time** *perfect privacy*

**Need to show:** $\Pr[M = m \mid C = c] = \Pr[M = m]$, where $m, c \in \{0,1\}^n$

$$\Pr[M = m \mid C = c] = \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]}$$ (Bayes' Theorem)

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

$$= \Pr[K \oplus m = c] = \Pr[K = m \oplus c] = \frac{1}{2^n}$$

$$\Pr[C = c] = \sum_{m \in M} \Pr[C = c \mid M = m] \cdot \Pr[M = m]$$ (Law of Total Probability)

$$= \sum_{m \in M} \frac{1}{2^n} \cdot \Pr[M = m] = \frac{1}{2^n} \cdot \sum_{m \in M} \Pr[M = m] = \frac{1}{2^n}$$

$$\Pr[M = m \mid C = c] = \frac{1/2^n \cdot \Pr[M = m]}{1/2^n} = \Pr[M = m]$$

# Proof of OTP One-Time Perfect Privacy

**Theorem:** The one-time pad encryption scheme has **one-time** *perfect privacy*

**Need to show:** $\Pr[\mathrm{Enc}_K(m) = c] = \Pr[\mathrm{Enc}_K(m') = c]$ for any two messages $m, m' \in \mathcal{M}$, and any ciphertext $c \in \mathcal{C}$, where $m, m', c \in \{0,1\}^n$

$$\Pr[\mathrm{Enc}_K(m) = c] = \Pr[K \oplus m = c]$$

$$= \Pr[K = m \oplus c] = \Pr[K = k_1] = \frac{1}{2^n}$$

$$\Pr[\mathrm{Enc}_K(m') = c] = \Pr[K \oplus m' = c]$$

$$= \Pr[K = m' \oplus c] = \Pr[K = k_2] = \frac{1}{2^n}$$

We conclude that the one-time pad is perfectly secret.

# Exercise - Perfectly Secret

Prove or refute: An encryption scheme with message space $\mathcal{M}$ is perfectly secret if and only if for every probability distribution on $\mathcal{M}$ and every $c_0, c_1 \in \mathcal{C}$, we have

$$\Pr[C = c_0] = \Pr[C = c_1]$$

**Solution:**

This is not true. Consider modifying the one-time pad so encryption appends a bit that is 0 with probability 1/4 and 1 with probability 3/4. This scheme will still be perfect secret, but ciphertexts ending with 1 are more likely that ciphertexts ending with 0.

# One-Time Pad – Perfect?

- **Provably** secure…
  - Ciphertext provides **no** info about plaintext
  - All plaintexts are equally likely
- OTP has perfect privacy … for ***one*** *message*
  - What happens if you use the same (unknown) key for two messages?
  - $c_1 \oplus c_2 = (k \oplus m_1) \oplus (k \oplus m_2) = m_1 \oplus m_2$
  - The adversary can learn where the two messages differ
- Key is as long as the message
  - Key management becomes very difficult
  - What happens if it is shorter?

**Theorem:** No encryption scheme can have perfect secrecy if $|\mathcal{K}| < |\mathcal{M}|$

- Nothing special about XOR: ROT-K also has one-time perfect privacy
  - Why doesn't this contradict what we saw earlier about ROT-K?

# Exercise – OTP Key Limitation

The following questions concern multiple encryptions of single-character ASCII plaintexts with the **one-time pad** using the same ***8-bit key***. You may assume that the plaintexts are either (upper- or lower-case) *English letters* or the *space character*.

a) Say you see the ciphertexts **1011 0111** and **1110 0111**. What can you deduce about the plaintext characters these correspond to?

b) Say you see the three ciphertexts **0110 0110**, **0011 0010**, and **0010 0011**. What can you deduce about the plaintext characters these correspond to?

```
Bit 4:     0000000000000000 1111111111111111 |
Bits 0-3:  0123456789ABCDEF 0123456789ABCDEF | Block:
---------+----------------------------------+--------------------------
Bits  00 | ................ ................ | Control characters
5-6:  01 |  !"#$%&'()*+,-./ 0123456789:;<=>? | Numbers and punctuation
      10 | @ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_ | Uppercase letters (mostly)
      11 | `abcdefghijklmno pqrstuvwxyz{|}~. | Lowercase letters (mostly)
```

# Presentation Outline

- Basic Probability
- Perfect Secrecy
- The One-Time Pad
- **Crypto Requirements**
- Crypto Taxonomy

# Cryptography Requirements

- Wanted: security definition for symmetric encryption
  - One-time perfect privacy: $\Pr[b' = b] = \frac{1}{2}$
  - Security holds for *any* adversary (no limit on resource usage)
  - Very strict requirements:
    - Keys need to be as long as message
    - Key can only be used for one message
- Modern cryptography – idea
  - Computational privacy: $\Pr[b' = b] = \frac{1}{2} \pm \varepsilon$
  - Security holds for *any* recourse bounded adversary
  - Very strict requirements:
    - Want keys to be short
    - Want to encrypt many messages using the same key

# Claude Shannon

- The founder of Information Theory
- 1949 paper: *Comm. Thy. of Secrecy Systems*
- Fundamental concepts
  - **Diffusion** — The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
    - This is achieved by having each plaintext digit affect the value of many ciphertext
  - **Confusion** — Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
    - Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key
- Proved one-time pad is secure
- One-time pad is confusion-only

# Presentation Outline

- Basic Probability
- Perfect Secrecy
- The One-Time Pad
- Crypto Requirements
- **Crypto Taxonomy**

