Maximizing Privacy with Minimal Secrecy: Zero delay encoding schemes that achieve optimal secrecy

Michael Freyberger[1], Paul Cuff[1], Sanket Satpathy[1]
[1]*Electrical Engineering, Princeton University*

MOTIVATION

- Shannon has provided an information theoretic definition of **perfect secrecy:** a cryptosystem which encrypts any plaintext message *X,* into cipher text *Y* with the following property:

$$\boldsymbol{P(X \mid Y) = P(X)}$$

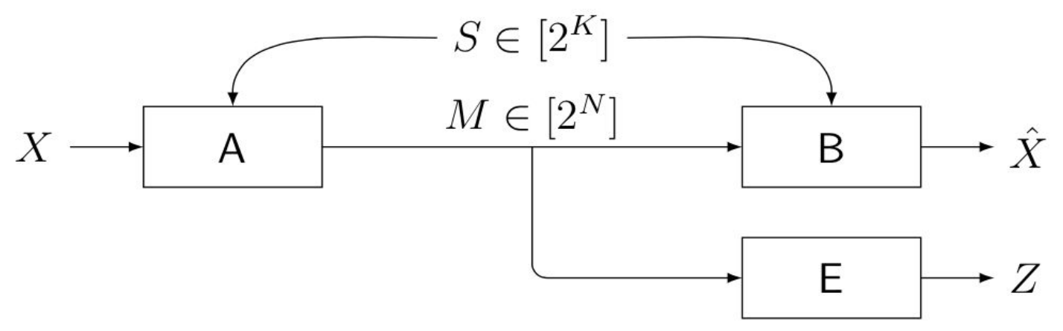
- In order for perfect secrecy to occur, a strict inequality must be met:

$$|\boldsymbol{K}| \geq |\boldsymbol{Y}| \geq |\boldsymbol{X}|$$

- This inequality forces the number of secret key bits to be greater than or equal to the number of bits in the message, which rarely occurs in practice.
- Most cryptosystems rely on a key that is long enough to make decoding at the eavesdropper computationally prohibitive.

What if we do not have access to a shared secret that is long enough? The **constraint for perfect secrecy** is **very limiting**, and there is currently no efficient means of maximizing privacy when the amount of shared secrecy is minimal.

PROBLEM SETUP



$\boldsymbol{X}$: A discrete random variable with a known probability distribution function P($\boldsymbol{X}$)
A: Encoder. Transforms the input source symbol to a public message.
M: Public message. Number of bits per message = **N**.
S: The shared secret key. Number of bits per key = **K**.
B: Decoder. Transforms the public message into a source symbol.
$\widehat{\boldsymbol{X}}$*:* The decoded value. In order for the system to be error free: $\mathrm{P}(\boldsymbol{X} = \widehat{\boldsymbol{X}}) = 1.$
E: Passive eavesdropper. Transforms the public message into a source symbol.
Z: The eavesdroppers guess. Effective encoding and decoding schemes increase $\mathrm{P}(\mathrm{X} \neq \mathrm{Z})$

ASSUMPTIONS

- **E** knows **A** and **B**, the encoding and decoding scheme. **E** has a perfect view of **M**, the public message. **E** does not know **S.**
- All symbols are independent. This removes any correlation between symbols.

$$\boldsymbol{P(X_{k+1}|X_k) = P(X)}$$

- **K < N** and we cannot rely on the key to make decoding at the eavesdropper computationally prohibitive.
- Therefore, a simple **one time pad** will not be most effective**.**
- **One time pad** implementation:
  - Extend the key with a pseudo random generator if necessary.
  - Encrypt the cipher text: XOR the plain text with the key.
  - Recover the plain text: XOR the cipher text with key.

ENCODER & DECODER

- Encoder can be understood as a bipartite graph connecting input source values to bit patterns.
- The amount of incoming edges for each bit pattern is the number of secret key values

