

Background Review (2): Secure Communications & Image Proc.

Min Wu

Electrical & Computer Engineering
Univ. of Maryland, College Park

<http://www.ece.umd.edu/class/enee739m/>

minwu@eng.umd.edu



Review of Last Class

- Communication model
 - source coding => channel coding => modulation => CH ...
- Optimal detection of one bit under i.i.d. Gaussian noise
 - Hypothesis testing formulation

=> 1st lecture notes

http://www.ece.umd.edu/class/enee739m/lec/739S02_lec1.pdf

- Today
 - Finish optimal one-bit detection
 - Secure communication: confidentiality and integrity
 - Review of image transform and compression



Conveying One-bit Through Noisy Channel (cont'd)

$$\begin{cases} H_{-1}: y_i = -s_i + d_i & (\text{if } b = -1) \\ H_{+1}: y_i = +s_i + d_i & (\text{if } b = +1) \end{cases} \text{ for } i = 1 \sim n$$

- Optimal detection ~ minimize prob. of error

MAP ~ maximize posterior probability

=> ML ~ maximum likelihood detector [for equal prior]

=> Minimum distance detector [for iid Gaussian noise]

=> Maximum correlation detector [for equal-energy sig.]

- Detection statistics

– [correlator] $\sum_i y_i s_i$

• Prob. distribution under each hypothesis ~ $N(\pm \|\mathbf{s}\|^2, \|\mathbf{s}\|^2 \sigma_d^2)$

– [correlator with unit-variance] $\sum_i y_i s_i / [(\sum_i s_i^2) \sigma_d^2]^{1/2} \sim N(\pm \|\mathbf{s}\|/\sigma_d, 1)$



Performance of Optimal Detector

- Probability of detection error = $\mathcal{L}(\|\mathbf{s}\|/\sigma_d)$
 - $\mathcal{L}(x)$ is monotonically decreasing for non-negative x
 - Signal-to-noise ratio (SNR) ~ $(\|\mathbf{s}\|^2/n) / \sigma_d^2$
- Communications under very low SNR
 - Choose large n
 - collect info. (energy) from many signal components
 - a basic idea behind “spread spectrum communications”
- Useful in invisible watermarking (data hiding)
 - Adding or subtracting a weak signal to convey one-bit hidden info.
 - Will go into more details next time
- Extension for non-i.i.d. Gaussian noise



Secure Communications



Add Security Layers to Communications

- **Confidentiality** \Rightarrow
 - Messages for “your eyes” only
- **Integrity**
 - Message is what sender intended to deliver at this moment
- **Threats and Attacks on information**
 - (1) Use limited info. to find out ways to decipher confidential msg.
 - ♦ *Prefer a system s.t. the best attack strategy is guessing and exhaustive search*
 - \Rightarrow *unbreakable within reasonable time period*
 - (2) Altering a message s.t. authentication system still regard it as unaltered
 - (3) Replaying an old message as if it is being sent by sender right now

\rightarrow *This course will not address network security such as intrusion detection, denial-of-service, and protocol designs.*
See Prof. Gligor's EE757.



Useful Crypto Tools/Building-Blocks

- **Crypto'ly strong one-way function $f(x)$**
 - Easy to compute $f(x)$ given x , but difficult to find x when given $f(x)$
 - Given a set of $(X_i, f(X_i))$ and $f(x)$, difficult to find x
 - SHA and DES are popular choice for one-way function
- **“Low-cost” crypto'ly strong random number generator**
 - Generating truly random seq. via natural randomness ~ flip coins, etc.
 - ♦ *slow and difficult to store/transmit efficiently*
 - ♦ *prefer low cost in both computation and storage/delivery*
 - Use “pseudo-random” generator that can
 - ♦ *Given a subset of output bits, the rest are unpredictable*
 - ♦ *Produce output using a small secret ~ say, a small set of parameters*
 - ♦ *Produce output fast and be easily implementation, say, in software*
 - Use one-way function to generate unpredictable bits $X_j = f(s + j)$
 - ♦ *seed “s”, one-way function “ $f()$ ”*



Useful Crypto Tools/Building-Blocks (cont'd)

