Secure encryption is subject to many requirements. So how should one go about building one? Before we begin doing cryptography, we must first understand the rules of the game. That is the objective of today's lecture.

Let's agree on some terminology first. The *honest parties* in a cryptographic scheme are the entities running the scheme's algorithms. In encryption, these are the encryptor Alice and the decryptor Bob. The adversary is the entity whose goal is to break the system. In encryption, this is Eve cracking the ciphertext. We have no control over how she will go about it.

1 Kerchoff's postulate

Cryptography is made possible by asymmetries in information. In encryption, Alice and Bob have a secret key that Eve doesn't. Is there anything else that Alice and Bob know but Eve doesn't?

Kerchoff's postulate has to do with knowledge of *algorithms*. Should Eve know how Alice encrypts and how Bob decrypts, or should the description of those algorithms be withheld? Historically, the inner workings of ciphers were closely guarded secrets. Surely Alice and Bob would gain nothing by revealing how they process their data. Modern cryptography assumes the opposite.

Kerchoff's Postulate. All algorithms in a cryptographic scheme are known to the adversary.

I know of three arguments in favor of Kerchoff's Postulate.

The methodological argument is that Kerchoff's postulate provides a useful abstraction: It separates public knowledge (the mechanics of the algorithms) from private information (the secret key). We can then clearly separate what Eve knows from what she doesn't know when reasoning about security.

The technological argument is that nothing can be gained by hiding the algorithm. Any information worth hiding can simply be incorporated into the shared secret key.. For example, if Alice and Bob intend to use a shift cipher, but double the shift on every fifth application, they can just prepend their secret key with the instruction `double the shift every fifth time`. There is no advantage in making this instruction part of the algorithm.[1]

The sociological argument is perhaps the most convincing one: Cryptography, like any scientific discipline, should develop in the open. Exposing candidate cryptographic designs to a large audience of scientists, practitioners, and students makes it more likely that potential vulnerabilities are detected. Conversely, if no weakness is found after a long period of study, our confidence in its security increases.

2 The Relative Effort postulate

The Relative Effort postulate has to do with the amount of effort, or work, that the honest parties and the adversary are willing to dedicate.

Relative Effort Postulate. The attacker puts substantially more effort into breaking the scheme than the honest parties take to run it.

To understand this postulate let us compare the effort expended by the honest parties and by the adversary in the ciphers from Lecture 1. Alice's and Bob's efforts are determined by the mechanics of the scheme. For instance in Caesar's cipher Alice's effort consists of shifting each letter forward by three. Bob shifts it back by three. On a modern computer both operations can be done in a tiny fraction of a second.

[1] Looking ahead, this argument is quite sensible for private-key cryptography, but will have to be revised in the public-key setting, where Alice and Bob do not have the opportunity to secretly share information.

Caesar's cipher, however, does not adhere to the Relative Effort Postulate. Eve can crack a ciphertext *with the same effort* that it took Bob to decrypt it.

How about the substitution cipher? There Alice and Bob merely had to substitute one letter for another to run the scheme. Eve, on the other hand, had to resort to frequencies, digrams, and statistical analysis to break it. Eve arguably expended more effort than Alice and Bob. But was the difference in effort substantial enough? What counts as substantial anyway?

There are two ways to answer this question. One is to look at the real world. Nowadays virtually all internet communication is encrypted. Encryption is ubiquitous because it can be performed at virtually no cost. The overhead of encrypted access to a website is a few milliseconds. How long does it take to break this encryption? Can it be cracked on my laptop computer by the end of the week? No. How about on my desktop computer by the end of the year? How about on the Amazon cloud by the end of the decade? Not a chance. The sun will burn out before the cloud gets anywhere.

This overwhelming disparity between making and breaking encryption is the fruit of ingenious advances in applied cryptography over the years. Cryptographic engineers setting mind-boggling security standards—the lifetime of the universe in the case of encryption—and develop remarkable technologies that meets these standards.