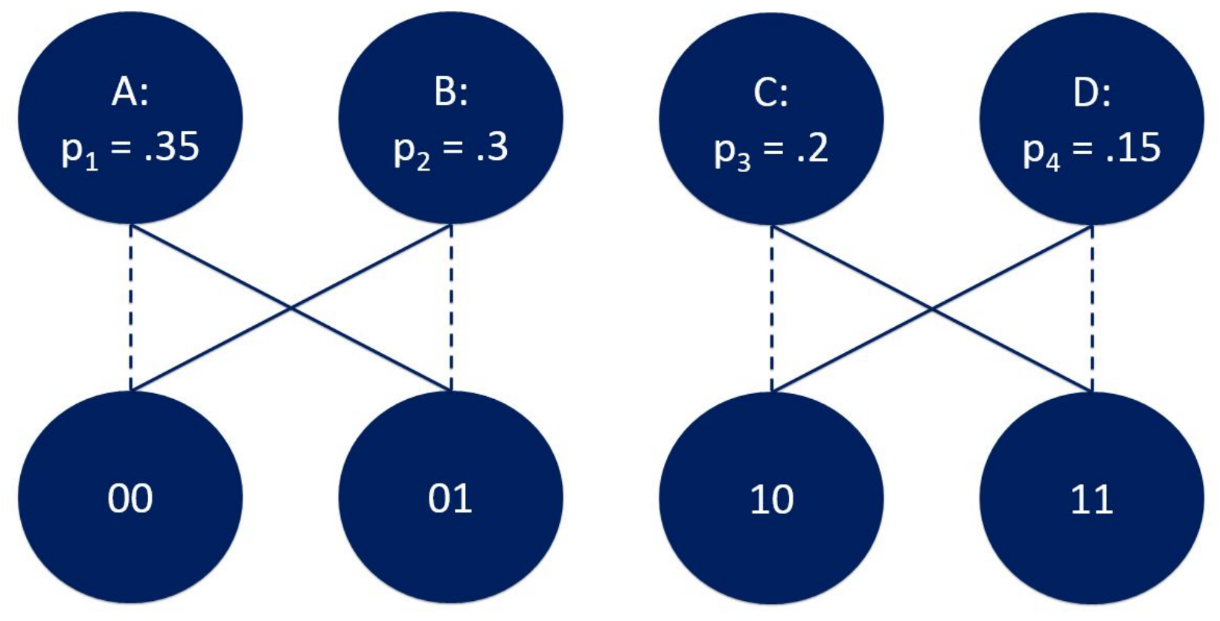


Figure 1: An example of a bipartite graph used for encoding and decoding with one bit of secrecy.

- This graph represents the case **K = 1**. Therefore, there are two possible secret key values.
- Dashed lines correspond to the bit pattern for **S = 0**, and solid lines correspond to the bit pattern for **S = 1.**
- Given the decoder knows the value of **S**, this scheme is error free: $\mathrm{P}(\boldsymbol{X} = \widehat{\boldsymbol{X}}) = 1.$

EAVESDROPPER SCHEME

- For each message **M**, the eavesdropper will select the source symbol that is most probable given the bit pattern.
- All other source symbols connected to that bit pattern will contribute to the **Hamming distortion**.
- There is no use of history. For each message the eavesdropper selects the source symbol that will decrease the probability of error for that specific message.
- This is the worst-case eavesdropper from the point of view of maximizing the eavesdroppers distortion.
- The total **Hamming distortion** for the above encoding scheme:

$$\sum_{C=1}^{C=|M|} \frac{1}{2}\min(P(C_0), P(C_1)) = \frac{1}{2}[.3 + .3 + .15 + .15] = \boldsymbol{.45}$$

where $\mathrm{P(C_0)}$ = Probability of the source symbol connected to a particular bit pattern when **S** = 0, and similarly for $\mathrm{P(C_1)}$ and **S** = 1.

- **Perfect secrecy**, which would occur with **K = 2** in the above example, achieves a Hamming distortion of *.65 = 1 - .35*

KEY TERMS

- **Perfect secrecy** yields the greatest **hamming distortion** given an unlimited number of secret key bits.
- **Optimal secrecy** yields the greatest **hamming distortion** given a fixed number of encodings and fixed number of secret key bits.
- **Maximal secrecy** yields the greatest **hamming distortion** given a fixed number of secret key bits and unlimited number of encodings. Therefore, **maximal secrecy** might not allow for efficient communication.
- The eavesdropper can either guess the key value, or always guess the most probable input source for **maximal secrecy**. Therefore, if the **hamming distortion** for the encoding scheme is:

$$\min(1 - 2^{-K}, 1 - p_{max})$$

than the encoding scheme achieves **maximal secrecy.**

- In the case **maximal secrecy** cannot be met, it is important to determine the encoding scheme that achieves **optimal secrecy**.
- It is possible for schemes that achieve **maximal secrecy** with the **hamming distortion** $= 1 - p_{max}$ to also achieve **perfect secrecy**.
- All schemes that achieve **maximal secrecy**, also achieve **optimal secrecy**.
- **Zero delay encoding** means each source symbol is encoded directly, the block length is equal to 1.

