# Noisy channels and cryptography

Burton Rosenberg

March 7, 2003

## *Preface*

These notes were written for a combined undergraduate/graduate cryptography course. (Csc609/507 Spring 2003.) This material can mostly be found in the text assigned to the graduate students, *Codes and Cryptography* by Dominic Welsh, but not in the text assigned to the undergraduates, *Introduction to Cryptography with Coding Theory* by Trappe and Washington. These notes are meant to fill the gap for the undergraduate students. We shall cover Welsh's material but not in the depth found in the book.

Furthermore, we give the relationship of noisy channels to cryptography. One aspect of this is found at the end of Welsh's text in the wire-tap channel. The other aspect, oblivious transfer, has probably not yet be presented in the context of an introductory cryptography course.

## Motivation

We have considered the question of secure communication where the channel of communication is noiseless. Alice and Bob communicate through a public channel with Eve eavesdropping on the conversation. All parties hear the same information but due to secret information shared by Alice and Bob, Eve cannot understand their conversation. The security of this model was given by Shannon's theory of perfect secrecy, $H(M|C) = H(M)$, the entropy of the message space does not decrease due to the revelation of ciphertext.

A one-time pad is a perfect cipher. Few other ciphers are perfect. Most can be broken by exhaustive search of the key space. Other approaches, depending upon computational complexity gaps (which are currently conjectured but unproven) are even more imperfect, — Trappe and Washington point out that $H(M|C) = 0$ for RSA! Semanatic security deals with secure communication without complexity assumptions, relating security to communication theory. That is why we are studying this.

## Binary Symmetric Channel

Our model of a communication channel is a box which accepts input symbols and for each input symbol admitted, emits an output symbol with a certain probability. The *binary symmetric channel* accepts a 0 or 1 input and outputs 0 or 1, usually the same symbol as the input, however there is a probability $p$ of error whereby a 0 is changed to a 1 or a 1 changed to a 0.

The channel is memoryless. A symbol is inverted with probability $p$ independent of the history of symbols or errors. For instance, the probability of two consecutive symbols being transmitted with error is $p^2$. We can summarize the probabilities as a matrix of conditional probabilities:

$$[a_{ij}] = \mathrm{Prob}(\sigma_j \text{ output} \,|\, \sigma'_i \text{ input})$$

for all output symbols in the output symbol set $\sigma_j \in \Sigma_O$ and all input symbols in the input symbol set $\sigma'_i \in \Sigma_I$.

$$\left[\begin{array}{cc} 1-p & p \\ p & 1-p \end{array}\right]$$

There is also a cute little drawing of the channel with nodes and arrows, but I'm not currently in the mood to render it into LaTeX's picture mode.

## Binary Erasure Channel

We introduce a channel which drops bits with probability $\epsilon$ but otherwise transmits them correctly. The input alphabet is $\Sigma_I = \{\,0,1\,\}$, the output alphabet is $\Sigma_O = \{\,0,1,*\,\}$ and the channel matrix is:

$$\left[\begin{array}{ccc} 1-\epsilon & 0 & \epsilon \\ 0 & 1-\epsilon & \epsilon \end{array}\right]$$

Again, I should draw this for you in LaTeX picture mode, but it would take some time.

## Encoding of source symbols

In order to reduce error, or at least detect it, the message is first encoded into a block of input symbols. For simplicity, let us assume that our messages are $l$ length strings of 0 and 1, and that we encode into $n$ length strings of 0 and 1 according to some encoding rule,

$$e : \{\,0,1\,\}^l \rightarrow \{\,0,1\,\}^n, \quad l \leq n.$$

The range of the function $e$ are the *codewords* of the encoding and we denote it by $M_n$. The *code rate* is the amount of bit expansion the code entails, calculated by $\log_2 |M_n|/n$.