Before applied cryptographers can begin their work, however, they need a proof of concept that the Relative Effort Postulate is at all realistic for their problem. This is where theoretical cryptographers come in. Using ideas from probability and computation, the theory of cryptography can precisely quantify the gap in effort between honest parties and attackers. To explain it, we turn to the mother of all encryption schemes.

3 The One-time Pad

The one-time pad is an extreme variant of the short pad (Vigenère) cipher from Lecture 1. In the one-time pad, the key is as long as the message. The key symbols are uniformly random and independent. To encrypt, Alice adds the message and the key letter-wise modulo 26:

```
    lady catherine i have nothing further to say you know my sentiments
+   nakh dlnausfwj q hocj vjnltwd paxzhez se tum cfb pggy hv dnabzwfkmd
=   YANF FLGHYJNJN Y OOXN IXGSBJJ UUOSOIQ LS LUK ATV ZTUU TT VRNUHIJXFV
```

To decrypt, Bob subtracts the key from the ciphertext and recovers the plaintext.

Is the one-time pad vulnerable to frequency attacks? Not at all. How about digram attacks? Not a chance. The one-time pad is unbreakable. To understand why, we need a brief detour into statistics.

4 Indistinguishability

Recall that a random variable is a function that assigns a value to every outcome in a probability space. In cryptography the values are numbers in modular arithmetic, or sequences of numbers.

There are two common ways to describe a random variable. I could tell you the probability of every possible value a.k.a. the *probability mass function (pmf)*. Alternatively, I could describe a procedure that generates samples of it.

A *sampler* is an algorithm that converts pure randomness into a random value. The output of a sampler is a random variable. *Pure randomness* means a sequence of independent uniform values over some prespecified set like the English alphabet $\{\mathtt{a}, \ldots, \mathtt{z}\}$, or the bits $\{\mathtt{0}, \mathtt{1}\}$. Here is an example:

Sampler S: Output the triplet $(a, b, a + b)$ modulo 2, where a, b are pure random bits.

The pmf of the output of S is shown in Table 1a.

TABLE 1: (a) pmf of the output of S; (b) samples from S and T.

outcome	probability
000	1/4
011	1/4
101	1/4
110	1/4

110	011	101	000	000	000	101	000
110	101	110	000	101	101	101	011
101	011	110	110	101	101	110	110
110	011	011	101	101	110	011	110

The output of every sampler is a random variable with an associated pmf. However, the same pmf can arise from different samplers. For instance, here is another sampler whose output pmf is also the one in Table 1a:

Sampler T: Output $(a+b, b+c, c+a)$ modulo 2, where a, b, and c are pure random bits.

The fact that the same random variable can be sampled in different ways has an important consequence for cryptography: No matter how many samples Eve observes, she has no way to guess which sampler they came out of. To make it concrete, Table 1b shows two batches of sixteen samples, one produced by S, the other one by T. Can Eve tell which is which?

The inability to tell *how* a random variable was sampled from the outcome of sampling is called indistinguishability. It is the second most important concept you will learn in this course.

Definition 1. Two random variables are *perfectly indistinguishable* if they have the same probability mass function.

Thus the outputs of S and T are perfectly indistinguishable.

5 The gold standard of security

The one-time pad is unbreakable because encryptions of any two messages are perfectly indistinguishable. Suppose Eve observed the ciphertext

$$c = \texttt{YANFFLGHYJNJNYOOXNIXGSBJJUUOSOIQLSLUKATVZTUUTTVRNUHIJXFV}$$

We saw that this ciphertext arises as an encryption of the message m_1: `lady catherine i have nothing further to say you know my sentiments`. But could just as easily arise as an encryption of another message under a different key:

m_2 :		`read it aloud said their father for i hardly know myself what it is ab`
	+	`hwnc xs gwkpk rnql vqjag bsicfd pab g bqupaw anfz nvcqio zknb zp bf fu`
c	=	`YANF FL GHYJN JNYO OXNIX GSBJJU UOS O IQLSLU KATV ZTUUTT VTNU HI JX FV`

From Eve's perspective, there are two hypotheses (among many others) that are consistent with her having observed ciphertext c. Hypothesis 1 is that Alice encrypted m_1. Hypothesis 2 is that Alice encrypted m_2. Does observing c help Eve pick one hypothesis over the other in any way? It doesn't. The reason is that the encryptions of m_1 and m_2 are perfectly indistinguishable.

