CS 6222 Grad Crypto August 28, 2025

Lecture 2: Shannon's Secretive Studies

Lecturer: Jack Doerner *Scribe: John Berberian, Jr.*

1 Topics Covered

- Claude Shannon
- Definitions of Secrecy (& Their Equivalence)
- One-Time Pad

2 Claude Shannon

Claude Shannon was a fascinating fellow. A brief history of his many contributions to the field of computer science is shown in Figure 1. Today we will be discussing information theory; specifically, the concept of secrecy. Details of his life ranging from his rocket-powered frisbee to his adventures with juggling will be deferred to another lecture.

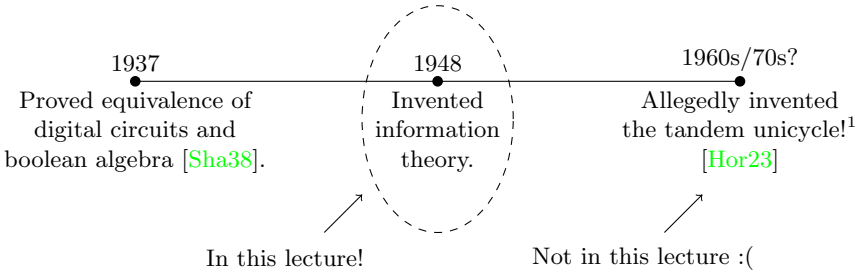

Figure 1: A cursory timeline of Shannon's most significant achievements.

3 Secrecy in Three Flavors

We will start off by revisiting the situation we discussed last class (Figure 2). Alice and Bob are trying to communicate without Eve listening in. To that end, they have established some secret key k ahead of time, by some means upon which Eve could not eavesdrop. Alice is using k and some encryption algorithm Enc to encrypt a secret message m into the ciphertext c ($c = \mathsf{Enc}_k(m)$). Bob, upon receiving the ciphertext, will use a corresponding decryption algorithm Dec and k to recover the message m ($m = \mathsf{Dec}_k(c)$).

[1] Extra credit opportunity: give your final project presentation while riding Shannon's tandem unicycle. For an additional challenge, juggle while you ride and present.

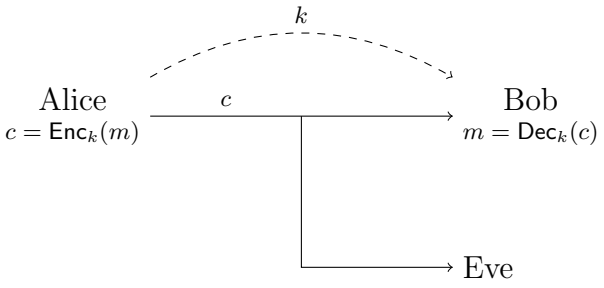

Figure 2: Communication Diagram

Last class, we were trying to informally define what it might mean for their communication to be *secure*. We walked through a few definitions, but none of them was quite right. This time, we're going to use a more formal approach to define exactly what we mean by secure communication. But first we need to introduce the syntax we'll be using.

Definition -1 (Symmetric-Key Encryption Syntax)**.** *Let $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{C}$ be spaces of keys, messages, and ciphertexts, respectively. A* Symmetric-Key Encryption Scheme (SKE)[2] *is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

Gen*: Randomly samples a key from the universe $\mathcal{K}$. Written as $k \leftarrow \mathsf{Gen}$*

Enc*: Operates on $k \in \mathcal{K}, m \in \mathcal{M}$, outputs $c \in \mathcal{C}$. Written as $c \leftarrow \mathsf{Enc}_k(m)$*

Dec*: Operates on $k \in \mathcal{K}, c \in \mathcal{C}$, outputs $m \in \mathcal{M}$. Written as $m := \mathsf{Dec}_k(c)$*

Note 1 (Notation for Sampling and Assignment)**.** *The arrow $\leftarrow$ denotes random assignment from something; the something depends on context. It might be a randomized algorithm,*[3] *it might be a set, it might be a distribution. If it is a set, then sampling is performed from the uniform distribution over that set. The colon-equals $:=$ denotes deterministic assignment, in which there is no randomness whatsoever.*

In order to have a useful encryption-decryption scheme, we need the decryption operation to be the opposite of encryption. For today, we will require the strongest possible version of this: for all possible keys, the decryption algorithm should always (that is, deterministically—not just with some probability) perfectly recover any message that could be encrypted. Put more formally:

Definition 0 (Perfect Correctness)**.** *A symmetric-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* perfectly correct *iff*

$$\forall m \in \mathcal{M}, \quad \Pr[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m : k \leftarrow \mathsf{Gen}] = 1$$

[2]Also known as a *Private*-Key or *Secret*-Key Encryption Scheme.