- **Crypto'ly strong hash or digest function $H()$**
 - One-way “compression” function
 - ♦ *M-bit input to N-bit output often with fixed N and $M \gg N$*
 - ♦ *Often used to produce a short ID for identifying the input*
 - Properties to be satisfied:
 - 1) Given a message m , $H(m)$ can be calculated very quickly
 - 2) Given a digest y , it is computationally infeasible to find a message m s.t. $H(m) = y$ (i.e., H is one-way)
 - 3) It is computationally infeasible to find messages m_1 & m_2 s.t. $H(m_1) = H(m_2)$ (i.e. H is strongly collision-free)
 - Keyed Hash:
 - ♦ $H(k, m) = \text{Hash}(\text{concatenated string derived from } k \text{ \& } m)$
 - Commonly used crypto hash
 - ♦ *160-bit SHA (Secure Hash Algorithm) by NIST*
 - ♦ *128-bit MD4 and MD5 by Rivest*



Encryption / Ciphers

- **Examples** \cong
 - Shift cipher: e.g. “plaintext” => “sodlqwhaw” (shift by +3)
 - Substitution cipher ~ equiv. to apply a permutation of alphabet to plaintext
 - Stream cipher using XOR ~ $X_i (+) K_i = Y_i$
 - ♦ one-time pad with key size as large as the message
 - Block cipher
 - ♦ encrypt a large block of data at a time to make freq. attack difficult
 - ♦ many modern ciphers are block ciphers
- **Attacks**
 - A small number of searches/guesses
 - Ciphertext and Plaintext attack
 - ♦ use some knowns to find/guess unknowns ~ solving equation arrays
 - Frequency analysis (esp. when plaintext is natural language)



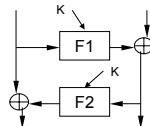
Encryption Keys

- **Symmetric**
 - Encryption and decryption share the same key
 - Key establishment and update are often non-trivial
- **Asymmetric (public-key crypto)**
 - Different keys for encryption and decryption
 - Difficult to derive one key from the other key
 - Useful for confidentiality, identity verification, key establishment, etc.
 - Message for Bob’s eye
 - ♦ Alice encrypts a msg using Bob’s public key
 - ♦ only private key holder can decrypt a ciphertext encrypted by the corresponding public key
 - Message only Bob can produce
 - ♦ Bob encrypts a msg using his private key
 - ♦ only private key holder can produce a ciphertext decryptable by the corresponding public key



A Few Widely Used Ciphers

- **DES and new AES**
 - A building block (“Feistel”) scrambles the input
 - Apply a given number of rounds of Feistel blocks
 - Extensive cryptanalysis
 - ♦ A good crypto system should not rely on the secrecy of the algorithm
- **RSA (public-key encryption):**
 - Security strength based on discrete log problem
 - ♦ Fix a large prime p , let nonzero integer a and $b \pmod{p}$ s.t. $b = a^x$
=> difficult to find x
 - Encryption and Decryption perform exponential modulo operation with different exponents
 - ♦ slow



Data Integrity Verification (data authentication)

- **Authentication is always “relative”**
 - with respect to a reference
- **How to establish and use a reference**
 - [Method-1] Give a “genuine” copy to a trusted 3rd party
 - [Method-2] Append “check bits”
 - Want hard to find a different meaningful msg. with same “check bits”
=> use crypto’ly strong hash
 - Want tamper-proof if hash func. is public
 - ♦ Encrypt concatenated version of message and hash
 - ♦ Keyed Hash (Message Authentication Code) ~ no extra encryption needed
- **Digital signature algo. (using public-key crypto)**
 - ♦ Signed Msg|Hash ~ i.e., encrypt by private key s.t. others can’t forge



References on Cryptography

- **UMCP course -- Math/CMSC456 Cryptology**
 - <http://www.math.umd.edu/~lcw/crypto.html>
 - ◆ *Links*
 - Textbook



Summary

- **Wrap up optimal 1-bit detection**
 - Performance is determined by SNR and signal length (# observations)
 - Detection under low SNR ~ use longer signal
 - ◆ *Spread spectrum communications*
- **Cryptographic tools for secure communications**
 - Building blocks: pseudo-random # generator, one-way func., hash
 - Encryption
 - Integrity verification (tampering detection)



Suggested reading

- [Detection] Poor Chapt. 2.2 & 3.2
- [Crypto] Trappe/Washington Chapt.1,2,4,6,8
- [Image Proc.] Jain's and Wang's book

- Next class:
 - ◆ *Quick review of image proc.*
 - ◆ *Intro. to Data Hiding / Watermarking; Additive embedding*