ONE BIT SECRECY

Problem Definition

With one bit of secrecy, the problem is essentially selecting the correct permutation of edges that complete the graph that looks like the following. The graph is initialized with the edges associated with **S = 0**. All possible connections for these initial edges are theoretically the same, so rather than select all 2***|X|** edges, the encoding scheme must only determine the edges for **S = 1.** The encoding scheme must achieve **optimal secrecy.**

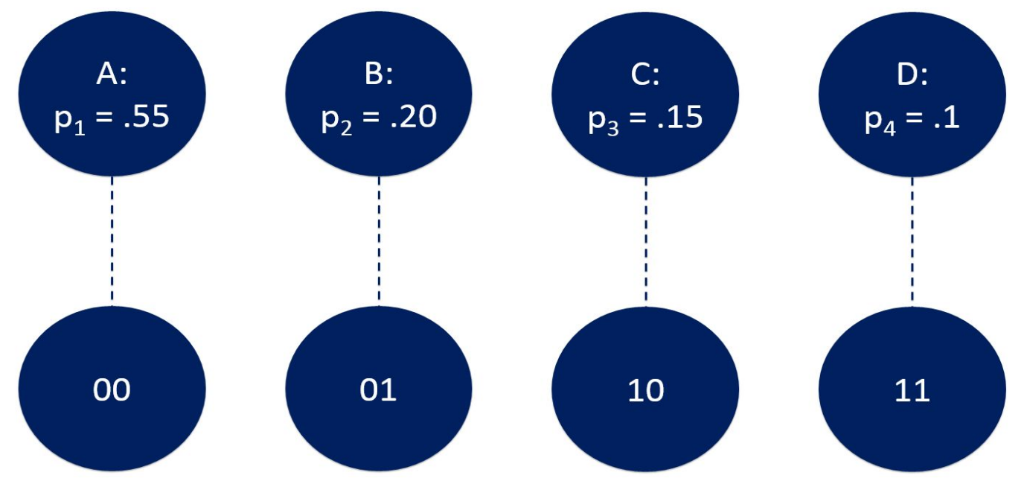


Figure 2: An example of a bipartite graph when the encoding scheme is partially defined for one key value.

Observations

- The search space for this problem is **|X|!**, which is super exponential.
- The final graph could be a disjoint set of subgraphs. Such as the graph provided in Figure 1, which is a disjoint set of 2 subgraphs, each with 2 input source symbols.
- Any subgraph that has 4 input source symbols can be divided into 2 disjoint subgraphs with 2 input source symbols in order to increase the hamming distortion.
- The hamming distortion for a subgraph with two input source symbols is always the smaller probability, p_2 ,because no matter what the key value is, the eavesdropper will select the larger probability.

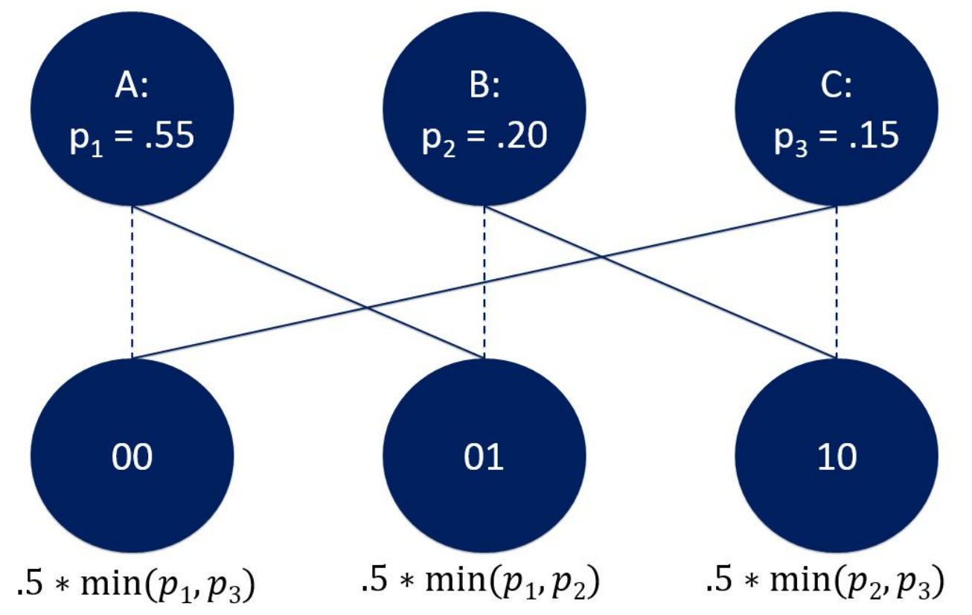


Figure 3: An example of a subgraph with three input source symbols. The hamming distortion associated with each bit pattern is below each node.