[3]Be aware, an implicit translation applies for algorithms: one may write the randomness of the random assignment ($\leftarrow$) as an explicit parameter in a deterministic assignment ($:=$). By convention, any randomness parameters are separated from other parameters by a semicolon, rather than the comma usually used for parameters. So $c \leftarrow \mathsf{Enc}_k(m)$ translates to $c := \mathsf{Enc}_k(m; r)$, where r is a random value. Unless otherwise specified, r is sampled from the uniform distribution over appropriately-long sequences of bits.

Note 2 (Notation for Probability)**.** *The equation in Definition 0 measures the probability that the* event *(or predicate) given on the left-hand side of the colon occurs in the* experiment *(or setup) given on the right-hand side, over the random coins of all algorithms involved. Some authors (including Pass and shelat [Ps10]) write the event on the right and the experiment on the left.*

So we are able to recover the message that we put in—excellent! But we're greedy; we want more. We would also like the encryption scheme to keep the contents of the message secret. In short, seeing the ciphertext c should not give an attacker any more information about the message contents than they already had. If we wanted to describe this concept more formally, we might say that the probability distribution of messages D is identical before and after conditioning on the ciphertext, for all possible ciphertexts.

Definition 1 (Shannon Secrecy)**.** *A symmetric-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* Shannon-secret *iff, for all message distributions D on $\mathcal{M}$, all messages $m^* \in \mathcal{M}$, and all ciphertexts $\forall c \in \mathcal{C}$ such that* $\Pr[c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D] > 0$*, we have*

$$\Pr[m^* = m | c = \mathsf{Enc}_k(m) : m \leftarrow D, k \leftarrow \mathsf{Gen}] = \Pr[m^* = m : m \leftarrow D]$$

To gain some intuition about this, we might imagine that Eve is wondering if the message m is some specific message m^*. Before seeing the ciphertext, she calculates some probability $\Pr[m^* = m : m \leftarrow D]$ based on D, which is some expected distribution of messages. Then she sees some ciphertext derived from m and k (it must be a valid ciphertext—that's what the $\Pr[...] > 0$ condition is saying). If the cipher is Shannon-secret, seeing this ciphertext must not result in a change in how likely she thinks $m = m^*$ is—and this must be true for all possible combinations of keys, messages m and m^*, and ciphertexts that could be derived from m and k. This is a pretty strong secrecy condition: it means that Eve isn't gaining any information at all about what the message is when she sees the ciphertext.

We'll also discuss another sort of secrecy. Another way to make sure that we don't reveal anything about the message is to have each possible ciphertext be equally probable for each possible message, over the random choice of the key.

Definition 2 (Perfect Secrecy)**.** *A symmetric-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* perfectly secret *iff*

$$\begin{aligned} &\forall m_1, m_2 \in \mathcal{M}, \forall c \in \mathcal{C}, \\ &\Pr[\mathsf{Enc}_k(m_1) = c : k \leftarrow \mathsf{Gen}] = \Pr[\mathsf{Enc}_k(m_2) = c : k \leftarrow \mathsf{Gen}] \end{aligned}$$

This one is much easier to interpret. For any pair of messages, all ciphertexts must be equally likely for each of the two.[4] It turns out that this definition is exactly equivalent to the last one. We will prove one direction of that implication; the other direction is left as an exercise to the reader. It may also be found on pg. 13 of Pass and shelat [Ps10].

[4]Note that two distinct messages can only yield the same specific ciphertext under *different* keys if the scheme is perfectly correct. Here we are arguing only about distributions.

Theorem 1. *Definition 2 implies Definition 1.*

Proof. For some perfectly secret encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we begin with the left-hand side of Definition 1.

$$\Pr[m^* = m | c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D]$$

This conditional probability can be rewritten as a quotient.[5]

$$\frac{\Pr[m^* = m \wedge c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D]}{\Pr[c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D]}$$

If the event described in the numerator occurs, $m = m^*$. This means it is equivalent to express the second condition in that expression in terms of m^*.

$$\frac{\Pr[m^* = m \wedge c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}, m \leftarrow D]}{\Pr[c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D]}$$

We recognize that the two intersected events in the numerator are entirely independent of one another. Moreover the first does not depend on k and the second does not depend on m. Therefore we can express the intersection as a product.

$$\frac{\Pr[m^* = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]}{\Pr[c = \mathsf{Enc}_k(m) : k \leftarrow \mathsf{Gen}, m \leftarrow D]}$$

The denominator is the probability that the fixed ciphertext c results from sampling a key according to Gen and a message according to D. We can rewrite that as the sum over all possible messages m' of the probability that the message m' is sampled from D, and the probability that m' yields the ciphertext c.

$$\frac{\Pr[m^* = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]}{\sum_{m' \in \mathcal{M}} \Pr[m' = m \wedge c = \mathsf{Enc}_k(m') : k \leftarrow \mathsf{Gen}, m \leftarrow D]}$$

