


Information Security

CS 526

Topic 3



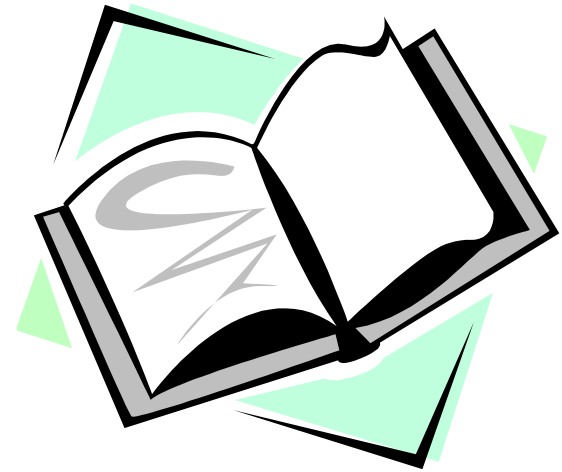
Cryptography: One-time Pad, Information
Theoretic Security, and Stream Ciphers

Announcements

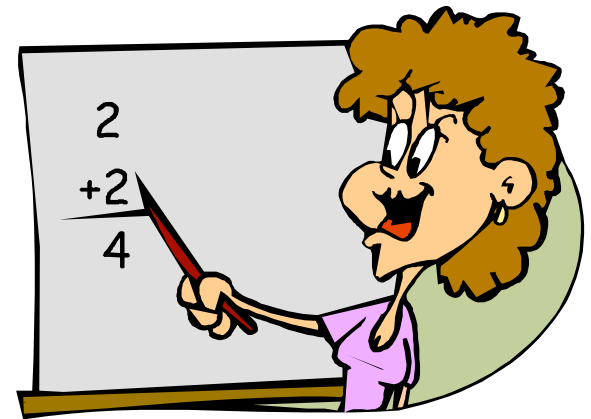
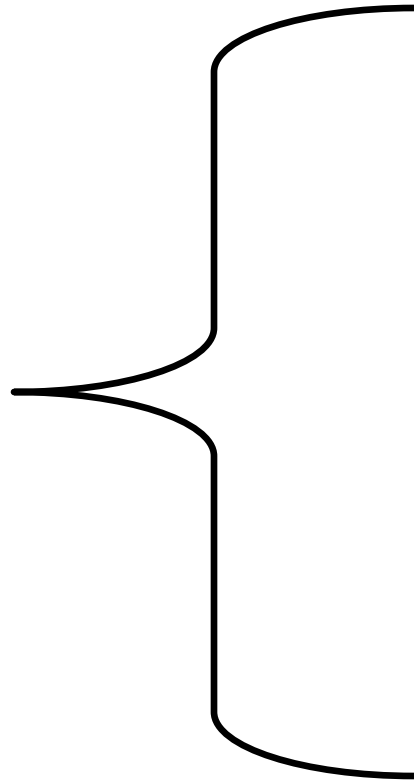
- HW1 is out, due on Sept 11
 - Start early, late policy is 3 total late days
- HW2 will be assigned on Sept 11, due on Sept 25
- Planned project 1: implementation of file encryption, to be assigned on Sept 25

Readings for This Lecture

- Required reading from wikipedia
 - [One-Time Pad](#)
 - [Information theoretic security](#)
 - [Stream cipher](#)
 - [Pseudorandom number generator](#)



Begin Math



Random Variable

Definition

A **discrete random variable**, \mathbf{X} , consists of a finite set \mathcal{X} , and a probability distribution defined on \mathcal{X} . The probability that the random variable \mathbf{X} takes on the value x is denoted $\Pr[\mathbf{X} = x]$; sometimes, we will abbreviate this to $\Pr[x]$ if the random variable \mathbf{X} is fixed. It must be that