Let's explain why. Had Alice encrypted m_1, which starts with an `l` (11), how likely is Eve to observe `Y` (24) as the first symbol of the ciphertext? The only way this would have happened is if `n` (13) was the first symbol of the key because 13 is the only value that, when added to 11, gives 24 modulo 26. As all keys are equally likely, the probability of Eve observing this `Y` is 1/26. By the same argument, the probability of Eve observing `A` as the second symbol is an independent 1/26, and so on. So the probability that Eve observed ciphertext c given that Alice encrypted m_1 is $1/26^n$, where $n = 56$ is the message length.

What if Alice had encrypted m_2 instead? By the same reasoning, the only way the first ciphertext symbol is a `Y` is if the first message letter `r` (17) was shifted by key symbol `h` (7), an outcome of probability 1/26, and so on, leading to the same probability $1/26^n$ of having observed ciphertext c.

Thus the probability of Eve observing ciphertext c is the same under hypotheses 1 and 2. The act of observing c helps Eve not one bit in favoring one hypothesis over the other. But there was nothing special about our specific choices of ciphertext c and messages m_1 and m_2 to hypothesize about. No matter which ciphertext Eve observes, it does not help her in favoring any one plaintext over any other. This is statistics' way of saying that Eve did not learn anything about the plaintext by observing the ciphertext.

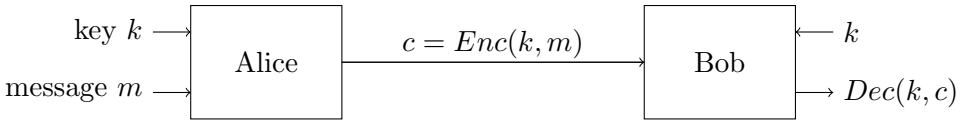


FIGURE 1: The components of private-key encryption.

It is instructive to reframe this argument using indistinguishability. Figure 1 illustrates the components of a private-key encryption scheme. The encryptor Alice takes two inputs—the key k and the message m—and outputs the ciphertext. If we fix the message, we can view Alice as a sampler: She converts the pure randomness of the key k into a ciphertext c. For two different fixed messages m_1 and m_2 this sampler works differently. In the first case, Alice adds m_1 to her key modulo 26. In the second case, she adds it to m_2. Nevertheless, the resulting ciphertexts are indistinguishable.

Theorem 2. *In the one-time pad, the encryptions of any two messages are perfectly indistinguishable.*

This is a mathematical theorem. Even if you dislike mathematical proofs, it is worth taking the time to understand why this one is true. To simplify things let's pretend that the message and the key are a single number modulo 26. The encryption of any fixed message m under the random key k is the number $m + k$ modulo 26. When we fix m, the function that maps k to $m + k$ becomes a permutation. It rearranges the numbers modulo 26. Permutations preserve uniformity of random variables. If k is equally likely to take each of the 26 possible values, so is $m + k$.

Consequently, no matter what the message m is, the encryption $m + k$ is uniformly distributed—all ciphertexts are equally likely—provided that k itself is. But why can we assume that all keys are equally likely? This is our third and last postulate.

Free Randomness Postulate. All participants have unlimited access to pure randomness.

In the context of private-key encryption, the Free Randomness Postulate says that the key Alice and Bob agreed on was chosen uniformly at random from all possible keys, with all symbols mutually independent. How Alice and Bob should make this choice in the real world is a fascinating question, but one outside the scope of cryptography. The Free Randomness Postulate endows them with this ability.

6 Distinguishing advantage

To get a feel for cryptographic logic, it is useful to reformulate Theorem 2 at a higher level of abstraction. This theorem says that the one-time pad is secure in a particular sense. Let's give it a name.