We can apply independence again to separate the denominator.

$$\frac{\Pr[m^* = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]}{\sum_{m' \in \mathcal{M}} \Pr[m' = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m') : k \leftarrow \mathsf{Gen}]}$$

At this point, we may apply the perfect secrecy property we assumed at the beginning of this proof. By Definition 2, $\Pr[c = \mathsf{Enc}_k(m') : k \leftarrow \mathsf{Gen}] = \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]$ for all m', m^*. We substitute that in the denominator.

$$\frac{\Pr[m^* = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]}{\sum_{m' \in \mathcal{M}} \Pr[m' = m : m \leftarrow D] \Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]}$$

Since $\Pr[c = \mathsf{Enc}_k(m^*) : k \leftarrow \mathsf{Gen}]$ is constant with respect to m', we can factor it out of the sum and cancel it with the matching term in the numerator.

$$\frac{\Pr[m^* = m : m \leftarrow D]}{\sum_{m' \in \mathcal{M}} \Pr[m' = m : m \leftarrow D]}$$

The denominator is equivalent to integrating D over $\mathcal{M}$, so it must be 1. □

[5] The expressions in this proof are all equal, but I don't have sufficient line width to write two of them side-by-side with an equals sign in the middle. I hope you can make up in imagination what I lack in space.

The definitions haven't yet explicitly mentioned Eve. So let's play a little game with Eve. This is much closer to how we'll be making definitions for the rest of the class. Let us imagine that Eve has given us two messages, m_1 and m_2. We select one of them at random, encrypt it, and send it. Then Eve will try to guess which one we sent.[6] We would like it to be impossible for Eve to distinguish which message we sent—that is, her chance of guessing which of her two messages we chose should be no better than guessing randomly. This property is called *perfect indistinguishability*.

Definition 3 (Perfect Indistinguishability)**.** *A symmetric-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* perfectly indistinguishable *iff for all two-part unbounded algorithms* $(\mathcal{A}_1, \mathcal{A}_2)$*,*

$$\Pr\left[b' = b : \begin{array}{r}(m_0, m_1, s) \leftarrow \mathcal{A}_1, b \leftarrow \{0,1\}, \\ k \leftarrow \mathsf{Gen}, c \leftarrow \mathsf{Enc}_k(m_b), b' \leftarrow \mathcal{A}_2(c, s)\end{array}\right] = \frac{1}{2}$$

If the above definition is bit tough to read, then it may be easier to start by reading the experiment (i.e. start reading at the colon) and come back to the event later. In the experiment, we begin by getting a pair of messages (m_0, m_1) from Eve, whose choices we represent with the algorithm $\mathcal{A}_1$. The algorithm also gives us some opaque state s. Later on, when she sees the ciphertext, we'll describe her choices with $\mathcal{A}_2$. The s parameter exists to allow her to remember something between those two interactions—essentially, this is allowing those two algorithms to communicate. We flip a coin (b) to select our message, and encrypt it with a key that we generate. Then Eve looks at the ciphertext (and remembers anything she put in s) and makes a guess b' about which message we chose. The process she follows to arrive at that guess is described by $\mathcal{A}_2$. If the encryption scheme is perfectly indistinguishable, Eve's guess about which message we chose should be exactly as good as a random guess (1/2 chance of being correct), no matter what algorithm she uses to determine it. In other words: Eve can never do any better[7] than random guessing when playing this game.

This is a pretty strong statement—and note that it doesn't leave room for any particularly weak messages! Eve gets to stack the deck here, so to speak. If there were certain messages that were easier to tell apart than others, Eve could choose those and do better than a random guess. In order to satisfy perfect indistinguishability, Eve cannot gain any useful information about the message by looking at the ciphertext. It turns out this is exactly equivalent to perfect secrecy and (by consequence) Shannon secrecy.

4 One-Time Pad

We now come to a fascinatingly-constructed cipher: the one-time pad (OTP). It predated Shannon by a good bit,[8] but he was the first to prove its security. Let's walk through how it works.

[6] Presumably, the messages will be padded to be the same length. Otherwise figuring out which one we've sent would be pretty easy.

[7] Nor, by consequence of the binary choice, any worse!

[8] Wikipedia claims it originated around 1882. Take that with your preferred wikipedian salt dosage.