- The total distortion for a subgraph of size 3 is therefore $\boldsymbol{p_3 + \frac{1}{2}p_2}$, when $\boldsymbol{p_1 > p_2 > p_3}$.
- The optimal scheme will be some combination of size 2 and size 3 subgraphs.
- The only other possible subgraph is a subgraph of size 1, which could occur on the least likely input source symbol.
- Subgraphs of size 1 introduce zero hamming distortion.

Optimal Encoding

This encoding can be calculated with a **dynamic program** in linear time. The general approach involves optimizing a small subset of the smaller probabilities, and slowly increase the size of the subset until the whole set has been optimized. This optimal encoding only achieves **maximal secrecy** when there are two input source symbols, **|X| = 2.** In Figure 1, the two subgraphs of size 2 yields a distortion of $.45 = p_2 + p_4 > p_3 + .5p_2 = .35$, which would occur with a size 3 subgraph. The encoding in Figure 1 achieves **optimal secrecy.**

STOCHASTIC ENCODING

Problem Definition

Still with only one bit of secrecy, the amount of hamming distortion can be increased by increasing **|M|.** When **|M| > |X|,** the encoder must be stochastic, assuming we are sending the full set of values of **X**, because the same input source symbol must go to two different bit patterns for the same secret key value.

3 Messages, 4 Encodings, 1 Bit of Secrecy

- Some of the input source symbols will have more than 2 outgoing edges.
- When there is more than one outgoing edge for a particular value of **S**, in order to simplify the analysis, intermediate nodes are placed between the source symbols and bit patterns.
- The following two graphs can achieve **optimal secrecy** for a particular input source probability distribution.

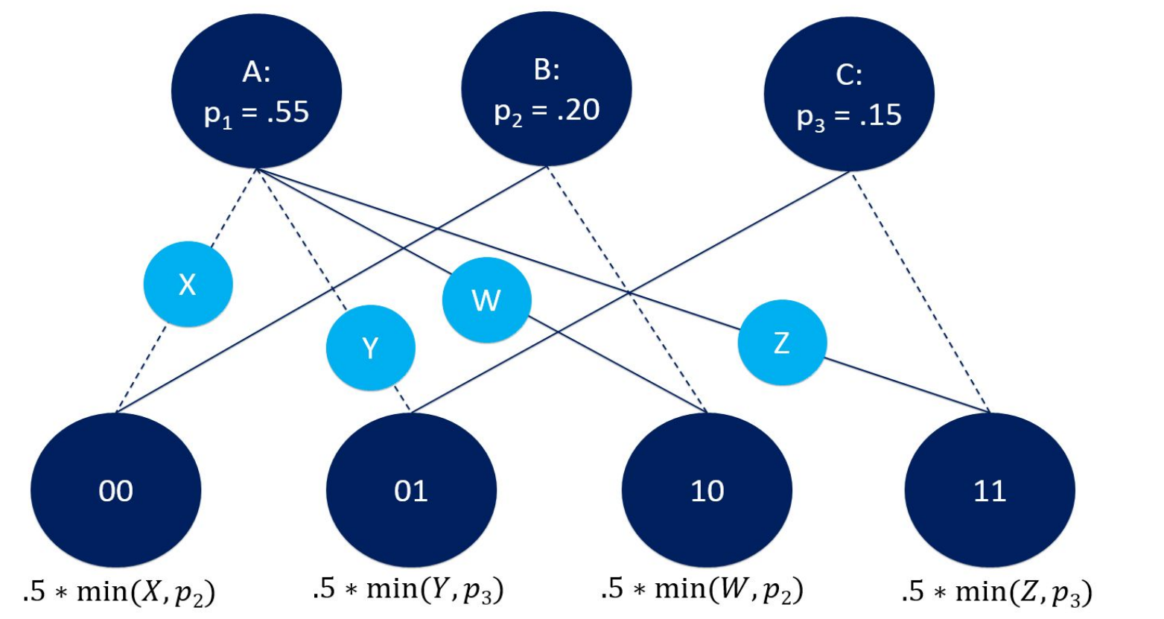


Figure 4: An example of a stochastic encoder with the stochastic process on the most probable input source symbol.