$$0 \leq \Pr[x] \quad \text{for all } x \in \mathcal{X}$$

$$\sum_{x \in \mathcal{X}} \Pr[x] = 1$$

Example of Random Variables

- Let random variable \mathbf{D}_1 denote the outcome of throwing one die (with numbers 0 to 5 on the 6 sides) randomly, then $\mathcal{D}=\{0,1,2,3,4,5\}$ and $\Pr[\mathbf{D}_1=i] = 1/6$ for $0 \leq i \leq 5$
- Let random variable \mathbf{D}_2 denote the outcome of throwing a second such die randomly
- Let random variable \mathbf{S}_1 denote the sum of the two dice, then $\mathcal{S} = \{0, 1, 2, \dots, 10\}$, and
$$\Pr[\mathbf{S}_1=0] = \Pr[\mathbf{S}_1=10] = 1/36$$
$$\Pr[\mathbf{S}_1=1] = \Pr[\mathbf{S}_1=9] = 2/36 = 1/18$$

...
- Let random variable \mathbf{S}_2 denote the sum of the two dice modulo 6, what is the distribution of \mathbf{S}_2 ?

Relationships between Two Random Variables

Definitions

Assume \mathbf{X} and \mathbf{Y} are two random variables, then we define:

- **joint probability**: $\Pr[x, y]$ is the probability that \mathbf{X} takes value x and \mathbf{Y} takes value y .
- **conditional probability**: $\Pr[x|y]$ is the probability that \mathbf{X} takes value x given that \mathbf{Y} takes value y .
$$\Pr[x|y] = \Pr[x, y] / \Pr[y]$$
- **independent random variables**: \mathbf{X} and \mathbf{Y} are said to be independent if $\Pr[x, y] = \Pr[x]P[y]$, for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$.

Examples

- Joint probability of \mathbf{D}_1 and \mathbf{D}_2
for $0 \leq i, j \leq 5$, $\Pr[\mathbf{D}_1=i, \mathbf{D}_2=j] = ?$
- Are \mathbf{D}_1 and \mathbf{D}_2 independent?
- Suppose \mathbf{D}_1 is plaintext and \mathbf{D}_2 is key, and \mathbf{S}_1 and \mathbf{S}_2 are ciphertexts of two different ciphers, which cipher would you use?

Examples to think after class

- What is the joint probability of \mathbf{D}_1 and \mathbf{S}_1 ?
- What is the joint probability of \mathbf{D}_2 and \mathbf{S}_2 ?

- What is the conditional probability $\Pr[\mathbf{S}_1=s \mid \mathbf{D}_1=i]$ for $0 \leq i \leq 5$ and $0 \leq s \leq 10$?
- What is the conditional probability $\Pr[\mathbf{D}_1=i \mid \mathbf{S}_2=s]$ for $0 \leq i \leq 5$ and $0 \leq s \leq 5$?

- Are \mathbf{D}_1 and \mathbf{S}_1 independent?
- Are \mathbf{D}_1 and \mathbf{S}_2 independent?

Bayes' Theorem

If $P[y] > 0$ then

$$P[x | y] = \frac{P[x]P[y | x]}{P[y]}$$

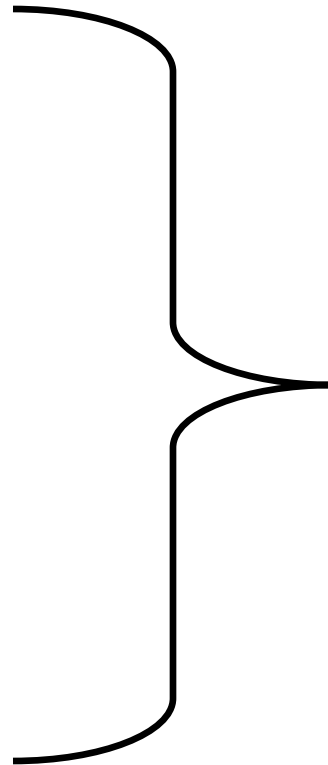
$$P[y] = \sum_{x \in X} P[x, y] = \sum_{x \in X} P[x]p[y | x]$$

Corollary

X and Y are independent random variables
iff $P[x|y] = P[x]$, for all $x \in X$ and all $y \in Y$.