Definition 4 (One-Time Pad for n-bit Messages)**.** *Let* $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$*. Then, the tuple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is defined as:*

$$\begin{aligned}\mathsf{Gen} &: k \leftarrow \{0,1\}^n \\ \mathsf{Enc} &: k, m \mapsto k \oplus m \\ \mathsf{Dec} &: k, c \mapsto k \oplus c\end{aligned}$$

In this definition, $\oplus$ denotes bitwise XOR. We can verify the correctness of this scheme:

$$\begin{aligned}\mathsf{Dec}_k(\mathsf{Enc}_k(m)) &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m && \text{by associativity} \\ &= 0 \oplus m \\ &= m\end{aligned}$$

It happens that this scheme is also perfectly secret. We could use any of the three definitions we gave for secrecy to prove this (they are, after all, equivalent), but the second one is the most straightforward to use.

Theorem 2. *OTP is perfectly secret*

Proof. We begin with one side of Definition 2. For any pair of $m \in \mathcal{M}, c \in \mathcal{C}$,

$$\begin{aligned}\Pr[\mathsf{Enc}_k(m) = c : k \leftarrow \mathsf{Gen}] &= \Pr[k \oplus m = c : k \leftarrow \{0,1\}^n] \\ &= \Pr[k = c \oplus m : k \leftarrow \{0,1\}^n] \\ &= 2^{-n}\end{aligned}$$

The last step deserves some justification. We haven't put any conditions on m and c here, so $m \oplus c$ is just some fixed value in the space $\{0,1\}^n$. The key k is uniformly sampled from that space, so each key value has the same probability of occurring: 2^{-n}. Since $\Pr[\mathsf{Enc}_k(m) = c : k \leftarrow \mathsf{Gen}]$ is independent of the message m, it must be equal for any pair $m_1, m_2 \in \mathcal{M}$. □

This is really cool! But also really inconvenient: the keys need to be as large as the message and you can't reuse a key. If you do reuse a key, you suddenly reveal a lot of information to Eve: she can compute $c_1 \oplus c_2 = (m_1 \oplus k) \oplus (k \oplus m_2) = m_1 \oplus m_2$. But this possibility for a leak doesn't contradict our proof! We assumed (see Definition 4) that you randomly generate the key for each message. Used properly, this scheme is unbreakable. It is absolutely and entirely impossible to break. Note that we've put no conditions on our adversary here. Eve might have infinite computational power or resources or anything—even an oracle that can solve undecidable problems. None of that matters; it cannot be broken if we play by the rules we've established.

In fact, not only is it unbreakable, but it turns out to be the most efficient unbreakable scheme. Alas, we cannot get perfect security without paying for our lunch somehow: in this case, long keys. Let's try to prove that.

Theorem 3. *If a symmetric-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is Shannon-secret and perfectly correct, then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. Let D be uniform over $\mathcal{M} = \{0,1\}^n$. We will fix c such that $\Pr[c = \mathsf{Enc}_k(m) : m \leftarrow D, k \leftarrow \mathsf{Gen}] > 0$. Let us notate the set of possible messages that could lead to c as

$$M(c) = \{m \in \mathcal{M} : \exists k \in \mathcal{K} \text{ such that } m = \mathsf{Dec}_k(c)\}$$

Since Dec is deterministic,[9] we know that the set $M(c)$ must be at most as large as $\mathcal{K}$.

$$|M(c)| \leq |\mathcal{K}|$$

Assume toward contradiction that $|\mathcal{K}| < |\mathcal{M}|$. Then, there must be some message m' in $\mathcal{M}$ that is not in $M(c)$.

$$\exists m' \in \mathcal{M} \text{ such that } m' \notin M(c)$$

Then, by correctness:

$$\Pr[m = m' | c = \mathsf{Enc}_k(m) : m \leftarrow D, k \leftarrow \mathsf{Gen}] = 0$$

But this is not equal to $\Pr[m = m' : m \leftarrow D] = 2^{-n}$. We have contradicted Shannon secrecy. If we want both correctness and shannon secrecy, we cannot have $|\mathcal{K}| < |\mathcal{M}|$; instead we must have $|\mathcal{K}| \geq |\mathcal{M}|$. □

Corollary 1. *The key length of OTP is optimal for a perfectly-secret, perfectly-correct encryption scheme.*

Note 3 (A fun fact)**.** *if you have even one bit more message than key, Eve's chance of correctly distinguishing between messages (as in Definition 3) rises to at least* 5/8*, no matter what you do [Ps10].*

Next time, we will begin to constrain our adversary. If we make assumptions about Eve's computing power or ability to solve certain hard problems, we can build much more practical schemes.

References

[Hor23] John Horgan. My meeting with claude shannon, father of the information age. `https://johnhorgan.org/cross-check/my-meeting-with-claude-shannon-father-of-the-information-age`, 2023.

[Ps10] Rafael Pass and abhi shelat. A course in cryptography. `https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf`, 2010.

[Sha38] Claude E. Shannon. A symbolic analysis of relay and switching circuits. *Transactions of the American Institute of Electrical Engineers*, 57(12):713–723, 1938.

[9] This must be true if the scheme is perfectly correct.