Constraint: $\boldsymbol{X + Y = W + Z = p_1}$
In order this constraint to be met, the intermediate nodes take upon the following values:
$\boldsymbol{Z = Y = p_3}$, $\boldsymbol{X = W = p_1 - p_3}$
When simplified, the total distortion becomes:

$$\boldsymbol{\min(p_1, 1 - p_1)}$$

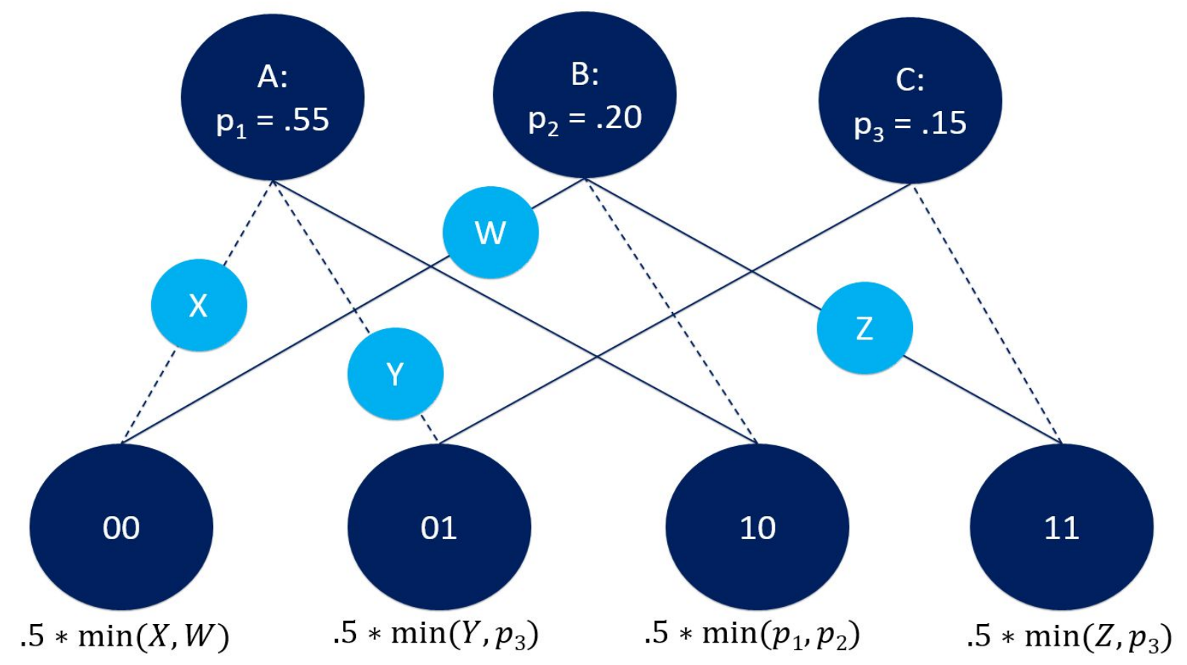


Figure 5: An example of a stochastic encoder with a stochastic process on the most probable, and second most probable input source symbol.

Constraint: $\boldsymbol{X + Y = p_1 \ and \ Z + W = p_2}$
In order this constraint to be met, the intermediate nodes take upon the following values:
$\boldsymbol{Z = p_3}$ $\boldsymbol{W = p_2 - p_3}$ $\boldsymbol{Y = p_3}$ $\boldsymbol{X = p_1 - p_3}$
When simplified, the total distortion becomes:

$$\boldsymbol{p_2 + \frac{p_3}{2}}$$

- Case 1: $p_1 > p_2 + p_3$**:** the top encoding scheme is **optimal** and achieves **maximal secrecy: Hamming distortion** = $1 - p_{max}$
- Case 2: $p_2 + p_3 > p_1 > p_2 + \frac{p_3}{2}$**:** the top encoding scheme achieves **optimal secrecy.**
- Case 3: $p_1 < p_2 + \frac{p_3}{2}$ **:** bottom scheme achieves **optimal secrecy.**