In class we considered several coding rules including the identity rule, parity, exclusive-or encoding and the repetition encoding,

$$\begin{array}{rcl} e:\ \{\,0,1\,\} & \to & \{\,0,1\,\}^n \\ 0 & \mapsto & \overbrace{0\ldots0}^{n} \\ 1 & \mapsto & \overbrace{1\ldots1}^{n} \end{array}$$

If we consider a decoding rule which accepts only $\overbrace{0\ldots0}^{n}$ and $\overbrace{1\ldots1}^{n}$, discarding anything else. The error probability for a binary symmetric channel will be $1/p^n$, the probability of flipping all $n$ bits. Hence given an arbitrarily small error probability $\epsilon > 0$, letting $n \geq \log_2 \epsilon / \log_2 p$ we can achieve that error probability with a code rate of $1/n$.

The Shannon Noisy Coding Theorem formalizes this observation, giving the theoretical best code rate for a given channel capacity, where channel capacity is related to the channel noise.

## Decoding of channel codes

A codeword $m \in M_n$ will be received at the other end of the binary symmetric channel as possibly any $n$ bit 0, 1 string. The decoding rule must choose a codeword for each such string, including the possibility of a null codeword, meaning that the received code block is to be discarded.

The *Ideal Observer* would ask what is the most likely codeword $m \in M_n$ sent given the observed channel output $\sigma$,

$$\max_{m \in M_n} \text{Prob}(m \,|\, \sigma)$$

and decode $\sigma$ as the maximizing $m$. This rule requires knowledge of both the channel structure and the probability distribution on $M_n$. For instance, suppose $\sigma$ is very close to a codeword $m$, however the probability of $m$ ever being sent is zero. Then $\sigma$ should not be decoded to $m$, but to some other codeword $m'$, even if $m'$ is more dissimilar to $\sigma$.

The *Maximum Likelihood* rule instead maximizes the probability of observing $\sigma$ assuming that the codeword was sent,

$$\max_{m \in M_n} \text{Prob}(\sigma \,|\, m)$$

and decode $\sigma$ as the maximizing $m$. This calculation does not require knowledge of the distribution of $M_n$, only of the channel structure.

Recall Bayes Theorem,

$$\text{Prob}(m \,|\, \sigma) = \frac{\text{Prob}(\sigma \,|\, m)\text{Prob}(m)}{\text{Prob}(\sigma)}.$$

For the case of equally likely codewords, the $m$ maximizing $\text{Prob}(\sigma \,|\, m)$ also maximizes $\text{Prob}(m \,|\, \sigma)$, for a fixed $\sigma$. Hence in the case of equally likely codewords, the Ideal Observer is the same as the Maximum Likelihood.

## Hamming distance

Given two equal length strings over the same alphabet, define the *Hamming distance* to be the number of locations where the strings disagree. For instance, the Hamming distance from 00110 to 10100 is 2. Note that the Hamming distance, $d(\sigma, \tau)$, satisfies the usual axioms for a distance function,

1. $d(\sigma, \tau) \geq 0$ with equality if and only if $\sigma = \tau$.
2. *Symmetry:* $d(\sigma, \tau) = d(\tau, \sigma)$.
3. *Triangle inequality:* for all $\gamma$, $d(\sigma, \tau) \leq d(\sigma, \gamma) + d(\gamma, \tau)$.

For the binary symmetric channel with $p \leq 1/2$, the maximum likelihood rule consists of finding the codeword closest in Hamming distance to the observed channel output. (If $p > 1/2$, complement all output bits and now the channel error is $(1-p) \leq 1/2$.)

## Channel capacity

We now measure the capacity of a channel. Recall our definition of information. Given distributions $S$ and $T$, the *information of $S$ given by $T$* is defined as,

$$\begin{aligned} I(S|T) &= H(S) - H(S|T) \\ &= H(S) + H(T) - H(S,T) \\ &= H(T) - H(T|S) \\ &= I(T|S) \end{aligned}$$

Let $S$ be the distribution of the input to the channel and $T$ the distribution of the output of the channel. We define channel capacity by maximizing $I(S|T)$ over all input sources $S$.

For the binary symmetric channel, the source is a 0, 1 bit source of bias $\alpha$. By symmetry, $I(S|T)$ will be symmetric around $\alpha = 1/2$. We will guess that it is a convex function reaching its maximum at $\alpha = 1/2$. This will save us a lot of fumbling around. Of course, this guess is correct.