Example: What is $\Pr[\mathbf{D}_1=1 | \mathbf{S}_1=3]$?

End Math



One-Time Pad

- Fix the vulnerability of the Vigenere cipher by using very long keys
- Key is a random string that is at least as long as the plaintext
- Encryption is similar to shift cipher
- Invented by Vernam in the 1920s

One-Time Pad

Let $Z_m = \{0, 1, \dots, m-1\}$ be
the alphabet.



Plaintext space = Ciphertext space = Key space =
 $(Z_m)^n$

The key is chosen uniformly randomly

Plaintext $X = (x_1 \ x_2 \ \dots \ x_n)$

Key $K = (k_1 \ k_2 \ \dots \ k_n)$

Ciphertext $Y = (y_1 \ y_2 \ \dots \ y_n)$

$e_k(X) = (x_1+k_1 \ x_2+k_2 \ \dots \ x_n+k_n) \bmod m$

$d_k(Y) = (y_1-k_1 \ y_2-k_2 \ \dots \ y_n-k_n) \bmod m$

The Binary Version of One-Time Pad

Plaintext space = Ciphertext space =

Keyspace = $\{0,1\}^n$

Key is chosen randomly

For example:

- Plaintext is 11011011
- Key is 01101001
- Then ciphertext is 10110010

Bit Operators

- Bit AND

$$0 \wedge 0 = 0 \quad 0 \wedge 1 = 0 \quad 1 \wedge 0 = 0 \quad 1 \wedge 1 = 1$$

- Bit OR

$$0 \vee 0 = 0 \quad 0 \vee 1 = 1 \quad 1 \vee 0 = 1 \quad 1 \vee 1 = 1$$

- Addition mod 2 (also known as Bit XOR)

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

- Can we use operators other than Bit XOR for binary version of One-Time Pad?

How Good is One-Time Pad?

- Intuitively, it is secure ...
 - The key is random, so the ciphertext is completely random
- How to formalize the confidentiality requirement?
 - Want to say “certain thing” is not learnable by the adversary (who sees the ciphertext). But what is the “certain thing”?
- Which (if any) of the following is the correct answer?
 - The key.
 - The plaintext.
 - Any bit of the plaintext.
 - Any information about the plaintext.
 - E.g., the first bit is 1, the parity is 0, or that the plaintext is not “aaaa”, and so on

Shannon (Information-Theoretic)

Security = Perfect Secrecy

- Basic Idea: Ciphertext should reveal no “information” about Plaintext

Definition. An encryption over a message space \mathcal{M} is perfectly secure if

\forall probability distribution over \mathcal{M}

\forall message $m \in \mathcal{M}$

\forall ciphertext $c \in \mathcal{C}$ for which $\Pr[C=c] > 0$

We have

$$\Pr [\mathbf{PT}=m \mid \mathbf{CT}=c] = \Pr [\mathbf{PT} = m].$$

Explanation of the Definition

- $\Pr [\mathbf{PT} = m]$ is what the adversary believes the probability that the plaintext is m , before seeing the ciphertext
- $\Pr [\mathbf{PT} = m \mid \mathbf{CT}=c]$ is what the adversary believes after seeing that the ciphertext is c
- $\Pr [\mathbf{PT}=m \mid \mathbf{CT}=c] = \Pr [\mathbf{PT} = m]$ means that after knowing that the ciphertext is C_0 , the adversary's belief does not change.

An Equivalent Definition of Perfect Secrecy

Definition. An encryption scheme is perfectly secure if and only if **for any ciphertext c , and any two plaintext m_1 and m_2 , the probability that m_1 is encrypted to c is the same as the probability that m_2 is encrypted to c .**

\forall message m_1, m_2

\forall ciphertext c

$$\Pr [\mathbf{CT}=c \mid \mathbf{PT} = m_1] = \Pr [\mathbf{CT} = c \mid \mathbf{PT} = m_2]$$

Example for Information Theoretical Security

- Consider an example of encrypting the result of a 6-side dice (1 to 6).
 - Method 1: randomly generate $K=[1..6]$, ciphertext is result + K.
 - What is plaintext distribution? After seeing that the ciphertext is 3, what could be the plaintext. After seeing that the ciphertext is 12, what could be the plaintext?
 - Method 2: randomly generate $K=[1..6]$, ciphertext is (result + K) mod 6.
 - Same questions.
 - Can one do a brute-force attack?