Definition 3. An encryption scheme is perfectly *message indistinguishable* if the encryptions of any two messages are perfectly indistinguishable.

The one-time pad is thus message indistinguishable. What have we gained from this definition?

Mathematically, message indistinguishability is a universally quantified predicate. It asks that for every pair of messages m_1 and m_2, their encryptions cannot be told apart by Eve. We can visualize this dry definition as a *game* between Alice and Eve. Eve picks the two messages and asks Alice to encrypt either one. Alice picks one of them and shows the resulting ciphertext c to Eve. To win the game, Eve has to guess whether c comes from m_1 or from m_2 (see Figure 2).

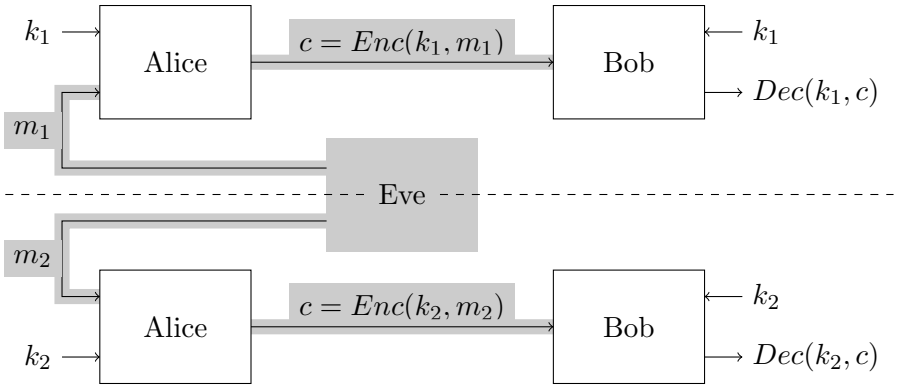


FIGURE 2: Message indistinguishability. After choosing m_1 and m_2 Eve cannot tell apart their encryptions.

One strategy available to Eve is to guess at random, without even looking at the ciphertext. Her chances of getting it right are 50 percent. Message indistinguishability tells us that she can do no better after seeing c. The ciphertext has revealed to her nothing of use about the identity of the message.

Message indistinguishability is a *notion of security* for encryption. Framing it as a game between the adversary Eve and the honest parties Alice and Bob has two advantages. The pedagogical advantage is that it shifts our perspective as students of cryptography from passive listeners to active participants. It invites us to place ourselves in Eve's position and concoct strategies for besting Alice and Bob.

The methodological advantage is even more significant. We can now precisely quantify what it means for Eve to succeed in breaking encryption. To do this we need one more concept from statistics.

Suppose Eve observes an outcome, and her objective is to tell whether it was sampled by S or by T. If she thinks the sample comes from S, she should accept. If she thinks it comes from T, she should reject.

Definition 4. Eve's *advantage* in distinguishing S from T is the difference in probabilities that she accepts a sample from S versus a sample from T:

$$\text{advantage} = \Pr(\text{Eve accepts } S) - \Pr(\text{Eve accepts } T).$$

To make sense of advantage let's look at a few examples.

Example 1. Suppose S samples a random even number from 0 to 9, and T samples a random odd number from 0 to 9. Clearly Eve should accept if, and only if, she observes an even number. Her advantage is

$$\Pr(S \text{ is even}) - \Pr(T \text{ is even}) = 1 - 0 = 1.$$

Advantage one means that Eve can *always* tell random variables S and T apart.

Example 2. Now S is a uniformly random number between 0 and 9 and T is always 0. If Eve observes any nonzero value she should clearly accept. What if she observes 0? Zero is a possible outcome of both S and T. However, all other things being equal, it is much more likely to have come from T than from S: Outcome zero has probability 1 in T but only probability 1/10 in S. Thus Eve should reject zero. Eve's advantage now is

$$\Pr(S \text{ is nonzero}) - \Pr(T \text{ is nonzero}) = 9/10 - 0 = 9/10.$$

Eve's advantage is not quite one, but it is high. She is quite likely to distinguish outcomes of S and T.