For the binary symmetric channel, the conditional entropy is,

$$\begin{aligned} H(S|T) &= -(p(0,0)\log p(0|0) + p(0,1)\log p(0|1) + p(1,0)\log p(1|0) + p(1,1)\log p(1|1)) \\ &= -((1-p)/2\log(1-p) + p/2\log p + p/2\log p + (1-p)/2\log(1-p)) \\ &= -(1-p)\log(1-p) - p\log p. \end{aligned}$$

So $C_{BSC}(p)$, the channel capacity for a binary symmetric channel with error rate $p$, is,

$$C_{BSC}(p) = H(S) - H(S|T) = 1 + p\log p + (1-p)\log(1-p)$$

For the binary erasure channel we make the same, correct, assumption that an unbiased bit source will maximize the channel capacity and calculate,

$$\begin{aligned} H(S|T) &= -(p(0,0)\log p(0|0) + p(0,*)\log p(0|*) + p(1,*)\log p(1|*) + p(1,1)\log p(1|1)) \\ &= -((1-\epsilon)/2\log 1 + \epsilon/2\log 1/2 + \epsilon/2\log 1/2 + (1-\epsilon)/2\log 1) \\ &= \epsilon \end{aligned}$$

So $C_{BEC}(\epsilon)$, the channel capacity of a binary erasure channel with erasure rate $\epsilon$, is,

$$C_{BEC}(\epsilon) = H(S) - H(S|T) = 1 - \epsilon.$$

## Shannon noisy coding theorem

We state the version given in Welsh's book, without proof. Define the maximum error probability of a code to be,

$$\hat{e} = \max_{m\in M_n} \text{Prob}(\text{error} \,|\, m \text{ input})$$

**Theorem 1 (Shannon noisy coding theorem)** *Consider a binary symmetric channel of capacity $C$ and a desired code rate of $R$ where $0 < R < C$. There exists a sequence of codes of code rate $R$ and codeword length $n$, $\{M_n\}_{n=1,2,\ldots}$, such that arbitrarily small $\hat{e} > 0$ is achieved by using codes of sufficiently large codeword length, $n > n_o$, where $n_o$ is a function of $\hat{e}$.*

The meaning of code rate is that $|M_n| \leq 2^{Rn}$. This is also the best possible result. If the code rate exceeds the channel capacity, $C < R$, then the error of transmission cannot be decreased.

## Wyner wiretap channel

Chaos is the natural ally of cryptographers.

Shannon's perfect secrecy result modeled secure communication over noiseless channels. Wyner considered this variant model: let legitimate players Alice and Bob talk over a noiseless channel, and the eavesdropper Eve imperfectly tap the channel. Model Eve's tap as a binary symmetric channel of error probability $0 < p < 1/2$. We shall give a method by which Alice and Bob can communicate while Eve's interception yeilds negligible information about their conversation.

Alice choses $n-1$ random bits $b_1, \ldots, b_{n-1}$. To encode bit $b$, Alice selects the $n$-th bit $b_n$ so that $b_1 \oplus b_2 \oplus \ldots \oplus b_n = b$. Alice sends the bits to Bob who performs the sum and recovers $b$.

The probability that Eve can perform the same calculation and get the correct result is the probability that the error channel makes an even number of errors. Let the result of Eve's calculation be $b'$,

$$\begin{aligned}
\mathrm{Prob}(b = b') &= \sum_{i=0,2,\ldots,n-(n \bmod 2)} \binom{n}{i} q^{n-i} p^i \\
&= (1/2)((q+p)^n + (q-p)^n) \\
&= (1/2)(1 + (1-2p)^n)
\end{aligned}$$

Taking $n$ to infinity, Eve's probability of guessing the bit communicated from Alice to Bob goes to 1/2. Eve might as well flip a coin.

For example, for $p = 1/8$, letting $n = 10$ then Eve's probability of correctly intercepting the bit is 52%. Just a two percent advantage over blind guessing. For $n = 50$ the probability is 50.00003%.

For additional discussion, see Welsh's text. This result was extended by Csiszar and Korner to include independent binary symmetric channels where the error from Alice to Bob is less than the error from Alice to Eve.