Perfect Secrecy

- Fact: When keys are uniformly chosen in a cipher, the cipher has perfect secrecy iff. the number of keys encrypting M to C is the same for any (M,C)
 - This implies that
$$\forall c \forall m_1 \forall m_2 \Pr[\mathbf{CT}=c \mid \mathbf{PT}=m_1] = \Pr[\mathbf{CT}=c \mid \mathbf{PT}=m_2]$$
- One-time pad has perfect secrecy when limited to messages over the same length (**Proof?**)

Key Randomness in One-Time Pad

- One-Time Pad uses a very long key, what if the key is not chosen randomly, instead, texts from, e.g., a book are used as keys.
 - this is not One-Time Pad anymore
 - this does not have perfect secrecy
 - this can be broken
 - How?
- The key in One-Time Pad should never be reused.
 - If it is reused, it is Two-Time Pad, and is insecure!
 - Why?

Usage of One-Time Pad

- To use one-time pad, one must have keys as long as the messages.
- To send messages totaling certain size, sender and receiver must agree on a shared secret key of that size.
 - typically by sending the key over a secure channel
- Key agreement is difficult to do in practice.
- Can't one use the channel for sending the key to send the messages instead?
- Why is OTP still useful, even though difficult to use?

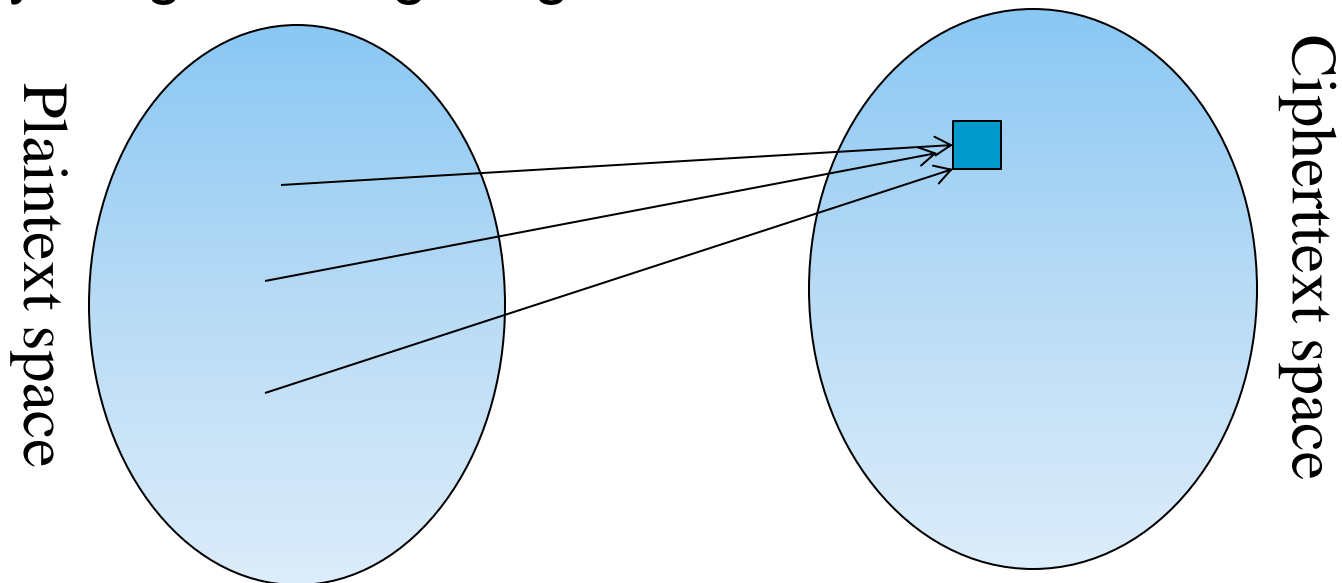
Usage of One-Time Pad

- The channel for distributing keys may exist at a different time from when one has messages to send.
- The channel for distributing keys may have the property that keys can be leaked, but such leakage will be detected
 - Such as in Quantum cryptography