Example 3. What if S is uniformly random between 0 and 9 and T is a random odd number in the same range? An even number can never come from T, so Eve should clearly accept those. An odd number can

come from S or from T, but all things being equal it is more likely to come from T, so Even should reject it. Eve's advantage is

$$\Pr(S \text{ is even}) - \Pr(T \text{ is even}) = 1/2 - 0 = 1/2.$$

Is advantage half high or low?

Example 4. Finally, suppose S and T are the two samplers from Section 4. What is Eve's advantage in distinguishing them? Each outcome has exactly the same probability in S as in T, so Eve should be indifferent to accepting or rejecting it. No matter what Eve does, her distinguishing advantage will be zero. As expected, there is no advantage to be gained for perfectly indistinguishable random variables.

We can now give yet another formulation of perfect message indistinguishability: No matter what Eve does, her advantage in distinguishing encryptions of any two messages is zero.

Framing security in terms of distinguishing advantage is helpful for gauging compatibility with the Relative Effort postulate. If Eve is unable to get any distinguishing advantage for the one-time pad, it is pointless for her to put any effort into breaking it. Not even an infinite amount of effort will pay off.

We will see shortly that the one-time pad is unique among encryption schemes in that it achieves zero distinguishing advantage. To move beyond it we will have to concede to Eve from time to time. We then need to make a quantitative call. Should we call things secure if Eve fails to win 99% of the time or, say, just 1% of the time? Or should we settle for 50%?

Cryptographers set extremely high standards for security. If Eve's advantage is anything but a tiny number above zero, say 2^{-127}, the scheme is considered insecure. Eve should not be able to win even if the rules of the game are stacked overwhelmingly in her favor.

There are two reasons why such a stringent degree of security is the norm in cryptography. First, it is achievable. Throughout this course we will see examples of cryptographic functionalities that are effectively unbreakable. Second, it often happens in cryptography that what appears to be a minor and insignificant vulnerability is an omen of a complete breakdown. If Eve discovers an attack that succeeds once in a million attempts, she can often greatly amplify her advantage with just a little more effort. This is why cryptographers are uncomfortable tolerating insecurity beyond a negligibly small advantage.

7 Shannon's theorem

The one-time pad has a glaring inconvenience. It can only be used once. What happens if Alice encrypts multiple messages with it?

To be specific, suppose Alice's key is `lqaz` and she encrypts the three messages `lets take five`. Eve will observe the ciphertexts `WUTR EQKD QYVD`, namely their encryptions under the repeated key `lqaz lqaz lqaz`. The one-time pad, used multiple times, is no different from the short pad (i.e., the Vigenère cipher). It is completely insecure.

For the one-time pad to be used securely, the shared key of Alice and Bob has to be as long as all the messages that are ever to be exchanged between them. In this day and age, when messages are routinely gigabytes long, agreeing on such long random keys in advance is impractical, if not impossible. What is the alternative?

Famed mathematician Claude Shannon was the first to make sense of this question. He framed it like this. Suppose you are allowed to pick any encryption scheme you like. The only requirement is that *the key should be shorter than the message* by any amount, even by one letter. Shannon concluded that under such conditions, encryption cannot be message indistinguishable. In contrapositive form,

Shannon's theorem. In any perfectly message indistinguishable scheme, the key must be at least as long as the message.

The next part is optional but highly recommended.

Here is the reason Shannon's theorem is true. We will argue the contrapositive. Suppose you concocted an encryption scheme with a key shorter than the message. We will conclude that Eve *can* distinguish encryptions of *some* pair of messages.

Suppose Alice used this scheme to encrypt some message m_1 into ciphertext c. How many (other) messages can possibly encrypt to c?

First, no two messages can encrypt to c under the same key. If they did, Bob would not be able to decrypt c under this key. Thus the key determines the message that encrypted to c (if any).