Suppose now that both channels are binary symmetric with non-zero error, perhaps with an error rate $\epsilon$ for Alice to Bob larger than the error rate $\delta$ from Alice to Eve. Secure communication is still possible if we assume an error free "back channel". For instance a bulletin board of some sort, public to all and therefore easy to construct reliably.

From the two binary symmetric channels and the error free back channel we construct the equivalent to two binary symmetric channels, one from Bob to Alice with error rate $\epsilon$, and one from Bob to Eve with error rate $\epsilon + \delta - 2\epsilon\delta$. Since $\epsilon < 1/2$, we have that the error rate from Bob to Eve is greater than that from Bob to Alice, so the result of Csiszar and Korner applies.

The construction is as follows. Alice sends Bob a random bit $x$ through the binary symmetric channel. Bob receives $y$ and Eve receives $z$. Bob sends Alice and Eve $y \oplus b$ via the public, error free channel. Alice interprets this as $b_a = x \oplus y \oplus b$. The probability that $b_a \neq b$ is the probablity that $x \neq y$ which is $\epsilon$.

Eve must deduce $b$ from $y \oplus b$ and $z$. Suppose she calculates $b_e = z \oplus y \oplus b$. The probability of $b_e \neq b$ is the probability that $z \neq y$, which is $(1-\epsilon)\delta + \epsilon(1-\delta) = \epsilon + \delta - 2\epsilon\delta$.

## Oblivious transfer

There are other problems in cryptography besides the communication by secret means. Many of these problems have been motivated by the use of computers to communicate or interact. Suppose Alice and Bob want to flip a coin, but they are not face to face, rather they interact by computer communications. How can this be done with each party assured that the flip was fair?

One method proposed is for Alice to chose a bit $b_a$ and send it Bob in a sealed envelope. Bob sends Alice a bit $b_b$ in the same manner. Once both envelopes are received both Alice and Bob open the envelopes and take the exclusive or of the two bits $b_a \oplus b_b$. They interpret the result as heads or tails.

For this to be fair, neither Bob nor Alice should be able to force the outcome of the coin flip. Therefore:

1. Bob cannot know any thing about the bit in Alice's envelope until Alice opens it for him.
2. Alice cannot change her choice of bit once the envelope is sealed.
3. The same conditions hold for Bob's envelope which Alice is holding.

This is called *bit commitment*.

Later in the course we will learn how bit commitment is done using standard complexity theoretic cryptography. Here we describe doing it using a binary erasure channel, without complexity theoretic assumptions. We build bit commitment from *oblivious transfer,* a rather peculiar, but very strong, cryptographic primitive.

**Definition 1 (Oblivious transfer)** *Alice sends Bob a bit which Bob receives with a certain probability $p$. Bob knows if he has received to bit or not. If he does not receive the bit he learns nothing about the bit. Alice does not know if Bob received the bit or not.*

Obviously our binary erasure channel implements oblivious transfer. Here is how we use oblivious transfer to do bit commitment. For Alice to commit to a bit $b$ she chooses $n-1$ random bits $r_1, \ldots, r_{n-1}$ and the bit $r_n$ such that

$$b = r_1 \oplus r_2 \oplus \ldots \oplus r_n$$

She sends these $n$ bits to Bob using oblivious transfer. Bob (probably) does not receive all the bits so he has no idea of $b$, where the probability that Bob receives all the bits can be made arbitrarily small by increasing $n$. However, Alice does not know which bits Bob is lacking. To open the envelope and reveal the bit, Alice sends Bob the bits again, this time using an error free channel. Alice can change $b$ by changing any one of the $r_i$, however if it is one of the $r_i$ that Bob did receive then Bob can detect the cheating. Since a fraction of about $p$ of the bits were lost, Alice's probability of successfully cheating is $p$. Running the commitment protocol multiple times in parallel, we can make the probability of Alice successfully cheating arbitrarily small.

To review:

1. We use the binary erasure channel to implement oblivious transfer.
2. We use oblivious transfer to implement bit commitment.
3. We use bit commitment to flip a coin.