# Taxonomy of Cryptography

- **Symmetric Key**
  - Same key for encryption and decryption
  - Two types: Stream ciphers and Block ciphers
- **Public Key** (or asymmetric crypto)
  - Two keys, one for encryption (public), and one for decryption (private)
  - And digital signatures — nothing comparable in symmetric key crypto
- **Hash Algorithms**
  - Can be viewed as “one way” crypto

# Outline of Course

| | Message Privacy | Message Integrity / Authentication |
|---|---|---|
| **Symmetric Keys** | Symmetric Encryption (private-key encryption) | Message Authentication Codes (MAC) |
| **Asymmetric Keys** | Asymmetric Encryption (public-key encryption) | Digital Signatures |

Part I

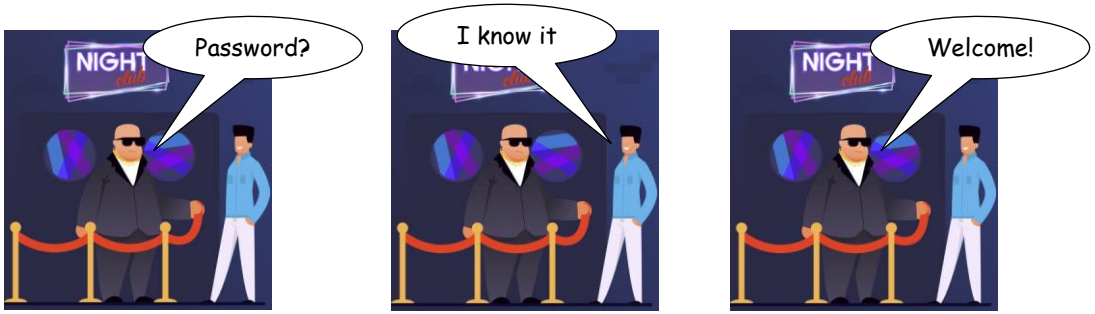
# Outline of Course

| | Message Privacy | Message Integrity / Authentication |
|---|---|---|
| **Symmetric Keys** | Symmetric Encryption (private-key encryption) | Message Authentication Codes (MAC) |
| **Asymmetric Keys** | Asymmetric Encryption (public-key encryption) | Digital Signatures |

Part II

# Much More to Cryptography

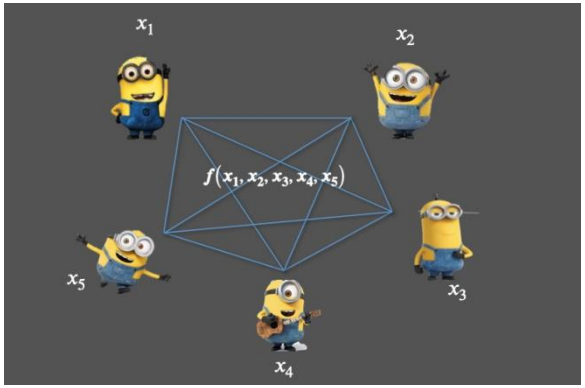
- Zero-knowledge proofs



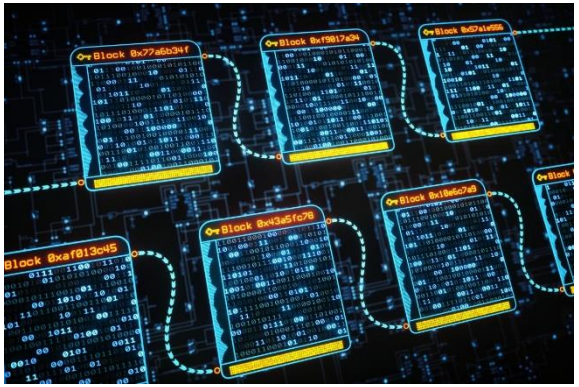

- Fully-homomorphic encryption

$$Enc(K, M_1 + M_2) = Enc(K, M_1) + Enc(K, M_2)$$

- Multi-party computation




- Blockchain



# Exercise – OTP Key Limitation

The following question concerns multiple encryptions of single-character ASCII plaintexts with the ***one-time pad*** using the same 8-bit key. You may assume that the plaintexts are either (upper-case or lower-case) English letters or digit characters (0, 1, …, 9). Say you see the three ciphertexts $(\mathbf{EF})_{16}$, $(\mathbf{A4})_{16}$, and $(\mathbf{D3})_{16}$. What can you deduce about the plaintext characters these correspond to?

| Bit 4: | 0000000000000000 1111111111111111 | |
|---|---|---|
| Bits 0-3: | 0123456789ABCDEF 0123456789ABCDEF | Block: |
| Bits 5-6: 00 | ..... ..... | Control characters |
| 01 | !"#$%&'()*+,-./ 0123456789:;<=>? | Numbers and punctuation |
| 10 | @ABCDEFGHIJKLMNO PQRSTUVWXYZ[\]^_ | Uppercase letters (mostly) |
| 11 | `abcdefghijklmno pqrstuvwxyz{\|}~. | Lowercase letters (mostly) |

# Slides Original Source

- Jonathan Katz and Yehuda Lindell, “Introduction to Modern Cryptography,” Third Edition, 2021
- M. Stamp, “Information Security: Principles and Practice,” John Wiley
- B. Forouzan, “Cryptography and Network Security,” McGraw-Hill