Therefore, the number of possible messages that encrypt to c under *some* key can be at most as large as the number of keys. If the key has length ℓ, then there are 26^ℓ possible keys. This accounts for at most 26^ℓ messages.

However, if the key is shorter than the message, then there are more messages than there are keys. If $\ell < \ell_m$ then $26^{\ell_m} > 26^\ell$. By the pigeonhole principle, there must then be at least one message that cannot encrypt to c under any key! Let's call this message m_2.

We conclude that c is a possible encryption of m_1, but not a possible encryption of m_2. In other words, the probability that Alice encrypted m_1 to c is nonzero, but the probability that she encrypted m_2 to c is zero. So the two are distinguishable!

Cryptographers are, by temperament, a rather optimistic lot. They are not easily deterred by impossibility. A theorem cannot be wrong, but perhaps it is the standard that is too high. What if Eve is allowed *some* distinguishing advantage?

Unfortunately, the same Shannon showed that not only can Eve win the message indistinguishability game, but she can do so by a great margin. If $\ell < \ell_m$, Eve can distinguish m_1 from m_2 with advantage $1 - 26^{\ell - \ell_m}$. If, say $\ell = 100$ and $\ell_m = 110$, this is extremely close to one. How does Eve do that?

Eve's strategy is pretty simple. To be concrete, suppose Eve wants to distinguish encryptions of $m_1 =$ `gorgonzola` from encryptions of $m_2 =$ `prosciutto`. She observes ciphertext $c =$ `SKTWENCESL`. How should she go about distinguishing?

Eve's most advantageous strategy is to try all possible keys. If m_1 encrypts to c under any of the 26^{10} possible keys, she knows that she cannot possibly be looking at an encryption of m_2. So she can distinguish between the two.

Using this strategy, Eve can reliably distinguish encryptions of some pair of messages. She can in fact distinguish most pairs. The catch is that this brute-force strategy requires Eve to try 26^ℓ possible keys. If the key is of length $\ell = 100$ this is an enormous number!

In conclusion, Shannon's theorem says that when the key is shorter than the message, breaking encryption is always possible. But it may take a lot of work for Eve to do that. This is our opening gambit for building encryption that is mind-bogglingly secure yet extremely fast.

Before we conclude, let us draw one more realization from Shannon's line of reasoning. Shannon says that if the key is shorter than the message, then *any* encryption can be broken by trying all 26^ℓ possible keys. Anything that Alice makes with a key of length ℓ, Eve can break in 26^ℓ attempts. Qualitatively, the attacker can break anything that the honest parties can make, provided she is willing to put in exponentially more effort.

This important insight reveals the limits of the Relative Effort Postulate. The best security we can hope to achieve is an *exponential gap* between the efforts expended by the honest parties and the attacker. The closer a cryptographic scheme is to meeting this exponential gap, the more valuable it is in practice.

8 Summary

As scientists, we are taught to question everything. Before we start questioning anything, however, we must agree on some universally accepted *postulates*—the rules of the game.

In cryptography, Kerchoff's postulate gives the adversary complete access to the cryptographic scheme's blueprint. The free randomness postulate gives everyone the one resource that is indispensable for cryptography: randomness.

Kerchoff's postulate and the free randomness postulate are useful *conventions*. Without them, cryptography would still be possible, at least in some form. However, our valuable attention would be lost on irrelevant details. Without these two postulates, we would have a hard time seeing the cryptographic forest from its trees.

The Relative Effort Postulate occupies a different place altogether. Here it is, rehearsed and renamed:

The Central Dogma of cryptography:

The attacker puts substantially more effort into breaking the scheme than the honest parties take to run it.

The one-time pad is the primordial incarnation of the Central Dogma. It is message indistinguishable. Eve cannot break it no matter how much effort she puts in. Perfection, however, comes at a high price: The key must be as long as all the messages Alice and Bob are to exchange, ever. Eve can, in principle, break any encryption scheme whose keys are shorter than its messages. But she may have to work exponentially harder than Alice and Bob.