The “Bad News” Theorem for Perfect Secrecy

- Question: OTP requires key as long as messages, is this an inherent requirement for achieving perfect secrecy?
- Answer. Yes. Perfect secrecy implies that $\text{key-length} \geq \text{msg-length}$

Proof:



- Implication: Perfect secrecy difficult to achieve in practice

Stream Ciphers

- In One-Time Pad, a key is a random string of length at least the same as the message
- Stream ciphers:
 - Idea: replace “rand” by “pseudo rand”
 - Use Pseudo Random Number Generator
 - PRNG: $\{0,1\}^s \rightarrow \{0,1\}^n$
 - expand a short (e.g., 128-bit) random seed into a long (e.g., 10^6 bit) string that “looks random”
 - Secret key is the seed
 - Basic encryption method: $E_{\text{key}}[M] = M \oplus \text{PRNG}(\text{key})$

The RC4 Stream Cipher

- A proprietary cipher owned by RSA, designed by Ron Rivest in 1987.
- Became public in 1994.
- Simple and effective design.
- Variable key size (typical 40 to 256 bits),
- Output length unbounded
- Widely used (web SSL/TLS, wireless WEP).
- Extensively studied, not a completely secure PRNG, first part of output biased, when used as stream cipher, should use RC4-Drop[n]
 - Which drops first n bytes before using the output
 - Conservatively, set $n=3072$

Pseudo Random Number Generator

- Useful for cryptography, simulation, randomized algorithm, etc.
 - Stream ciphers, generating session keys
- The same seed always gives the same output stream
 - Why is this necessary for stream ciphers?
- Simulation requires uniform distributed sequences
 - E.g., having a number of statistical properties
- **Cryptographically secure pseudo-random number generator** requires unpredictable sequences
 - satisfies the "next-bit test": given consecutive sequence of bits output (but not seed), next bit must be hard to predict
- Some PRNG's are weak: knowing output sequence of sufficient length, can recover key.
 - Do not use these for cryptographic purposes

Properties of Stream Ciphers

- Typical stream ciphers are very fast
- Widely used, often incorrectly
 - Content Scrambling System (uses Linear Feedback Shift Registers incorrectly),
 - Wired Equivalent Privacy (uses RC4 incorrectly)
 - SSL (uses RC4, SSLv3 has no known major flaw)

Security Properties of Stream Ciphers

- Under known plaintext, chosen plaintext, or chosen ciphertext, the adversary knows the key stream (i.e., $\text{PRNG}(\text{key})$)
 - Security depends on PRNG
 - PRNG must be “unpredictable”
- Do stream ciphers have perfect secrecy?
- **How to break a stream cipher in a brute-force way?**
- If the same key stream is used twice, then easy to break.
 - This is a fundamental weakness of stream ciphers; it exists even if the PRNG used in the ciphers is strong

Using Stream Ciphers in Practice

- If the same key stream is used twice, then easy to break.
 - This is a fundamental weakness of stream ciphers; it exists even if the PRNG used in the ciphers is strong
- In practice, one key is used to encrypt many messages
 - Example: Wireless communication
 - Solution: Use Initial vectors (IV).
 - $E_{\text{key}}[M] = [IV, M \oplus \text{PRNG}(\text{key} || IV)]$
 - IV is sent in clear to receiver;
 - IV needs integrity protection, but not confidentiality protection
 - IV ensures that key streams do not repeat, but does not increase cost of brute-force attacks
 - Without key, knowing IV still cannot decrypt
 - **Need to ensure that IV never repeats! How?**

Coming Attractions ...

- Cryptography: Semantic Security, Block ciphers, encryption modes, cryptographic functions

